

МАТЕРИАЛЫ ЗАДАНИЙ
командной инженерной олимпиады школьников
«Олимпиада Кружкового движения
Национальной технологической инициативы»
по профилю
«Искусственный интеллект»

2020/21 учебный год

<http://nti-contest.ru>

Оглавление

1 Профиль	
«Искусственный интеллект»	5
I Первый отборочный этап	7
I.1 Задачи первого этапа. Информатика	7
I.1.1 Первая попытка. Задачи 8–11 класса	7
I.1.2 Вторая попытка. Задачи 8–11 класса	24
I.1.3 Третья попытка. Задачи 8–11 класса	41
I.1.4 Четвертая попытка. Задачи 8–11 класса	58
I.2 Задачи первого этапа. Математика	66
I.2.1 Первая попытка. Задачи 8–9 класса	66
I.2.2 Первая попытка. Задачи 10–11 класса	69
I.2.3 Вторая попытка. Задачи 8–9 класса	74
I.2.4 Вторая попытка. Задачи 10–11 класса	77
I.2.5 Третья попытка. Задачи 8–9 класса	82
I.2.6 Третья попытка. Задачи 10–11 класса	84
I.2.7 Четвертая попытка. Задачи 8–9 класса	88
I.2.8 Четвертая попытка. Задачи 10–11 класса	90
II Второй отборочный этап	97
III Заключительный этап	107
III.1 Индивидуальный предметный тур	107
III.1.1 Информатика. 8–11 класс	107
III.1.2 Математика. 8-9 класс	123
III.1.3 Математика. 10-11 класс	127
III.2 Командный практический тур	131
III.2.1 Задача командного тура	131
III.2.2 Актуальность задачи	132

III.2.3 Описание данных	132
III.2.4 Метрика и формат решений	133
III.2.5 Базовое решение	134
IV Критерии определения победителей и призеров	148

Профиль «Искусственный интеллект»

Цель профиля «Искусственный интеллект» — развитие у школьников прикладных навыков в сфере искусственного интеллекта. Профиль знакомит школьников с методами и технологиями, применяемыми для решения актуальных для науки и бизнеса задач в сфере искусственного интеллекта. В 2020/21 учебном году участникам было предложено углубиться в сферу обработки естественного языка (NLP). Разработки в данном направлении активно используются для автоматизации колл-центров и сервисов клиентской поддержки, в чат-ботах и поисковых машинах. Учащиеся 8–11 классов, прошедшие все этапы Олимпиады КД НТИ, демонстрировали понимание основных операций, необходимых для построения модели машинного обучения: подготовка и анализ данных с целью выявления признаков, необходимых для решения поставленной задачи, построение самой модели и интерпретация результатов ее работы для оценки качества. Для того чтобы сделать это возможным в ходе олимпиады проводился цикл образовательных и отборочных мероприятий.

Первый отборочный дистанционный этап предполагает индивидуальное участие и определяет общий уровень подготовки школьников по предметам математика и информатика. Задачи по информатике относятся к разделам алгоритмы, программирование и методы оптимизации, решая которые, школьники должны продемонстрировать знание численных методов и навыки составления и отладки программ, обрабатывающих массивы данных, и понимание таких тем, как комбинаторика, операции со строками, вычислительная геометрия, теория графов. Задачи по математике проверяли у участников знания по алгебре, комбинаторике, теории вероятности и математической статистике. Количество попыток сдачи решения задач не ограничивалось. Таким образом, задачи первого этапа выявляли наличие у участников знаний необходимых не только для решения задач следующего этапа, но и задач заключительного этапа.

Задача второго отборочного этапа заключалась в формировании навыков по работе с текстом, необходимых для решения задачи заключительного этапа, а также в отборе наиболее сильных участников, поэтому задача решалась каждым участником самостоятельно. В ходе отборочного этапа было проведено 3 вебинара от разработчиков задачи: с введением в тему вопросно-ответных систем, разбором базового решения задачи и вопросов от участников соревнования. Помимо этого, участникам были предоставлены материалы по программированию на Python и использованию основных библиотек для анализа данных, основам машинного обучения, теории вероятностей, практикумы и сборники задач прошлых сезонов олимпиады. Эти и дополнительные знания позволили участникам создать алгоритм, способный ответить на вопросы по тексту, которые требуют построения причинно-следственных связей.

Для подготовки к заключительному этапу для участников был проведен образовательных хакатон, в котором им было предложено решить одну из базовых задач по машинному чтению, адаптированную для русского языка — Стэнфордский датасет с ответами на вопросы (SQuAD). В рамках хакатона для участников был проведен дополнительный вебинар с разбором этой задачи и ответами на вопросы от участников соревнования.

Командная задача практического тура заключительного этапа стала следующим логичным шагом в работе с выбранной темой. Участникам предстояло создать алгоритм, который дополняет текст, учитывая не только доступную информацию, но и здравый смысл. Для решения данной задачи было предложено 3 набора данных, состоящих из 75 000 новостных вырезок, собранных из открытых источников. Два набора предназначались для обучения модели и настройки параметров, третий — для проверки.

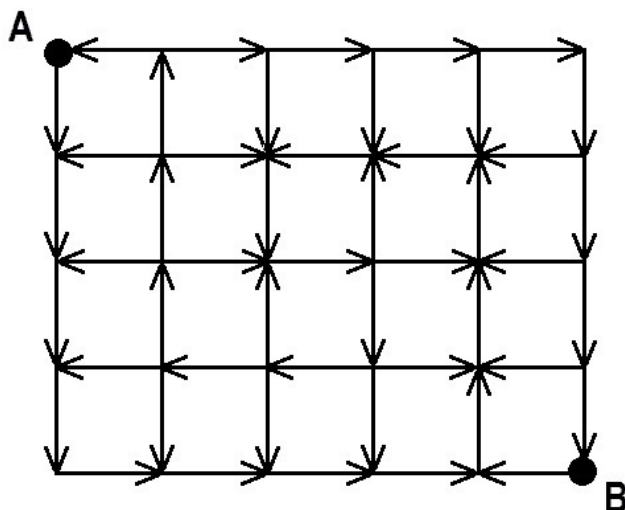
Также в заключительный этап входил индивидуальный предметный тур, в ходе которого участники решали задачи по математике и информатике. Задачи по математике покрывали следующие области математики: оптимизация, комбинаторика, алгебра и геометрия, теория вероятности. А темы задач по информатике проверяли компетенции участников в области программирования, написания кода и отладки программ, применения теории алгоритмов, графов, численных методов и др. — все, что необходимо решению задач на анализ данных.

Первый отборочный этап

Задачи первого этапа. Информатика

Первая попытка. Задачи 8–11 класса

Задача I.1.1.1. Город перекрестков (20 баллов)



Вы разрабатываете навигатор для одного города. Этот город разбит улицами на квадратные кварталы, причем движение по любому из отрезков улицы в пределах каждого квартала строго одностороннее. С каждого перекрестка можно выехать только в разрешенных знаками направлениях. Требуется по прилагаемой карте города с указанными на ней разрешенными направлениями перемещения проложить самый короткий маршрут из точки A в точку B .

Формат входных данных

На вход подается карта перекрестков города. В первой строке содержатся два числа N — число кварталов с севера на юг и M — число кварталов с запада на восток ($1 \leq n \leq 50$). Точка A самая северо-западная, точка B самая юго-восточная. Далее в $2 \cdot N + 1$ строках содержится описание разрешенных направлений движения. Улицы города запад-восток описаны в нечетных строках. В каждой такой строке содержится по M символов без пробела, указывающих разрешенное движение на соответствующем участке. В четных строках содержится описание улиц север-юг. В этих строках содержится по $M + 1$ символов, указывающих возможное движение по отрезкам улиц север-юг. Движение на север, юг, запад, восток обозначается буквами n , s , w , e соответственно.


```

1 //Алгоритм Дейкстры
2
3 #include <bits/stdc++.h>
4
5 #define pb push_back
6 #define mp make_pair
7 #define f first
8 #define s second
9
10 #define all(x) (x).begin(), (x).end()
11 #define sz(a) (int)(a).size()
12 #define for0(i, n) for (int i = 0; i < (int)(n); i++)
13 #define for1(i, n) for (int i = 1; i <= (int)(n); i++)
14 #define int long long
15
16 using namespace std;
17 typedef pair<int, int> pii;
18 typedef long double ld;
19
20 int dx[4] = {-1, 0, 1, 0},
21     dy[4] = {0, 1, 0, -1};
22 char z[4] = {'n', 'e', 's', 'w'};
23
24 signed main(){
25
26     ios::sync_with_stdio(0), cin.tie(0), cout.tie(0);
27
28     int n, m;
29     cin >> n >> m;
30     vector<vector<vector<int> > > G(n+1, vector<vector<int> >(m+1));
31
32     string s;
33     for0(i, 2*n+1){
34         cin >> s;
35         if(i%2){
36             int tx = i/2;
37             for0(j, sz(s))
38                 if(s[j] == 's')
39                     G[tx][j].pb(2);
40             else
41                 G[tx+1][j].pb(0);
42
43         }
44         else{
45             int tx = i/2;
46             for0(j, sz(s))
47                 if(s[j] == 'w')
48                     G[tx][j+1].pb(3);
49             else
50                 G[tx][j].pb(1);
51         }
52     }
53
54     string inf;
55     inf.resize(6000, '#');
56     vector<vector<string> > d(n+1, vector<string>(m+1, inf));
57     vector<vector<int> > mark(n+1, vector<int>(m+1, 0));
58     d[0][0] = "";
59
60     for0(u, (n+1)*(m+1)){

```



```

61     int tx, ty, mn = 1e9;
62     for0(i, n+1) for0(j, m+1) if(mark[i][j] == 0 && sz(d[i][j]) < mn){
63         tx = i;
64         ty = j;
65         mn = sz(d[i][j]);
66     }
67
68     mark[tx][ty] = 1;
69
70     for0(i, sz(G[tx][ty])){
71         int tdir = G[tx][ty][i];
72         int nx = tx + dx[tdir];
73         int ny = ty + dy[tdir];
74         string ts = d[tx][ty] + z[tdir];
75
76         if(sz(ts) < sz(d[nx][ny]) || (sz(ts) == sz(d[nx][ny]) && ts < d[nx][ny]))
77             d[nx][ny] = ts;
78     }
79 }
80
81 cout<<sz(d[n][m])<<endl;
82 cout<<d[n][m];
83 }

```

Пример программы-решения

Ниже представлено решение на языке C++.

```

1  //Обход в ширину
2
3  #include <bits/stdc++.h>
4
5  #define pb push_back
6  #define mp make_pair
7  #define x first
8  #define y second
9
10 #define all(x) (x).begin(), (x).end()
11 #define sz(a) (int)(a).size()
12 #define for0(i, n) for (int i = 0; i < (int)(n); i++)
13 #define for1(i, n) for (int i = 1; i <= (int)(n); i++)
14 #define int long long
15
16 using namespace std;
17 typedef pair<int, int> pii;
18 typedef long double ld;
19
20 int n, m;
21 int dx[4] = {0, -1, 1, 0},
22     dy[4] = {1, 0, 0, -1};
23 char z[4] = {'e', 'n', 's', 'w'};
24
25 bool in_sq(int x, int y){
26
27     return (x >= 0 && y >= 0 && x <= 2*n && y <= 2*m);
28 }
29
30 signed main(){
31     ios::sync_with_stdio(0), cin.tie(0), cout.tie(0);

```

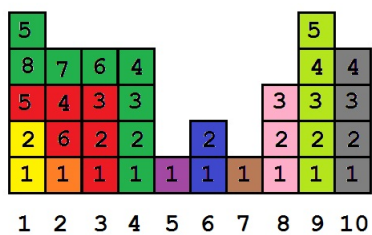
```

32
33     string ans[200][200];
34     cin >> n >> m;
35     vector<string> G;
36     string s;
37     for0(i, 2*n+1){
38         cin >> s;
39         string z;
40         if(i%2 == 0){
41             for(auto q : s) z = z + "0" + q;
42             z += "0";
43         }
44         if(i%2 == 1){
45             for(auto q : s) z = z + q + " ";
46         }
47         G.pb(z);
48     }
49     deque<pii> och;
50     och.pb({0, 0});
51     G[0][0] = '#';
52     ans[0][0] = "";
53
54     while(sz(och) > 0){
55         pii tp = och[0];
56         och.pop_front();
57
58         for0(i, 4){
59             int nx = tp.x + dx[i];
60             int ny = tp.y + dy[i];
61             if(in_sq(nx, ny) && G[nx][ny] == z[i] && G[nx + dx[i]][ny + dy[i]] == '0'
62                ↪ ){
63                 G[nx + dx[i]][ny + dy[i]] = '#';
64                 ans[nx + dx[i]][ny + dy[i]] = ans[tp.x][tp.y] + z[i];
65                 och.pb({nx + dx[i], ny + dy[i]});
66             }
67         }
68
69         cout<<sz(ans[2*n][2*m])<<endl;
70         cout<<ans[2*n][2*m];
71     }

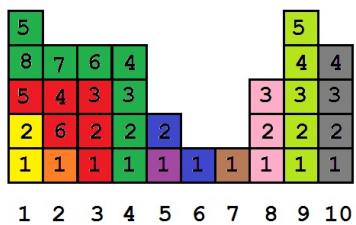
```

Задача I.1.1.2. WALL-E и кубики (20 баллов)

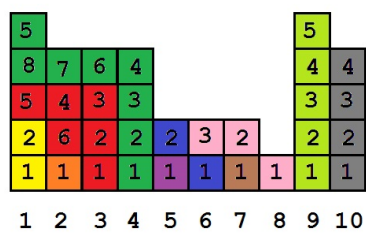
Робот WALL-E пытается упорядочить груду кубиков. В зоне его ответственности есть n расположенных друг за другом столбиков, каждый состоит из кубиков, стоящих друг на друге (в i -м столбике содержится h_i кубиков, столбики нумеруются слева направо). Робот действует по следующему алгоритму: он ищет самый левый столбик с номером i , в котором кубиков больше, чем в предыдущем, то есть $h_{i-1} < h_i$. Далее он берет верхний кубик столбика номер i и сбрасывает его на столбик номер $i - 1$. Если существует столбик с номером $i - 2$ и в нем теперь меньше чем в $i - 1$ -м столбике, то данный кубик перемещается далее на $i - 2$ -й столбик. И так до тех пор, пока этот кубик, либо не дойдет до верха первого столбика либо не упрется в более высокий столбик слева. Далее эта последовательность операций повторяется до тех пор, пока есть два рядом стоящих столбика таких, что $h_{i-1} < h_i$. По значениям высот исходных n столбиков вывести высоты упорядоченных n столбиков.



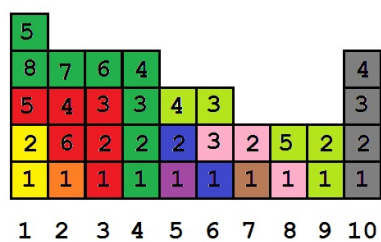
4. Упорядочили столбик 6



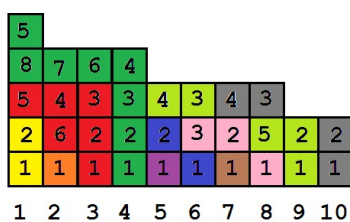
5. Упорядочили столбик 8



6. Упорядочили столбик 9



7. Упорядочили столбик 10, итоговое расположение кубиков

*Примеры**Пример №1*

Стандартный ввод
10
2 1 6 8 1 2 1 3 5 4
Стандартный вывод
5 4 4 4 3 3 3 3 2 2

Пример программы-решения

Ниже представлено решение на языке C++.

```

1 //Стек nap
2 #include <bits/stdc++.h>
3
4 #define pb push_back
5 #define mp make_pair
6 #define x first
7 #define y second
8
9 #define all(x) (x).begin(), (x).end()
10 #define sz(a) (int)(a).size()
11 #define for0(i, n) for (int i = 0; i < (int)(n); i++)
12 #define for1(i, n) for (int i = 1; i <= (int)(n); i++)
13 #define int long long
14
15 using namespace std;
16 typedef pair<int, int> pii;
17 typedef long double ld;
18
19 signed main(){
20     ios::sync_with_stdio(0), cin.tie(0), cout.tie(0);
21
22     int n;
23     cin >> n;
24     vector<int> h(n);
25     for0(i, n) cin >> h[i];
26
27     vector<pii> st;
28     st.pb({2e9, 1});

```

```

29
30     for0(i, n){
31
32         if(h[i] < st.back().x) st.pb({h[i], 1});
33         else
34         if(h[i] == st.back().x) st.back().y++;
35         else
36         if(h[i] > st.back().x){
37             int thb = st.back().x;
38             int ost = h[i] - thb;
39             int d = 1;
40             int v = 0;
41             int sump = 0;
42
43             bool ok = 0;
44             while(!ok){
45                 int sp = d * v - sump;
46
47                 if(ost <= sp){
48                     ok = 1;
49                     int tr = ost + sump;
50                     int c = tr / d;
51                     int o = tr % d;
52
53                     if(o > 0){
54                         int thh = thb + c + 1;
55
56                         if(thh == st.back().x) st.back().y += o;
57                         else st.pb({thh, o});
58                     }
59
60                     int thh = thb + c;
61
62                     if(thh == st.back().x) st.back().y += (d - o);
63                     else st.pb({thh, d - o});
64                 }
65                 else{
66                     d += st.back().y;
67                     v = st[sz(st)-2].x - thb;
68                     sump += (st.back().x - thb) * st.back().y;
69                     st.pop_back();
70                 }
71             }
72         }
73     }
74
75     for(int i = 1; i < sz(st); i++)
76         for0(j, st[i].y) cout<<st[i].x<<' ';
77 }

```

Задача I.1.1.3. Послание внеземного разума (10 баллов)

Сенсация! Известный специалист по UFO профессор Персигов получил послание от внеземного разума. Во всяком случае он так считает. Послание представляет собой непрерывную последовательность из сигналов нескольких типов, каждый из этих типов для определенности можно обозначить малой буквой латинского алфавита. Профессор считает, что доказательством искусственности происхождения сигнала служит его периодичность. При этом период должен быть строго равен «константе

Персикова» — числу P . Такую последовательность назовем P — периодичной.

К сожалению, некоторые сигналы из-за прохождения сквозь бездны пространства (а может даже и времени) были утеряны. Их заменили на знак вопроса. Теперь профессору для дальнейшей работы требуется выяснить, можно ли как-то заменить утерянные сигналы на буквы так, чтобы последовательность стала P — периодичной. Для чистоты эксперимента полученная из космоса последовательность была размещена среди себе подобных. Не подведите, в ваших руках — будущее межгалактических взаимоотношений!

Формат входных данных

В первой строке содержится два числа через пробел: N — общее количество строк, которые вам нужно проверить на P -периодичность и число P — константа Персикова ($1 \leq P \leq 5 \cdot 10^3$). Далее содержится N непустых строк, состоящих из малых букв латиницы и знаков вопроса. $1 \leq N \leq 5 \cdot 10^4$, суммарная длина всех строк не превосходит $2 \cdot 10^5$.

Формат выходных данных

Вывести N строк, в каждой из которых вывести либо «YES», если соответствующую строку можно сделать P -периодичной путем замены всех знаков вопроса на некоторые буквы, и «NO» в противном случае. Будем считать последовательность P -периодичной, если для любых двух символов, расстояние между которыми кратно P верно, что они совпадают.

Примеры

Пример №1

Стандартный ввод
8 4 abacabaca abracadabra aa?aaa?aaaaaaaaaa q ??????? q?er?w? q?erw?? q?er?wer?werw?er
Стандартный вывод
YES NO YES YES YES YES NO NO

Пример программы-решения

Ниже представлено решение на языке C++.

```

1 //Метод карманов
2 #include <bits/stdc++.h>
3 #define pb push_back
4 #define mp make_pair
5 #define x first
6 #define y second
7
8 #define all(x) (x).begin(), (x).end()
9 #define sz(a) (int)(a).size()
10 #define for0(i, n) for (int i = 0; i < (int)(n); i++)
11 #define for1(i, n) for (int i = 1; i <= (int)(n); i++)
12 #define int long long
13
14 using namespace std;
15 typedef pair<int, int> pii;
16 typedef long double ld;
17
18 signed main(){
19     ios::sync_with_stdio(0), cin.tie(0), cout.tie(0);
20
21     int u, p;
22     cin >> u >> p;
23     while(u--){
24         string s;
25         cin >> s;
26         vector<set<char> > v(p);
27         for0(i, sz(s))
28             if(s[i] != '?') v[i%p].insert(s[i]);
29
30         bool ok = 1;
31         for(auto q : v)
32             if(sz(q) > 1)
33                 ok = 0;
34
35         cout<< ((ok)? "YES\n" : "NO\n");
36     }
37 }

```

Пример программы-решения

Ниже представлено решение на языке C++.

```

1 //Эффективный перебор
2 #include <bits/stdc++.h>
3
4 #define pb push_back
5 #define mp make_pair
6 #define x first
7 #define y second
8
9 #define all(x) (x).begin(), (x).end()
10 #define sz(a) (int)(a).size()
11 #define for0(i, n) for (int i = 0; i < (int)(n); i++)
12 #define for1(i, n) for (int i = 1; i <= (int)(n); i++)

```



```

13  #define int long long
14
15  using namespace std;
16  typedef pair<int, int> pii;
17  typedef long double ld;
18
19  signed main(){
20      ios::sync_with_stdio(0), cin.tie(0), cout.tie(0);
21
22      int u, p;
23      cin >> u >> p;
24      while(u--){
25          string s;
26          cin >> s;
27          bool ok = 1;
28
29          for0(i, p){
30              char c = '#';
31              for(int j = 0; i+j < sz(s); j += p) {
32                  if(s[i+j] != '?')
33                      if(c == '#') c = s[i+j];
34                  else
35                      if(c != s[i+j]) ok = 0;
36              }
37          }
38
39          cout<< ((ok)? "YES\n" : "NO\n");
40      }
41  }
42  }

```

Пример программы-решения

Ниже представлено решение на языке Python3.

```

1  n, p = map(int, input().split())
2  lines = []
3  f = open('text.txt', 'w')
4  for i in range(n):
5      lines.append(input())
6  for i in range(n):
7      line = lines[i]
8      q = True
9      for j in range(p):
10         c = ''
11         k = j
12         while k < len(line):
13             if line[k] != '?':
14                 if c == '':
15                     c = line[k]
16             else:
17                 if line[k] != c:
18                     q = False
19                     break
20         k += p
21     if not q:
22         break
23
24

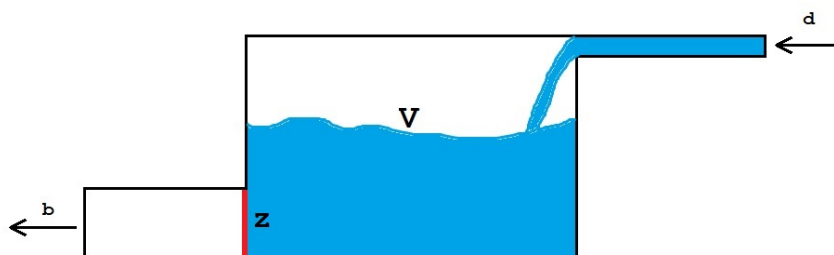
```

```

25     if q:
26         f.write("YES" + '\n')
27     else:
28         f.write("NO" + '\n')
29 f.close()

```

Задача I.1.1.4. Водопровод с резервной емкостью (20 баллов)



Рассмотрим следующую модель водопровода. Имеется резервная емкость объемом V литров. По входной трубе постоянно поступает вода со скоростью d литров в минуту. По выходной трубе жидкость выливается со скоростью b литров в минуту. Выполняется условие, что $d < b$. В начальный момент времени выход Z из емкости перекрыт. В тот момент, когда емкость наполняется, выход открывается и вода поступает на выпуск. Так как $d < b$, то в какой-то момент емкость полностью опустеет, и заслонка Z снова закрывается, после чего процесс повторяется снова.

Очевидно, что при этом в работе водопровода случаются паузы, в течение которых происходит заполнение резервной емкости. К ним относится и первая пауза, служащая для первичного наполнения емкости водой. По техническим причинам требуется, чтобы длина каждой такой паузы не превышала величины p . При этом нас интересует суммарное время t , в течение которого выходная труба была открыта. Количество пауз n при этом должно быть минимальным возможным.

В рассматриваемой модели используются сколь угодно малые доли времени и объема. Считается, что заслонка Z срабатывает мгновенно.

По заданным величинам d , b , t , p требуется найти такой целый объем V , при котором число пауз n было минимально возможным для заданного времени подачи воды t , а среди всех объемов, обеспечивающих такое время работы подачи воды и такое количество пауз найти самый маленький.

Формат входных данных

В первой строке содержится четыре целых числа d , b , t , p через пробел: $1 \leq d < b \leq 2 \cdot 10^9$, $1 \leq t \leq 15000$, $1 \leq p \leq 5000$.

Формат выходных данных

Вывести одно целое число — объем V резервной емкости, обеспечивающей необходимые ограничения на подачу воды. Если таких объемов несколько, вывести минимальный.

Пояснения к примеру

Рассмотрим первый пример из условия. Скорость подачи d равна 5 литров в минуту, скорость выпуска b равна 10 литров в минуту, требуется обеспечить суммарную работу выпуска в течение $t = 32$ минут, при этом максимальный размер любой паузы не должен превышать $p = 8$ минут.

Если мы установим объем резервной емкости 40 литров, то:

- ровно за 8 минут (что разрешено условием) она наполнится со скоростью 5 литров в минуту;
- далее пойдет процесс выпуска: 40 литров будут выпущены за 4 минуты со скоростью выпуска 10 литров в минуту, но за это время в емкость попадет $4 \cdot 5 = 20$ литров новой воды. Она будет выпущена за 2 минуты, но за это время поступит $2 \cdot 5 = 10$ литров новой воды, которая будет выпущена за 1 минуту и так далее. Так как доли времени и объема могут быть сколь угодно малыми, получим при подсчете времени работы выпуска следующую сумму: $4 + 2 + 1 + 0,5 + 0,25 + 0,125 + \dots$, которая в итоге равна 8. То есть через 8 минут емкость полностью опустеет, и заслонка закроется. Для того, чтобы время работы выпуска было равно 32 минуты, потребуется $32/8 = 4$ таких цикла, а значит и 4 паузы. За меньшее число пауз работу выпуска в течение 32 минут при существующих ограничениях обеспечить не получится.

Очевидно, что $V = 40$ литров — максимально возможный разрешенный объем, иначе первая же пауза будет больше допустимой. С другой стороны, если мы попробуем уменьшить объем до 39 литров, то получим:

- первоначальное заполнение будет произведено за $39/5 = 7.8$ минуты. Далее процесс выпуска будет работать $39/10 + 39/20 + 39/40 + \dots = 7,8$ минуты. Тогда за 4 цикла выпуск будет работать 31.2 минуты и для обеспечения 32 минут работы выпуска потребуется пятая пауза.

Примеры

Пример №1

Стандартный ввод
5 10 32 8
Стандартный вывод
40

Пример программы-решения

Ниже представлено решение на языке C++.

```

1 //Бинарный поиск
2 #include <bits/stdc++.h>
3
4 #define pb push_back
5 #define mp make_pair
6 #define x first
7 #define y second
8
9 #define all(x) (x).begin(), (x).end()
10 #define sz(a) (int)(a).size()

```

```

11 #define for0(i, n) for (int i = 0; i < (int)(n); i++)
12 #define for1(i, n) for (int i = 1; i <= (int)(n); i++)
13 #define int long long
14
15 using namespace std;
16 typedef pair<int, int> pii;
17 typedef long double ld;
18
19
20 signed main(){
21     ios::sync_with_stdio(0), cin.tie(0), cout.tie(0);
22
23     int d, b, t, p;
24
25     cin >> d >> b >> t >> p;
26
27     int y = t * (b-d);
28     int z = d * p;
29
30     int a = y / z + (y % z != 0);
31
32     int L = 0, R = d * p;
33
34     while(R - L > 1){
35         int M = (R + L) / 2;
36
37         if( M * a >= t * (b-d)) R = M; else L = M;
38
39     }
40
41     cout << R;
42 }

```

Задача I.1.1.5. Разбиения на различные множители (30 баллов)

Эта задача формулируется предельно кратко.

Дано натуральное число N . Требуется найти число способов представить его в виде произведения попарно различных множителей больших 1.

Формат входных данных

В первой строке содержится одно натуральное число $2 \leq N \leq 10^{12}$.

Формат выходных данных

Вывести одно число — количество способов представить число N в виде произведения попарно различных множителей больших 1.

Пояснения к примеру

Имеется 7 различных способов представить число 48 в виде произведения (в том числе и вырожденного) попарно различных множителей больших 1: 48, $2 \cdot 24$, $3 \cdot 16$,

$4 \cdot 12, 6 \cdot 8, 2 \cdot 3 \cdot 8, 2 \cdot 4 \cdot 6.$

Примеры

Пример №1

Стандартный ввод
48
Стандартный вывод
7

Пример программы-решения

Ниже представлено решение на языке C++.

```

1 //Динамическое программирование
2 #include <bits/stdc++.h>
3
4 #define pb push_back
5 #define mp make_pair
6 #define x first
7 #define y second
8
9 #define all(x) (x).begin(), (x).end()
10 #define sz(a) (int)(a).size()
11 #define for0(i, n) for (int i = 0; i < (int)(n); i++)
12 #define for1(i, n) for (int i = 1; i <= (int)(n); i++)
13 #define int long long
14
15 using namespace std;
16 typedef pair<int, int> pii;
17 typedef long double ld;
18
19 vector<pii> raz;
20 set<int> del;
21 vector<int> rdel;
22
23 signed main(){
24     ios::sync_with_stdio(0), cin.tie(0), cout.tie(0);
25
26     int n, dn;
27     cin >> n;
28     dn = n;
29
30     for(int i = 2; i <= 1e6+1 && dn > 1; i++) if(dn % i == 0){
31         int t = 0;
32         while(dn % i == 0){
33             t++;
34             dn /= i;
35         }
36         raz.pb({i, t});
37     }
38
39     if(dn > 1) raz.pb({dn, 1});
40
41     del.insert(1);
42
43     for(auto q : raz){

```

```

44     for0(i, q.y){
45         rdel.resize(0);
46         for(auto w : del) rdel.pb(w * q.x);
47         for(auto w : rdel) del.insert(w);
48     }
49 }
50
51 rdel.resize(0);
52 for(auto q : del) rdel.pb(q);
53 int r = sz(rdel);
54
55 map<int, int> rev;
56 for0(i, r) rev[rdel[i]] = i;
57
58 vector<vector<int> > dp(r, vector<int>(r, 0));
59
60 dp[0][0] = 1;
61 for(int i = 1; i < r; i++) dp[i][0] += dp[i-1][0];
62
63 for(int j = 1; j < r; j++){
64     for(int i = 1; i < r; i++) if(rdel[j] % rdel[i] == 0)
65         dp[i][j] = dp[i-1][rev[rdel[j] / rdel[i]]];
66     for(int i = 1; i < r; i++) dp[i][j] += dp[i-1][j];
67 }
68 }
69 cout<<dp[r-1][r-1];
70 }

```

Пример программы-решения

Ниже представлено решение на языке Python3.

```

1  from bisect import bisect_left
2
3  def binary_search(a, x):
4      i = bisect_left(a, x)
5      if i != len(a) and a[i] == x:
6          return i
7      else:
8          return -1
9
10 n = int(input())
11 dividers = []
12 for i in range(1, math.ceil(math.sqrt(n))):
13     if n % i == 0:
14         dividers.append(i)
15 divs = dividers.copy()
16 if int(math.sqrt(n)) ** 2 == n:
17     dividers.append(int(math.sqrt(n)))
18 for x in divs[::-1]:
19     dividers.append(n // x)
20 dp = [[0 for i in range(len(dividers))] for j in range(len(dividers))]
21 dp[0][0] = 1
22 for a in range(len(dividers) - 1):
23     pref_sum = 0
24     for i in range(len(dividers) - 1):
25         pref_sum += dp[a][i]
26         dp[a][i] = pref_sum
27     if dividers[a + 1] % dividers[i + 1] == 0:

```

```
28         bs = binary_search(dividers, dividers[a + 1] // dividers[i + 1])
29         if bs != -1:
30             dp[a + 1][i + 1] = dp[bs][i]
31     print(sum(dp[-1]))
```

Вторая попытка. Задачи 8–11 класса

Задача I.1.2.1. Распознавание деталей (15 баллов)

Ваша задача — написать программу для распознавания деталей заданного вида на конвейере. Изображение нужной детали вводится при начале распознавания. Далее вводится изображение части конвейера, на котором могут находиться детали различных видов. Требуется распознать и выделить все изображения заданной детали.

Формат входных данных

Сначала приведено изображение искомой детали. Оно имеет размер 5×5 . В 5 строках содержится по 5 символов «.» и «#», где решетки соответствуют детали, а точки — фону. Гарантируется, что изображение детали является 4-х связной фигурой. После изображения детали идет изображение текущего состояния конвейера. Это изображение имеет размер 10×20 . В последующих 10 строках содержится по 20 символов. Каждая деталь на конвейере 4-х связная и имеет свой цвет, обозначенный малой буквой латинского алфавита. Разные детали обозначены разными буквами. Таким образом изображение конвейера может содержать буквы от «a» до «z» и символ «.», по прежнему обозначающий фон. На конвейере находится не более 26 деталей. Следует учитывать, что требуемые детали могут быть повернуты на угол кратный 90 градусов и/или лежать обратной стороной.

Формат выходных данных

Требуется вывести изображение конвейера, на котором все вхождения заданной детали выделены соответствующими буквами, но переведенными в верхний регистр, остальные символы должны остаться без изменений.

Примеры

Пример №1

Стандартный ввод
<pre>##. ...## ...#.a.....aaac.....gg...a.ccc.....gg...b.c....d.....bbb...ddd.....b.....dhh.....f.....hh...e... ...ff.....h.eee...ff.....ee..mmmmmmmmmm </pre>
Стандартный вывод
<pre>A.....AAAC.....gg...A.CCC.....gg...B.C....d.....BBB...ddd.....B.....dHH.....F.....HH...e... ...FF.....H.eee...FF.....ee..mmmmmmmmmm </pre>

Пример программы-решения

Ниже представлено решение на языке C++.

```

1 //Сравнение с шаблоном, моделирование
2
3 #include <bits/stdc++.h>
4
5 #define pb push_back
6 #define mp make_pair
7 #define x first
8 #define y second
9
10 #define all(x) (x).begin(), (x).end()
11 #define sz(a) (int)(a).size()
12 #define for0(i, n) for (int i = 0; i < (int)(n); i++)
13 #define for1(i, n) for (int i = 1; i <= (int)(n); i++)
14 #define int long long
15
16 using namespace std;
17 typedef pair<int, int> pii;
18 typedef long double ld;
19 typedef vector<string> vs;

```



```

20
21 vs knv;
22 vector<vs > obr(8);
23 string s;
24 set<char> ans;
25
26 vs r_90(vs &a){
27     vs r = a;
28     for0(i, sz(a))
29         for0(j, sz(a[i])) r[j][sz(a) - i - 1] = a[i][j];
30     return r;
31 }
32
33 vs mirr(vs &a){
34     vs r = a;
35     for0(i, sz(r)) reverse(all(r[i]));
36     return r;
37 }
38
39 vs cut(int x, int y, char z){
40     vs r;
41     for0(i, 5) r.pb(knv[x+i].substr(y, 5));
42     for0(i, 5) for0(j, 5) if(r[i][j] != z) r[i][j] = '.'; else r[i][j] = '#';
43
44     return r;
45 }
46
47 int toko(vs &a, char z){
48     int r = 0;
49     for(auto q : a) for(auto w : q) r += (w == z);
50     return r;
51 }
52
53 signed main(){
54     ios::sync_with_stdio(0), cin.tie(0), cout.tie(0);
55
56     for0(i, 5){
57         cin >> s;
58         obr[0].pb(s);
59     }
60
61     for1(i, 3) obr[i] = r_90(obr[i-1]);
62
63     obr[4] = mirr(obr[0]);
64
65     for(int i = 5; i < 8; i++) obr[i] = r_90(obr[i-1]);
66
67     int ko = toko(obr[0], '#');
68
69     string z;
70     for0(i, 30) z += ".";
71     for0(i, 5) knv.pb(z);
72     for0(i, 10) {
73         cin >> s;
74         for0(j, 5) s = "." + s;
75         for0(j, 5) s += ".";
76         knv.pb(s);
77     }
78     for0(i, 5) knv.pb(z);
79

```

```

80 map<char, int> msk;
81 for(auto q : knv)
82     for(auto w : q) msk[w]++;
83
84 for(auto q : msk) if(q.y == ko){
85     bool ok = 0;
86     for0(i, 14) for0(j, 24) {
87         vs d = cut(i, j, q.x);
88
89         if(toko(d, '#') == ko){
90             bool ok1;
91             for0(j, 8){
92                 if(d == obr[j]){
93                     ans.insert(q.x);
94                 }
95             }
96         }
97     }
98 }
99
100 for(int i = 5; i < 15; i++){
101     for(int j = 5; j < 25; j++)
102         if(ans.find(knv[i][j]) != ans.end()) cout<<(char)(knv[i][j] - 'a' + 'A');
103     else cout<<knv[i][j];
104     cout<<endl;
105 }
106 }

```

Задача I.1.2.2. Мерцающие звезды (22 баллов)

Современных звездных путешественников очень трудно удивить. Однако фирма Amazing Star Travel хочет предложить нечто новое: наблюдения за мерцающими звездами. Это очень эффектное явление, возникающее в тот момент, когда мощную звезду заслоняет планета. Для этого разработан маршрут между двумя точками A и B . Специалисты фирмы выделили N наиболее ярких звезд в видимой части космоса и отметили M крупных планет. Осталось подсчитать, сколько раз за время путешествия по отрезку AB путешественники насладятся видом мерцающей звезды.

Формат входных данных

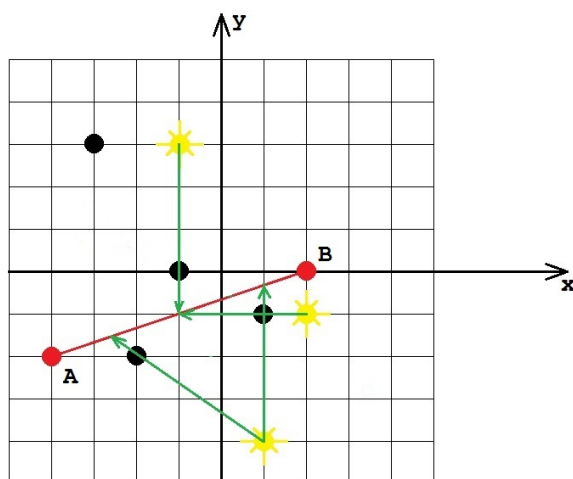
В первой строке содержится четыре целых числа через пробел XA, YA, XB, YB — координаты точек A и B . Во второй строке содержатся числа N и M , разделенные пробелом ($0 \leq N, M \leq 100$) — количество звезд и количество планет соответственно. В каждой из следующих N строк содержатся координаты очередной звезды. Далее в каждой из следующих M строк содержатся координаты очередной планеты. Все координаты целые, по модулю не превосходят 1000. Гарантируется, что никакие три точки из всех вышперечисленных не находятся на одной прямой.

Формат выходных данных

В ответе нужно выдать одно число — количество случаев, когда при движении по отрезку из точки A в точку B какая-либо звезда будет заслонена от наблюдателя планетой. Если какие-либо две звезды мерцают одновременно, то это считается как

два независимых случая. Все упомянутые объекты считаем материальными точками, для упрощения вычислений все рассматриваем на плоскости. Помимо этого, согласно теории относительности, путешествие с точки зрения внешнего наблюдателя, совершается мгновенно, то есть положение звезд и планет за время путешествия не изменяется, однако для путешественников оно достаточно длительное, чтобы насладиться захватывающими видами.

Пояснения к примеру



Примеры

Пример №1

Стандартный ввод
-4 -2 2 0
3 4
-1 3
2 -1
1 -4
-3 3
-1 0
-2 -2
1 -1
Стандартный вывод
4

Пример программы-решения

Ниже представлено решение на языке C++.

```

1 //Целочисленно, сравнение площадей
2 #include <bits/stdc++.h>
3
4 #define pb push_back
5 #define mp make_pair
6 #define x first

```

```

7  #define y second
8
9  #define all(x) (x).begin(), (x).end()
10 #define sz(a) (int)(a).size()
11 #define for0(i, n) for (int i = 0; i < (int)(n); i++)
12 #define for1(i, n) for (int i = 1; i <= (int)(n); i++)
13 #define int long long
14
15 using namespace std;
16 typedef long double ld;
17 typedef pair<int, int> pii;
18
19 int sqrt2(pii a, pii b, pii c){
20     pii v1 = {b.x - a.x, b.y - a.y};
21     pii v2 = {c.x - a.x, c.y - a.y};
22
23     return abs(v1.x*v2.y - v1.y*v2.x);
24 }
25
26 signed main(){
27     ios::sync_with_stdio(0), cin.tie(0), cout.tie(0);
28
29     pii A, B;
30     cin >> A.x >> A.y >> B.x >> B.y;
31
32     int n, m;
33     cin >> n >> m;
34     vector<pii> s(n), p(m);
35     for0(i, n) cin >> s[i].x >> s[i].y;
36     for0(i, m) cin >> p[i].x >> p[i].y;
37
38     int ans = 0;
39
40     for0(i, n){
41         int S = sqrt2(A, B, s[i]);
42         for0(j, m){
43             int s1 = sqrt2(A, B, p[j]);
44             int s2 = sqrt2(A, p[j], s[i]);
45             int s3 = sqrt2(B, p[j], s[i]);
46             if(s1 + s2 + s3 == S) ans++;
47         }
48     }
49     cout << ans;
50 }

```

Пример программы-решения

Ниже представлено решение на языке C++.

```

1  //вещественно, бинпоиском точку пересечения
2  #include <bits/stdc++.h>
3
4  #define pb push_back
5  #define mp make_pair
6  #define x first
7  #define y second
8
9  #define all(x) (x).begin(), (x).end()
10 #define sz(a) (int)(a).size()

```

```

11  #define for0(i, n) for (int i = 0; i < (int)(n); i++)
12  #define for1(i, n) for (int i = 1; i <= (int)(n); i++)
13  #define int long long
14
15  using namespace std;
16  typedef long double ld;
17  typedef pair<ld, ld> pld;
18
19  ld eps = 1e-7;
20
21  struct line{
22      ld A, B, C;
23  };
24
25  ld dist(pld a, pld b){
26      return sqrt((a.x-b.x)*(a.x-b.x)+(a.y-b.y)*(a.y-b.y));
27  }
28
29  line L(pld a1, pld a2){
30      ld A = a2.y - a1.y;
31      ld B = a1.x - a2.x;
32      ld C = a2.x*a1.y - a1.x*a2.y;
33
34      line r = {A, B, C};
35      return r;
36  }
37
38  int sgn(ld x){
39      if(x < -eps) return -1;
40      if(x > eps) return 1;
41      return 0;
42  }
43
44  pld tp(pld A, pld B, pld s, pld p){
45      line AB = L(A, B);
46
47      ld L = 0, R = 1e6;
48
49      pld v = {p.x - s.x, p.y - s.y};
50      pld ab = {B.x - A.x, B.y - A.y};
51      ld Fs = AB.A * (s.x + L * ab.x) + AB.B * (s.y + L * ab.y) + AB.C;
52
53      if(abs(v.x*ab.y - v.y*ab.x) < eps) return{1e6, 1e6};
54
55      while(R - L > eps){
56          ld M = (L+R)/2.0;
57
58          pld m = {s.x + M*v.x, s.y + M*v.y};
59
60          ld F = AB.A * m.x + AB.B * m.y + AB.C;
61
62          if(sgn(F) == sgn(Fs)) L = M; else R = M;
63      }
64
65      return {s.x + L*v.x, s.y + L*v.y};
66  }
67
68  signed main(){
69      ios::sync_with_stdio(0), cin.tie(0), cout.tie(0);
70

```

```

71     pld A, B;
72     cin >> A.x >> A.y >> B.x >> B.y;
73
74     int n, m;
75     cin >> n >> m;
76     vector<pld> s(n), p(m);
77     for0(i, n) cin >> s[i].x >> s[i].y;
78     for0(i, m) cin >> p[i].x >> p[i].y;
79
80     int ans = 0;
81
82     for0(i, n)
83     for0(j, m){
84         line AB = L(A, B);
85         ld tos = AB.A * s[i].x + AB.B * s[i].y + AB.C;
86         ld top = AB.A * p[j].x + AB.B * p[j].y + AB.C;
87
88         if(sgn(tos) == sgn(top)){
89
90             pld w = tp(A, B, s[i], p[j]);
91
92             if(abs(w.x) < 1e6 && abs(w.y) < 1e6){
93
94                 if(abs(dist(A, w) + dist(w, B) - dist(A, B)) < eps) {
95                     ans++;
96                 }
97             }
98
99         }
100     }
101
102     cout << ans;
103 }

```

Задача I.1.2.3. Послание внеземного разума 2 (23 баллов)

Профессор Персиков снова получил послание внеземного разума. Он по-прежнему считает, что доказательством этого является его периодичность. При этом период должен быть равен «константе Персикова» — числу P . К сожалению, послание не совсем периодичное. Если сказать более точно, то оно совсем не периодичное. Однако, это никак не останавливает исследователя космических глубин. Он говорит, что некоторые сигналы были неправильно откалиброваны, отфильтрованы и интерпретированы. По-прежнему мы будем считать, что все сигналы отображаются малыми буквами латинского алфавита, но в данной задаче все они распознаны и знаков вопроса нет. Тем не менее профессор, согласно своей теории, может заявить, что все вхождения такой-то буквы интерпретированы неверно и их **все** следует заменить на вхождения какой-то другой (одной и той же) буквы. Более строго: пусть на позициях $p_{i_1}, p_{i_2}, \dots, p_{i_k}$ и только на них в последовательности находится одна и та же буква. Профессор может выбрать любую другую букву (как встречающуюся в слове, так и не встречающуюся) и поставить ее во всех этих позициях. Например, в слове *qqzbbacabadaba* он может выбрать все вхождения буквы *b* и заменить их на букву *a*, получив слово *qqzaaacaadaaaa* (это считается одной заменой, независимо от числа вхождений). Очевидно, что таким образом любое послание можно сделать P -периодическим, но профессор заинтересован сделать как можно меньше таких замен. При этом он хочет получить лексикографически минимальное послание.

Формат входных данных

В первой строке содержится число P — константа Персикова ($1 \leq P \leq 10^5$). В следующей строке содержится непустая последовательность, состоящая из малых букв латиницы. Длина этой строки не превосходит $2 \cdot 10^2$.

Формат выходных данных

Вывести строку, которая получается из исходной путем минимального числа операций замены вхождений всех букв одного вида на вхождения какой-то (одной и той же) другой буквы. Итоговая строка должна быть P -периодической, то есть любые две ее буквы, расстояние между которыми кратно P должны совпадать. Среди всех таких строк вывести лексикографически минимальную (первую в алфавитном порядке).

Примеры

Пример №1

Стандартный ввод
4 qqzbbacabadaba
Стандартный вывод
аасааасааасаа

Пример программы-решения

Ниже представлено решение на языке C++.

```

1 //граф на буквах, обход в глубину
2 #include <bits/stdc++.h>
3
4 #define pb push_back
5 #define mp make_pair
6 #define x first
7 #define y second
8
9 #define all(x) (x).begin(), (x).end()
10 #define sz(a) (int)(a).size()
11 #define for0(i, n) for (int i = 0; i < (int)(n); i++)
12 #define for1(i, n) for (int i = 1; i <= (int)(n); i++)
13 #define int long long
14
15 using namespace std;
16 typedef long double ld;
17 typedef pair<int, int> pii;
18
19 map<char, set<char> > G;
20 map<char, int> rzb;
21 int num = 0;
22 vector<set<char> > toans;
23 set<char> emp;
24
25 void dfs(char a){
26     toans[rzb[a]].insert(a);

```

```

27     for(auto to : G[a]) if(rzb[to] == 0) {
28         rzb[to] = rzb[a];
29         dfs(to);
30     }
31 }
32
33 signed main(){
34     ios::sync_with_stdio(0), cin.tie(0), cout.tie(0);
35
36     int p;
37     cin >> p;
38     string s;
39     cin >> s;
40
41     vector<set<char> > op(p);
42
43     for0(i, sz(s)) op[i%p].insert(s[i]);
44
45     for0(i, p)
46         for(auto q : op[i])
47             for(auto w : op[i]) {
48                 G[q].insert(w);
49                 G[w].insert(q);
50             }
51
52     toans.pb(emp);
53     for(auto q : G)
54         if(rzb[q.x] == 0){
55             num++;
56             toans.pb(emp);
57             rzb[q.x] = num;
58             dfs(q.x);
59         }
60
61     for0(i, sz(s)) s[i] = *(toans[rzb[s[i]]].begin());
62     cout<<s;
63 }

```

Задача I.1.2.4. Проект «Ровные дороги» (10 баллов)

При проектировании новой автодороги было принято решение сделать ее не более чем из двух абсолютно горизонтальных участков. Будущую трассу разбили на n равных по длине малых отрезков. Будем считать, что в пределах одного малого отрезка местность имеет одну и ту же высоту h_i . При этом в целях эффективной трансформации местности требуется для выравнивания использовать исключительно грунт с этой же трассы. Это означает, что можно с некоторого малого отрезка высоты h_i взять некоторое количество грунта d так, что высота этого участка станет $h_i - d > 0$. Далее эти d единиц грунта обязательно нужно поместить на другой малый отрезок высоты h_j так, что его высота станет $h_j + d$. Перемещать грунт можно только в пределах одного из двух выбранных участков, то есть отрезки номер i и номер j должны принадлежать одному и тому же горизонтальному после выравнивания участку. Технические требования таковы, что эти два выровненных участка должны иметь целочисленную высоту над уровнем моря. Таким образом, для начала требуется определить сколькими способами можно переместить грунт с возвышенностей в низины так, что образуется не более чем два горизонтальных участка трассы, каждый из которых имеет целочисленную высоту.

Формат входных данных

В первой строке содержится число n — количество малых отрезков, на которые разбили трассу, $2 \leq n \leq 10^2$. Во второй строке указаны высоты h_i этих отрезков через пробел в порядке слева направо, $1 \leq h_i \leq 2 \cdot 10^4$.

Формат выходных данных

Вывести количество способов достичь требуемых показателей. Если у двух разбиений трассы в результате выравнивания конечные результаты совпадают, то они считаются одинаковыми.

Пояснения к примеру*Пример №1*

Можно указать следующие разбиения не более чем на два участка, после выравнивания в пределах которых образуются целые высоты: [10 4] и [2 7 5 8 6 6 15], результат выравнивания: [7 7] и [7 7 7 7 7 7]

[10 4 2 7 5 8] и [6 6 15], результат выравнивания: [6 6 6 6 6] и [9 9 9]

[10 4 2 7 5 8 6 6] и [15], результат выравнивания: [6 6 6 6 6 6 6] и [15]

[10 4 2 7 5 8 6 6 15], результат выравнивания: [7 7 7 7 7 7 7 7].

Результаты выравнивания в первом и последнем способе совпадают, таким образом общий ответ равен 3.

Пример №2

Во втором примере получатся следующие варианты разбиения и выравнивания: [3] [5 2 7 6 4 5 8 1 7], результат выравнивания: [3] и [5 5 5 5 5 5 5 5]

[3 5] [2 7 6 4 5 8 1 7], результат выравнивания: [4 4] и [5 5 5 5 5 5 5]

[3 5 2 7 6 4 5 8] [1 7], результат выравнивания: [5 5 5 5 5 5 5] и [4 4]

Примеры*Пример №1*

Стандартный ввод
9
10 4 2 7 5 8 6 6 15
Стандартный вывод
3

Пример №2

Стандартный ввод
10
3 5 2 7 6 4 5 8 1 7
Стандартный вывод
3


```

57     }
58     cout << ans;
59 }

```

Задача I.1.2.5. Прогулка по мостам (30 баллов)

Возможно, вы знаете историю про то, как Эйлер гулял по мостам Кенигсберга. Допустим теперь, что Эйлер попал на некий архипелаг, между некоторыми островами которого имеются мосты. Мосты построены таким образом, что между любыми двумя островами архипелага можно построить путь по этим мостам, причем этот путь единственный. Все острова пронумерованы от 1 до n . Эйлер находится на острове номер 1 и желает совершить следующую прогулку: с острова номер 1 он хочет дойти до острова номер 2, далее до острова номер 3 и так далее до острова номер n . С него он хочет вернуться на остров номер 1. Он прекрасно понимает, что при этом ему придется посещать некоторые мосты по многу раз. Осталось выяснить, сколько раз суммарно он пройдет по мостам во время своей прогулки.

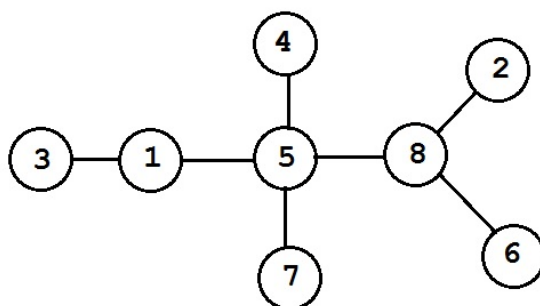
Формат входных данных

В первой строке содержится число n — количество островов в архипелаге, $2 \leq n \leq 10^5$. В следующих $n - 1$ строках содержатся по два числа a_i и b_i через пробел — номера островов, соединенных мостом, $1 \leq a_i, b_i \leq n, a_i \neq b_i$. Гарантируется, что между любыми двумя островами архипелага можно построить путь по этим мостам, причем этот путь единственный.

Формат выходных данных

Вывести длину описанного пути, то есть суммарное количество пересечений мостов, которые придется совершить.

Пояснения к примеру



На рисунке вы видите расположение мостов в архипелаге из примера. Прогулка Эйлера будет проходить следующим образом:

1 — 5 — 8 — 2 : 3 моста;

2 — 8 — 5 — 1 — 3 : 4 моста;

3 — 1 — 5 — 4 : 3 моста;

4 — 5 : 1 мост;

5 — 8 — 6 : 2 моста;

6 — 8 — 5 — 7 : 3 моста;

7 — 5 — 8 : 2 моста;

8 — 5 — 1 : 2 моста.

Итого он пройдет через $3 + 4 + 3 + 1 + 2 + 3 + 2 + 2 = 20$ мостов.

Примеры

Пример №1

Стандартный ввод
8
5 8
1 3
8 6
7 5
2 8
1 5
4 5
Стандартный вывод
20

Пример программы-решения

Ниже представлено решение на языке C++.

```

1 //lca двоичным подъемом, обход в глубину
2 #include <bits/stdc++.h>
3
4 #define pb push_back
5 #define mp make_pair
6 #define x first
7 #define y second
8
9 #define all(x) (x).begin(), (x).end()
10 #define sz(a) (int)(a).size()
11 #define for0(i, n) for (int i = 0; i < (int)(n); i++)
12 #define for1(i, n) for (int i = 1; i <= (int)(n); i++)
13 #define int long long
14
15 using namespace std;
16 typedef long double ld;
17 typedef pair<int, int> pii;
18
19 int n, a, b, timer;
20 vector<vector<int>> > G;
21 const int N = 1e5 + 228;
22 const int LG = 18;
23 vector<int> dep, in, out;
```

```

24
25 int up[LG][N];
26
27 void dfs(int a, int p) {
28     in[a] = timer++;
29     up[0][a] = p;
30     for (int i = 1; i < LG; i++) {
31         up[i][a] = up[i - 1][up[i - 1][a]];
32     }
33     for (int to : G[a]) {
34         if (to != p) {
35             dep[to] = dep[a] + 1;
36             dfs(to, a);
37         }
38     }
39     out[a] = timer;
40 }
41
42 bool upper(int u, int v) {
43     return in[u] <= in[v] && out[v] <= out[u];
44 }
45
46 int lca(int u, int v) {
47     if (in[u] > in[v]) {
48         swap(u, v);
49     }
50     if (upper(u, v)) {
51         return u;
52     }
53     for (int i = LG - 1; i >= 0; i--) {
54         if (!upper(up[i][u], v)) {
55             u = up[i][u];
56         }
57     }
58     return up[0][u];
59 }
60
61 int dist(int u, int v) {
62     int t = lca(u, v);
63     return dep[u] + dep[v] - 2 * dep[t];
64 }
65
66 signed main(){
67     ios::sync_with_stdio(0), cin.tie(0), cout.tie(0);
68
69     cin >> n;
70     G.resize(n);
71     for0(i, n-1){
72         cin >> a >> b;
73         a--;
74         b--;
75         G[a].pb(b);
76         G[b].pb(a);
77     }
78     dep.resize(n, 0);
79     in.resize(n);
80     out.resize(n);
81     dfs(0, 0);
82
83     int ans = 0;

```

```

84     for0(i, n){
85         ans += dist(i, (i+1)%n);
86     }
87
88     cout<<ans;
89 }

```

Пример программы-решения

Ниже представлено решение на языке C++.

```

1  // lca двоичным подъемом, обход в ширину без глубокой рекурсии
2  #include <bits/stdc++.h>
3
4  #define pb push_back
5  #define mp make_pair
6  #define x first
7  #define y second
8
9  #define all(x) (x).begin(), (x).end()
10 #define sz(a) (int)(a).size()
11 #define for0(i, n) for (int i = 0; i < (int)(n); i++)
12 #define for1(i, n) for (int i = 1; i <= (int)(n); i++)
13 #define int long long
14
15 using namespace std;
16 typedef long double ld;
17 typedef pair<int, int> pii;
18
19 int n, a, b, timer;
20 vector<vector<int> > G;
21 const int N = 1e5 + 228;
22 const int LG = 18;
23 vector<int> dep, in, out;
24
25 int up[LG][N];
26
27 bool upper(int u, int v) {
28     return in[u] <= in[v] && out[v] <= out[u];
29 }
30
31 int lca(int u, int v) {
32     if (in[u] > in[v]) {
33         swap(u, v);
34     }
35     if (upper(u, v)) {
36         return u;
37     }
38     for (int i = LG - 1; i >= 0; i--) {
39         if (!upper(up[i][u], v)) {
40             u = up[i][u];
41         }
42     }
43
44     return up[0][u];
45 }
46
47 int dist(int u, int v) {
48     int t = lca(u, v);

```

```

49         return dep[u] + dep[v] - 2 * dep[t];
50     }
51
52     signed main(){
53         ios::sync_with_stdio(0), cin.tie(0), cout.tie(0);
54
55         cin >> n;
56         G.resize(n);
57         for0(i, n-1){
58             cin >> a >> b;
59             a--;
60             b--;
61             G[a].pb(b);
62             G[b].pb(a);
63         }
64         dep.resize(n, 0);
65         in.resize(n);
66         out.resize(n, 0);
67
68         deque<int> och;
69         vector<int> par(n), d(n, 1), oo;
70         och.pb(0);
71         oo.pb(0);
72         dep[0] = 0;
73         par[0] = 0;
74         while(sz(och) > 0){
75             int tb = och[0];
76             och.pop_front();
77             for(auto to : G[tb]) if(to != par[tb]){
78                 dep[to] = dep[tb] + 1;
79                 par[to] = tb;
80
81                 up[0][to] = tb;
82                 for (int i = 1; i < LG; i++) {
83                     up[i][to] = up[i - 1][up[i - 1][to]];
84                 }
85                 och.pb(to);
86                 oo.pb(to);
87             }
88
89         }
90         for(int i = sz(oo)-1; i >= 1; i--){
91             d[par[oo[i]]] += d[oo[i]];
92
93         in[0] = 0;
94         for(int i = 1; i < sz(oo); i++){
95             in[oo[i]] = out[par[oo[i]]]+1;
96             out[oo[i]] = in[oo[i]] + 1;
97             out[par[oo[i]]] += 2*d[oo[i]];
98         }
99
100        int ans = 0;
101        for0(i, n){
102            ans += dist(i, (i+1)%n);
103        }
104        cout<<ans;
105    }

```

Третья попытка. Задачи 8–11 класса

Задача I.1.3.1. Послание Аресибо (15 баллов)

Имеется прямоугольное изображение, разбитое на единичные квадратики, размер этого изображения $n \times m$, ($5 \leq n, m$). Каждый его квадратик либо черный либо белый. Известно, что на этом изображении нарисована черным цветом на белом фоне одна четырехсвязная фигура. Фигура называется четырехсвязной, если между любыми двумя ее клетками можно построить путь по клеткам этой фигуры, в котором любые две рядом стоящие клетки являются соседними в изображении либо по горизонтали либо по вертикали. Далее изображение разбили на строки и соединили их в одну большую строку без пробелов и разделителей. Длина этой строки $n \cdot m$. После этого ее отправили в направлении шарового звездного скопления М13, находящегося на расстоянии 25000 световых лет в созвездии Геркулеса. Вы обитатель М13 и перед вами поставили задачу восстановить изображение, исходя из информации о его четырехсвязности. Гарантируется, что решение единственно.

Формат входных данных

На вход подается принятая строка. Она состоит из знаков «.» и «#». Точки соответствуют фону, а решетки — изображенной фигуре. Длина строки не превосходит 5184.

Формат выходных данных

Требуется вывести изображение в исходном виде, то есть разбить его на строки одинаковой длины и расположить их друг под другом.

Примеры

Пример №1

Стандартный ввод
.....###...#.#...#...###...#...#...#####...###.#. #####.#.#.#.#.##.##..
Стандартный вывод
..... ...###... #.#.#... #..###... #...#... ##### ..####..# ..####..# ..#.#...# ##..##..

Пример программы-решения

Ниже представлено решение на языке C++.


```

1 //dfs, нахождение 4-связной области
2
3 #include <bits/stdc++.h>
4
5 #define pb push_back
6 #define mp make_pair
7 #define x first
8 #define y second
9
10 #define all(x) (x).begin(), (x).end()
11 #define sz(a) (int)(a).size()
12 #define for0(i, n) for (int i = 0; i < (int)(n); i++)
13 #define for1(i, n) for (int i = 1; i <= (int)(n); i++)
14 #define int long long
15
16 using namespace std;
17 typedef pair<int, int> pii;
18 typedef long double ld;
19 typedef vector<string> vs;
20
21 int k;
22 int dx[4] = {-1, 0, 1, 0};
23 int dy[4] = {0, 1, 0, -1};
24
25 bool in_tr(int x, int y, int h, int w){
26     return (x >= 0 && y >= 0 && x < h && y < w);
27 }
28
29 void dfs(vector<string> &v, int x, int y){
30     v[x][y] = '#';
31     k++;
32     for0(i, 4){
33         int nx = x + dx[i];
34         int ny = y + dy[i];
35
36         if(in_tr(nx, ny, sz(v), sz(v[0])) && v[nx][ny] == '$')
37             dfs(v, nx, ny);
38     }
39 }
40
41 signed main(){
42     ios::sync_with_stdio(0), cin.tie(0), cout.tie(0);
43
44     string iss;
45     cin >> iss;
46
47     int n = sz(iss);
48     for0(i, n) if(iss[i] == '#') iss[i] = '$';
49     int kk = 0;
50     for(auto q : iss)
51         kk += (q == '$');
52
53     for(int i = 5; i < n; i++)
54         if(n % i == 0 && n/i >= 5){
55             string s = iss;
56             vector<string> v;
57             for0(j, n/i){
58                 v.pb(s.substr(0, i));
59                 if(sz(s) > i) s = s.substr(i);
60             }

```

```

61
62     k = 0;
63     int sx, sy;
64     for0(i, sz(v))
65         for0(j, sz(v[i]))
66             if(v[i][j] == '$'){
67                 sx = i;
68                 sy = j;
69             }
70
71     dfs(v, sx, sy);
72
73     if(k == kk)
74         for0(i, sz(v)) cout<<v[i]<<endl;
75 }
76 }

```

Задача I.1.3.2. Гравитационный параллакс (10 баллов)

Специалисты фирмы Amazing Star Travel очень ответственно относятся к безопасности своих клиентов. Прежде чем маршрут из точки A в точку B будет одобрен для увлекательных путешествий, он проходит всестороннюю экспертизу. Помимо очевидных опасностей, таких как наличие на маршруте вредных космических излучений, активность пиратов и прогнозирования вспышек сверхновых, в числе прочих анализируется и множество других, более скучных. Современные исследования показали, что на пролетающий на околосветовых скоростях по отрезку AB космический корабль негативное влияние может оказать гравитационное поле звезд и планет. С удалением звезды или планеты от отрезка, это влияние сначала возрастает, а потом скачкообразно падает до нуля. Особенно опасно, если звезда и планета расположены по разные стороны прямой AB . Тогда их воздействие мультиплицируется, то есть перемножается. Особо следует отметить, что эффект мультиплицирования возникает только в паре звезда-планета, а, например, для пар звезда-звезда или планета-планета не обнаружен. Кроме того, этот эффект не наблюдается, если звезда и планета расположены по одну сторону от прямой AB . Осталось сказать, что можно считать эффект гравитационного воздействия равным площади треугольника ABC , где A и B — начало и конец маршрута, а C — место расположения звезды или планеты. Вас просят оценить опасность заданного отрезка AB , то есть найти максимальную величину гравитационного воздействия на отрезок среди всех пар звезда-планета, расположенных по разные стороны прямой AB . Не стоит забывать и о простом воздействии отдельных объектов, которое может оказаться даже больше, чем воздействие пары.

Формат входных данных

В первой строке содержится четыре целых числа через пробел XA, YA, XB, YB — координаты точек A и B . Во второй строке содержатся числа N и M , разделенные пробелом ($0 \leq N, M \leq 100$) — количество звезд и количество планет соответственно. В каждой из следующих N строк содержатся координаты очередной звезды. Далее в каждой из следующих M строк содержатся координаты очередной планеты. Указанные звезды и планеты, и только они могут оказать ненулевое гравитационное влияние на отрезок AB . Все координаты целые, по модулю не превосходят 1000.

Гарантируется, что никакие три точки из всех вышеперечисленных не находятся на одной прямой.

Формат выходных данных

В ответе нужно выдать ответ на задачу — одно вещественное число округленное ровно до двух знаков после точки. Разделитель между целой и дробной частями ответа — точка.

Пояснения к примеру

Итоговое наибольшее воздействие на отрезок AB оказывают выделенные треугольниками звезда и планета. Воздействие звезды равно площади треугольника ABC , которая равна 11, воздействие планеты равно площади треугольника ABD , которая равна 14. Так как они находятся по разные стороны прямой AB , то их общее воздействие равно произведению $11 \cdot 14 = 154,00$

Примеры

Пример №1

Стандартный ввод
-4 -2 2 0
3 4
-2 4
2 -1
1 -4
-3 3
-1 0
-2 -2
1 -1
Стандартный вывод
154.00

Пример программы-решения

Ниже представлено решение на языке C++.

```

1 //разделение - знаки векторного произведения, площадь - модуль векторного
2 // произведения
3 #include <bits/stdc++.h>
4
5 #define pb push_back
6 #define mp make_pair
7 #define x first
8 #define y second
9
10 #define all(x) (x).begin(), (x).end()
11 #define sz(a) (int)(a).size()
12 #define for0(i, n) for (int i = 0; i < (int)(n); i++)
13 #define for1(i, n) for (int i = 1; i <= (int)(n); i++)
14 #define int long long
15

```

```
16 using namespace std;
17 typedef long double ld;
18 typedef pair<int, int> pii;
19
20 int vprod(pii a, pii b){
21     return a.x*b.y - b.x*a.y;
22 }
23
24 int sgn_vprod(pii a, pii b){
25     int vp = vprod(a, b);
26     if(vp < 0) return -1;
27     if(vp > 0) return 1;
28     return 0;
29 }
30
31 int sqt2(pii a, pii b, pii c){
32     pii v1 = {b.x - a.x, b.y - a.y};
33     pii v2 = {c.x - a.x, c.y - a.y};
34
35     return abs(vprod(v1, v2));
36 }
37
38 signed main(){
39     ios::sync_with_stdio(0), cin.tie(0), cout.tie(0);
40
41     pii A, B;
42     cin >> A.x >> A.y >> B.x >> B.y;
43     pii AB = {B.x - A.x, B.y - A.y};
44
45     int n, m;
46     cin >> n >> m;
47     vector<pii> s(n), p(m);
48     for0(i, n) cin >> s[i].x >> s[i].y;
49     for0(i, m) cin >> p[i].x >> p[i].y;
50
51     double ans = 0;
52
53     for0(i, n)
54         ans = max(ans, sqt2(A, B, s[i]) / 2.0);
55
56     for0(j, m)
57         ans = max(ans, sqt2(A, B, p[j]) / 2.0);
58
59     for0(i, n){
60         pii Bs = {s[i].x - B.x, s[i].y - B.y};
61
62         for0(j, m){
63             pii Bp = {p[j].x - B.x, p[j].y - B.y};
64
65             if(sgn_vprod(AB, Bs) != sgn_vprod(AB, Bp)){
66                 ans = max(ans, sqt2(A, B, s[i]) * sqt2(A, B, p[j]) / 4.0);
67             }
68         }
69     }
70
71     cout.precision(2);
72     cout<<fixed<<ans;
73 }
```

Пример программы-решения

Ниже представлено решение на языке C++.

```

1 //разделение - знаки подстановки в общее уравнение прямой, площадь --- формула
2 // Герона
3 #include <bits/stdc++.h>
4
5 #define pb push_back
6 #define mp make_pair
7 #define x first
8 #define y second
9
10 #define all(x) (x).begin(), (x).end()
11 #define sz(a) (int)(a).size()
12 #define for0(i, n) for (int i = 0; i < (int)(n); i++)
13 #define for1(i, n) for (int i = 1; i <= (int)(n); i++)
14 #define int long long
15
16 using namespace std;
17 typedef long double ld;
18 typedef pair<double, double> pii;
19 pii A, B;
20
21 struct line{
22     double A, B, C;
23 };
24
25 line L(pii a, pii b){
26     line r;
27     r.A = b.y - a.y;
28     r.B = a.x - b.x;
29     r.C = b.x*a.y - a.x*b.y;
30
31     return r;
32 }
33
34 double dist(pii a, pii b){
35     return sqrt((a.x - b.x)*(a.x - b.x)+(a.y - b.y)*(a.y - b.y));
36 }
37
38
39 double sqt2(pii a, pii b, pii c){
40     double p = (dist(a, b) + dist(b, c) + dist(a,c)) / 2.0;
41     return 2*(sqrt(p * (p-dist(a,b)) * (p-dist(b,c)) * (p-dist(a,c)) ));
42 }
43
44 signed main(){
45     ios::sync_with_stdio(0), cin.tie(0), cout.tie(0);
46
47     cin >> A.x >> A.y >> B.x >> B.y;
48
49     int n, m;
50     cin >> n >> m;
51     vector<pii> s(n), p(m);
52     for0(i, n) cin >> s[i].x >> s[i].y;
53     for0(i, m) cin >> p[i].x >> p[i].y;
54
55     double ans = 0;
56     line AB = L(A, B);

```

```

57
58     for0(i, n)
59         ans = max(ans, sqrt2(A, B, s[i]) / 2.0);
60
61     for0(j, m)
62         ans = max(ans, sqrt2(A, B, p[j]) / 2.0);
63
64     for0(i, n){
65         pii Bs = {s[i].x - B.x, s[i].y - B.y};
66         double stoAB = AB.A*s[i].x + AB.B*s[i].y + AB.C;
67
68         for0(j, m){
69             pii Bp = {p[j].x - B.x, p[j].y - B.y};
70
71             double ptoAB = AB.A*p[j].x + AB.B*p[j].y + AB.C;
72
73
74             if(stoAB < 0 && ptoAB > 0 || stoAB > 0 && ptoAB < 0){
75                 ans = max(ans, sqrt2(A, B, s[i]) * sqrt2(A, B, p[j]) / 4.0);
76             }
77         }
78     }
79
80     cout.precision(2);
81     cout<<fixed<<ans;
82 }

```

Задача I.1.3.3. Послание внеземного разума 3 (25 баллов)

Профессор Персиков снова на первых полосах новостных агрегаторов! Он сделал очередное гениальное предположение по поводу новой последовательности сигналов из глубин космоса.

- во-первых, говорит профессор, для подтверждения искусственности происхождения сигнала достаточно, чтобы он был периодическим с периодом, **не превосходящим известную** «константу Персикова» P .
- во-вторых, согласно закону диффузного рассеивания информации в некогерентных пространствах при нелинейно возрастающем коэффициенте Заальшютца – Персикова, качество сигнала падает при росте времени его передачи, что означает, что некоторое окончание последовательности можно отбросить как недостоверно опознанное.

Таким образом, все что осталось профессору — отбросить несколько подряд идущих сигналов из конца последовательности так, чтобы она стала периодической с периодом, не превосходящим P . Как обычно, профессор заинтересован удалить как можно меньше сигналов.

Формат входных данных

В первой строке содержится число P — константа Персикова ($1 \leq P \leq 10^5$). В следующей строке содержится непустая последовательность, состоящая из малых букв латиницы — послание из космоса. Длина этой строки не превосходит $2 \cdot 10^5$.

Формат выходных данных

Вывести одно число — минимальное количество символов, которые нужно удалить из конца последовательности, чтобы она стала периодической с периодом T , не превосходящим P . Последовательность имеет период T , если для любых двух ее символов, расстояние между которыми кратно T верно, что они совпадают.

Пояснения к примеру

Рассмотрим исходную последовательность **abcabcaabcaabcaabab**. Если из нее ничего не удалять, то ее наименьший период будет равен 15. Если удалить из ее конца одну букву, то период **abcabcaabcaabcaaba** не изменится, если же удалить две буквы, период **abcabcaabcaabcaab** станет равен 10. Если удалить три буквы, период **abcabcaabcaabca** равен 7. И этого достаточно для первого теста. Чтобы период не превосходил 3, придется удалить из конца 10 символов и получить **abcabca** с периодом 3. Ну а период 2 можно получить только у начала **ab**, для этого придется удалить 15 символов.

Примеры

Пример №1

Стандартный ввод
8 abcabcaabcaabcaabab
Стандартный вывод
3

Пример программы-решения

Ниже представлено решение на языке C++.

```

1 //префикс-функция, метод Кнута-Морриса-Пратта
2
3 #include <bits/stdc++.h>
4
5 #define pb push_back
6 #define mp make_pair
7 #define x first
8 #define y second
9
10 #define all(x) (x).begin(), (x).end()
11 #define sz(a) (int)(a).size()
12 #define for0(i, n) for (int i = 0; i < (int)(n); i++)
13 #define for1(i, n) for (int i = 1; i <= (int)(n); i++)
14 #define int long long
15
16 using namespace std;
17 typedef pair<int, int> pii;
18 typedef long double ld;
19 typedef vector<string> vs;
20
21 vector<int> pf (string s) {
22     int n = sz(s);

```

```

23     vector<int> pi(n);
24     for (int i=1; i<n; ++i) {
25         int j = pi[i-1];
26         while (j > 0 && s[i] != s[j])
27             j = pi[j-1];
28         if (s[i] == s[j]) ++j;
29         pi[i] = j;
30     }
31     return pi;
32 }
33
34 signed main(){
35     ios::sync_with_stdio(0), cin.tie(0), cout.tie(0);
36
37     int p;
38     string s;
39
40     cin >> p >> s;
41
42     vector<int> kmp = pf(s);
43
44     for(int i = sz(kmp)-1; i >= 0; i--)
45         if(i - kmp[i] + 1 <= p)
46             return cout<<sz(s)-1 - i, 0;
47 }

```

Задача I.1.3.4. Проект «Ровные дороги» 2 (30 баллов)

При проектировании новой автодороги было принято решение сделать ее не более чем из двух абсолютно горизонтальных участков. Будущую трассу разбили на n равных по длине малых отрезков. Будем считать, что в пределах одного малого отрезка местность имеет одну и ту же высоту h_i . При этом в целях эффективной трансформации местности требуется для выравнивания использовать исключительно грунт с этой же трассы. Это означает, что можно с некоторого малого отрезка высоты h_i взять некоторое количество грунта d так, что высота этого участка станет $h_i - d > 0$. Далее эти d единиц грунта обязательно нужно переместить на другой малый отрезок высоты h_j так, что его высота станет $h_j + d$. Перемещать грунт можно только в пределах одного из двух выбранных участков, то есть отрезки номер i и номер j должны принадлежать одному и тому же горизонтальному после выравнивания участку. В данной версии задачи высоты выравниваемых участков могут быть любыми положительными, в том числе и не целыми числами. Следующим важным вопросом при строительстве являются трудозатраты. По этой причине требуется выбрать такое разбиение трассы ровно на два непустых участка, чтобы суммарный объем перемещенного грунта был минимально возможным.

Формат входных данных

В первой строке содержится число n — количество малых отрезков, на которые разбили трассу, $2 \leq n \leq 2 \cdot 10^5$. Во второй строке указаны высоты h_i этих отрезков через пробел в порядке слева направо, $1 \leq h_i \leq 2 \cdot 10^5$.

Формат выходных данных

Вывести два ненулевых числа a и b через пробел. Их сумма должна равняться n . Отрезки с номерами с первого по a -й включительно будут принадлежать первому выровненному участку, отрезки с номерами от $a + 1$ до n будут принадлежать второму выровненному участку. При этом суммарный объем грунта, перемещенного для такого выравнивания, должен быть минимальным среди всех возможных разбиений трассы на два участка. Если минимальных вариантов несколько вывести тот, у которого число a меньше.

Пояснения к примеру

На следующих рисунках приведены рассуждения для трех вариантов разбиения трассы на два участка,

$$a = 3, b = 5:$$

$$a = 3 \quad b = 5$$

$$8 \ 20 \ 2 \ 10 \ 4 \ 3 \ 1 \ 1$$

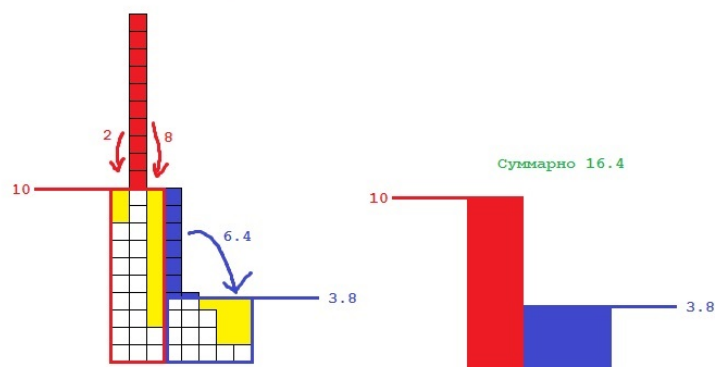
$$(8 + 20 + 2) / 3 = 10$$

на левом участке будет перемещено 10 единиц

$$(10 + 4 + 3 + 1 + 1) / 5 = 3.8$$

на правом участке будет перемещено 6.4 единиц

Всего будет перемещено $10 + 6.4 = 16.4$ единиц



$$a = 6, b = 2:$$

$$a = 6 \quad b = 2$$

8 20 2 10 4 3 1 1

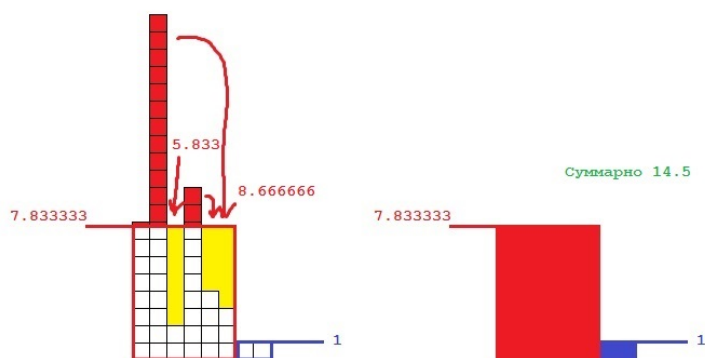
$$(8 + 20 + 2 + 10 + 4 + 3) / 6 = 7.8333333333$$

на левом участке будет перемещено 14.5 единиц

$$(1 + 1) / 2 = 1$$

на правом участке будет перемещено 0 единиц

Всего будет перемещено $14.5 + 0 = 14.5$ единиц



$a = 4, b = 4$:

$$a = 4 \quad b = 4$$

8 20 2 10 4 3 1 1

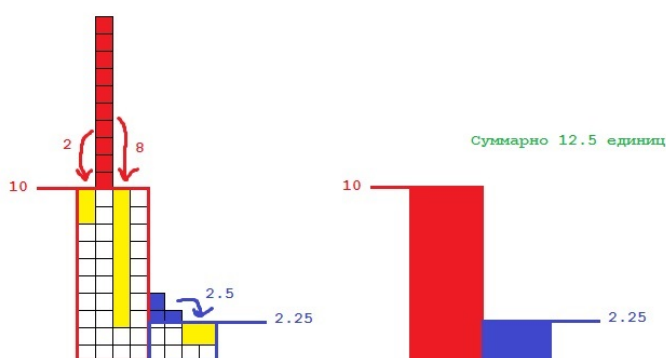
$$(8 + 20 + 2 + 10) / 4 = 10$$

на левом участке будет перемещено 10 единиц

$$(4 + 3 + 1 + 1) / 4 = 2.25$$

на правом участке будет перемещено 2.5 единиц

Всего будет перемещено $10 + 2.5 = 12.5$ единиц



Если перебрать все варианты разбиения трассы на две части, самым оптимальным будет разбиение 4 4 с результатом 12.5.

Примеры

Пример №1

Стандартный ввод
8 8 20 2 10 4 3 1 1
Стандартный вывод
4 4

Пример программы-решения

Ниже представлено решение на языке C++.

```

1 //два дерева отрезков на сумму
2
3 #include <bits/stdc++.h>
4
5 #define pb push_back
6 #define mp make_pair
7 #define x first
8 #define y second
9
10 #define all(x) (x).begin(), (x).end()

```

```

11  #define sz(a) (int)(a).size()
12  #define for0(i, n) for (int i = 0; i < (int)(n); i++)
13  #define for1(i, n) for (int i = 1; i <= (int)(n); i++)
14  #define int long long
15
16  using namespace std;
17  typedef pair<int, int> pii;
18  typedef long double ld;
19  typedef vector<string> vs;
20
21  int n, ans = 0, N = 262144;
22  vector<int> kl, su;
23  double eps = 1e-7;
24
25  int get(int l, int r, vector<int> &t){
26      int res=0;
27      for(l+=N, r+=N; l<=r; l=(l+1)>>1, r=(r-1)>>1){
28          if(l & 1)
29              res += t[l];
30          if(!(r & 1))
31              res += t[r];
32      }
33      return res;
34  }
35  void upd(int pos, int val, vector<int> &t)
36  {
37      pos += N;
38      t[pos] += val;
39      for(pos >>= 1; pos; pos >>= 1)
40      {
41          t[pos] = t[pos * 2] + t[pos * 2 + 1];
42      }
43  }
44
45  signed main(){
46      ios::sync_with_stdio(0), cin.tie(0), cout.tie(0);
47
48      cout.precision(3);
49
50      cin >> n;
51      vector<int> h(n);
52      vector<double> ld(n), rd(n), lsa(n), rsa(n);
53
54      for0(i, n)
55          cin >> h[i];
56
57      kl.resize(2*N, 0);
58      su.resize(2*N, 0);
59      double sum = 0;
60      for0(i, n){
61          sum += h[i];
62          lsa[i] = sum / (i+1);
63      }
64
65      for0(i, n){
66          upd(h[i], 1, kl);
67          upd(h[i], h[i], su);
68
69          int gr = lsa[i] + 1 + eps;
70          int k = get(gr, N-1, kl);

```

```

71     double sb = get(gr, N-1, su);
72
73     ld[i] = sb - k * lsa[i];
74 }
75
76 kl.resize(0);
77 su.resize(0);
78 kl.resize(2*N, 0);
79 su.resize(2*N, 0);
80 sum = 0;
81 for(int i = n-1; i >= 0; i--){
82     sum += h[i];
83     rsa[i] = sum / (n - i);
84 }
85
86 for(int i = n-1; i >= 0; i--){
87     upd(h[i], 1, kl);
88     upd(h[i], h[i], su);
89
90     int gr = rsa[i] + 1 + eps;
91     int k = get(gr, N-1, kl);
92     double sb = get(gr, N-1, su);
93
94     rd[i] = sb - k * rsa[i];
95 }
96
97 double mn = 1e18;
98 int a, b;
99 for(int i = 1; i < n; i++){
100     if(ld[i-1] + rd[i] < mn) {
101         mn = ld[i-1] + rd[i];
102         a = i;
103         b = n - i;
104     }
105     cout<<a<<' '<<b;
106 }

```

Задача I.1.3.5. Прогулка по мостам 2 (20 баллов)

Возможно вы знаете историю про то, как Эйлер гулял по мостам Кенигсберга. Допустим теперь, что Эйлер попал на некий архипелаг, между некоторыми островами которого имеются мосты. Мосты построены таким образом, что между любыми двумя островами архипелага можно построить путь по этим мостам, причем этот путь единственный. Все острова пронумерованы от 1 до n . Эйлер находится на острове номер 1 и желает совершить следующую прогулку: с острова номер 1 он хочет дойти до острова номер 2, далее до острова номер 3 и так далее до острова номер n . С него он хочет вернуться на остров номер 1. Он прекрасно понимает, что при этом ему придется посещать некоторые мосты по многу раз. У каждого моста есть смотритель. Одного из них заинтересовало сколько раз по его мосту пройдет Эйлер.

Формат входных данных

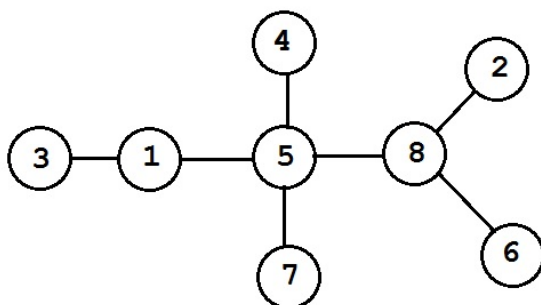
В первой строке содержится два числа через пробел: n — количество островов в архипелаге, $2 \leq n \leq 10^5$ и b — номер моста, для которого нужно узнать, сколько раз по нему проходит указанный маршрут. В следующих $n - 1$ строках содер-

жатыся по два числа a_i и b_i через пробел — номера островов, соединенных мостом, $1 \leq a_i, b_i \leq n, a_i \neq b_i$. Мосты нумеруются в порядке появления во входных данных, начиная с 1. Гарантируется, что между любыми двумя островами архипелага можно построить путь по этим мостам, причем этот путь единственный.

Формат выходных данных

Вывести одно число — ответ на задачу.

Пояснения к примеру



На рисунке вы видите расположение мостов в архипелаге из примера. Прогулка Эйлера будет проходить следующим образом:

1 — 5 — 8 — 2 — 8 — 5 — 1 — 3 — 1 — 5 — 4 — 5 — 8 — 6 — 8 — 5 — 7 — 5 — 8 — 5 — 1.

Интересующий нас мост номер 1 соединяет острова 5 и 8. Маршрут проходит по нему 6 раз.

Примеры

Пример №1

Стандартный ввод
8 1
5 8
1 3
8 6
7 5
2 8
1 5
4 5
Стандартный вывод
6

Пример программы-решения

Ниже представлено решение на языке C++.

```
1 //dfs
2 #include <bits/stdc++.h>
3
4 #define pb push_back
5 #define mp make_pair
6 #define x first
7 #define y second
8
9 #define all(x) (x).begin(), (x).end()
10 #define sz(a) (int)(a).size()
11 #define for0(i, n) for (int i = 0; i < (int)(n); i++)
12 #define for1(i, n) for (int i = 1; i <= (int)(n); i++)
13 #define int long long
14
15 using namespace std;
16 typedef long double ld;
17 typedef pair<int, int> pii;
18
19 int n, z, a, b, q1, q2;
20 vector<vector<int> > G;
21 vector<set<int> > rz(2);
22
23 void dfs(int a, int p, int dol){
24     rz[dol].insert(a);
25     for(auto to : G[a]) if(to != p) dfs(to, a, dol);
26 }
27
28 signed main(){
29     ios::sync_with_stdio(0), cin.tie(0), cout.tie(0);
30
31     cin >> n >> z;
32     z--;
33     G.resize(n);
34     for0(i, n-1){
35         cin >> a >> b;
36         a--;
37         b--;
38         if(i != z) {
39             G[a].pb(b);
40             G[b].pb(a);
41         }
42         else{
43             q1 = a;
44             q2 = b;
45         }
46     }
47
48     dfs(q1, q1, 0);
49     dfs(q2, q2, 1);
50
51     vector<int> rzb;
52     for(auto q : rz[0]) rzb.pb(q);
53
54     int ans = 0;
55     for0(i, sz(rzb)-1){
56         if(rzb[i]+1 != rzb[i+1]) ans++;
57     }
58     if(!(rzb.back() == n-1 && rzb[0] == 0)) ans++;
59     cout<<2*ans;
60 }
```

Пример программы-решения

Ниже представлено решение на языке C++.

```

1  //обход в ширину, без глубокой рекурсии
2  #include <bits/stdc++.h>
3
4  #define pb push_back
5  #define mp make_pair
6  #define x first
7  #define y second
8
9  #define all(x) (x).begin(), (x).end()
10 #define sz(a) (int)(a).size()
11 #define for0(i, n) for (int i = 0; i < (int)(n); i++)
12 #define for1(i, n) for (int i = 1; i <= (int)(n); i++)
13 #define int long long
14
15 using namespace std;
16 typedef long double ld;
17 typedef pair<int, int> pii;
18
19 int n, z, a, b, q1, q2;
20 vector<vector<int> > G;
21 vector<int> rz;
22 int mn = 1e9, aa, bb;
23
24 signed main(){
25     ios::sync_with_stdio(0), cin.tie(0), cout.tie(0);
26
27     cin >> n >> z;
28     z--;
29
30     G.resize(n+1);
31     for0(i, n-1){
32         cin >> a >> b;
33
34         if(i != z){
35             G[a].pb(b);
36             G[b].pb(a);
37         }
38         else{
39             q1 = a;
40             q2 = b;
41         }
42     }
43
44     vector<int> och;
45     vector<int> mark(n+1, -1);
46     och.pb(q1);
47     mark[q1] = q1;
48     int b = 0;
49
50     while(b < sz(och)){
51         int v = och[b];
52         for(auto to : G[v]) if(mark[to] == -1){
53             och.pb(to);
54             mark[to] = v;
55         }
56         b++;

```



```

57     }
58
59     sort(all(och));
60
61     int ans = 0;
62     for0(i, sz(och)-1){
63         if(och[i]+1 != och[i+1]) ans++;
64     }
65     if(!(och.back() == n && och[0] == 1)) ans++;
66
67     cout<<2*ans;
68 }

```

Четвертая попытка. Задачи 8–11 класса

Задача I.1.4.1. Скобки (10 баллов)

Задана строка, в которой могут быть встречены 3 типа скобок: фигурные, квадратные и круглые. Помимо скобок в строке встречаются и другие последовательно-сти символов. Вложенность скобок может быть произвольной. Необходимо проверить корректность скобочной записи: каждой открывающей скобке должна соответствовать следующая за ней закрывающая скобка того же типа на том же уровне вложенности, не должно быть открывающей или закрывающей скобки без пары.

Формат входных данных

Строка, содержащая произвольный набор символов (в т. ч. и без скобок).

Формат выходных данных

- Слово «correct», если запись корректна или не содержит скобок.
- Слово «incorrect», если запись не корректна.

Примеры

Пример №1

Стандартный ввод
(this [is] test)
Стандартный вывод
correct

Пример программы-решения

Ниже представлено решение на языке Python3.

```

1 def balancedBrackets(expr):
2     stack = []
3     for char in expr:
4         if char in ['(', '{', '[']:

```

```
5         stack.append(char)
6     elif char in [')', '}', ']']:
7         if(len(stack) == 0):
8             return False
9         top = stack.pop()
10        if(top == '(' and char != ')'):
11            return ""
12        if(top == '{' and char != '}'):
13            return False
14        if(top == '[' and char != ']'):
15            return False
16    else:
17        continue
18    if(len(stack) > 0):
19        return False
20    return True
21
22 expr = input()
23 print("correct") if balancedBrackets(expr) else print("incorrect")
```

Задача I.1.4.2. Рисунок (15 баллов)

Задан рисунок, состоящий из пронумерованных точек и линий между ними. Напишите программу, которая скажет, можно ли нарисовать этот рисунок (провести ручку по всем точкам и линиям), не отрывая руки и при этом не проводя одну линию дважды?

Формат входных данных

Первая строка содержит число N — количество точек (число от 0 до 1000 включительно).

Вторая строка содержит число M — количество линий (число от 0 до 1000 включительно).

Далее идет M строк, каждая из которых содержит пары номеров точек, соединенных линиями (нумерация точек с 1, между парой точек может быть проведено несколько линий).

Формат выходных данных

- «Yes», если рисунок можно нарисовать в соответствии с условием.
- «No», если нет.

*Примеры**Пример №1*

Стандартный ввод
3 2 1 2 2 3
Стандартный вывод
Yes

Пример программы-решения

Ниже представлено решение на языке Python3.

```

1 def task1():
2     vert_num = int(input())
3     edge_num = int(input())
4     verts = [{"watched": False} for _ in range(vert_num)]
5     edges = []
6     for _ in range(edge_num):
7         edges.append(tuple(int(i)-1 for i in input().split()))
8
9     if vert_num == 0:
10        return 'Yes'
11    line = [0, 0]
12    verts[0]['watched'] = True
13    flag = True
14    while flag:
15        flag = False
16        for edge in edges:
17            if edge[0] in line or edge[1] in line:
18                if edge[0] in line:
19                    verts[edge[1]]['watched'] = True
20                    if edge[0] == line[0]:
21                        line[0] = edge[1]
22                else:
23                    line[1] = edge[1]
24            else:
25                verts[edge[0]]['watched'] = True
26                if edge[1] == line[0]:
27                    line[0] = edge[0]
28                else:
29                    line[1] = edge[0]
30            edges.remove(edge)
31            flag = True
32    for v in verts:
33        if not v['watched']:
34            return 'No'
35    if len(edges):
36        return 'No'
37    return 'Yes'
38
39 print(task1())

```

Задача I.1.4.3. Часовой механизм (20 баллов)

К вам обратился владелец часовой мастерской, которая делает часы с прозрачным корпусом. В мастерской есть одна проблема: дизайнеры придумывают то, как будут выглядеть часы, но не задумываются о том, как шестеренки будут крутиться. Поэтому, когда дизайн получают мастера, им приходится проверять работоспособность нарисованного дизайнерами механизма — проверять не заклинит ли механизм от сцепки двух крутящихся в одном направлении шестеренок.

Напишите программу, которая будет делать эту работу за мастеров.

Формат входных данных

Первая строка содержит количество шестеренок N (число от 1 до 1000 включительно).

Вторая строка содержит количество связей между шестеренками M (число от 0 до 1000 включительно), которые сцеплены между собой (если одна из них крутится по часовой стрелке, вторая должна крутиться против часовой и наоборот).

Далее идет M строк с парами чисел, являющихся номерами сцепленных между собой шестеренок (нумерация начинается с 1).

Формат выходных данных

Программа должна вывести одно из 2 слов: «good», если механизм работоспособен или «bad», если механизм заклинит.

Примеры

Пример №1

Стандартный ввод
4
4
1 2
2 3
3 4
4 1
Стандартный вывод
good

Пример программы-решения

Ниже представлено решение на языке Python3.

```

1 def get_unwatched(verts):
2     i = 0
3     for vert in verts:
4         if not vert['watched']:
5             vert['orient'] = 1
6             return vert
7 
```

```

8 def task1():
9     vert_num = int(input())
10    edges = int(input())
11    verts = [{"edges": [], "orient": 0, "watched": False} for _ in range(vert_num)]
12    for _ in range(edges):
13        a, b = (int(i)-1 for i in input().split())
14        verts[a]["edges"].append(b)
15        verts[b]["edges"].append(a)
16
17    stack = []
18    while vert_num > 0:
19        if len(stack) == 0:
20            stack.append(get_unwatched(verts))
21        vert = stack.pop()
22        vert_num -= 1
23        vert["watched"] = True
24        for next_v in vert['edges']:
25            if not verts[next_v]["watched"] and verts[next_v] not in stack:
26                stack.append(verts[next_v])
27            if verts[next_v]["orient"] != 0 and verts[next_v]["orient"] +
28                ↪ vert['orient'] != 3:
29                return 'bad'
30            elif verts[next_v]["orient"] == 0:
31                verts[next_v]["orient"] = 3 - vert['orient']
32    return 'good'
33 print(task1())

```

Задача I.1.4.4. Парковка (20 баллов)

На парковке у бизнес-центра N контрольно-пропускных пунктов(КПП). Известно, что каждый день через каждый день работающие в центре сотрудники въезжают на парковку M раз. Каждый раз сотрудник въезжает в определенное время через один заранее известный КПП и выезжает в определенное время через другой (не обязательно совпадающий с первым). На въезде сотрудник получает пропуск, который он должен отдать на выезде. Один и тот же пропуск может быть использован разными сотрудниками, если они находятся на территории парковки в разное время.

Определите минимальное количество пропусков, которое должно быть суммарно на всех КПП к началу дня, чтобы хватило всем сотрудникам.

Если время въезда и выезда разных сотрудников на одном КПП совпадает, считать, что они могут воспользоваться одним пропуском

Формат входных данных

Первая строка содержит число N (число от 0 до 1000 включительно).

Вторая строка содержит число M (число от 0 до 1000 включительно).

Далее следует M строк, содержащих следующую информацию:

Время въезда (число от 6 до 22 включительно).

Время выезда (число от 7 до 23 включительно).

Номер КПП при въезде (нумерация с 1).

Номер КПП при выезде (нумерация с 1).

Формат выходных данных

Программа должна вывести количество пропусков необходимое суммарно на всех КПП к началу дня.

Примеры

Пример №1

Стандартный ввод
2
2
6 12 1 2
12 22 2 2
Стандартный вывод
1

Пример программы-решения

Ниже представлено решение на языке Python3.

```

1 def solve():
2     required = 0
3     kpps_count = int(input())
4     kpps = [0] * kpps_count
5     n_notes = int(input())
6     visitors = []
7     for _ in range(n_notes):
8         entry_hour, out_hour, in_kpp, out_kpp = [int(readed) for readed in
9             ↪ input().split()]
10        visitors.append((entry_hour, in_kpp - 1, True))
11        visitors.append((out_hour, out_kpp - 1, False))
12
13    visitors.sort(key = lambda visitor: (visitor[0], visitor[2]))
14
15    for visitor in visitors:
16        if visitor[2]:
17            if not kpps[visitor[1]]:
18                required += 1
19            else:
20                kpps[visitor[1]] -= 1
21        else:
22            kpps[visitor[1]] += 1
23    print(required)
24 solve()

```

Задача I.1.4.5. Обучаем Терминатора (35 баллов)

Перед отправкой Терминатора T-800 в прошлое для спасения Джона Коннора (события 2 части) обнаружилось, что при анализе текстовых документов, OCR-модуль машины допускает ошибки при чтении символов в записи моделей терми-

наторов. Времени на повторное обучение нейронной сети нет, поэтому было принято решение написать hot-fix на символы «Г», «0», «1», «8» и «-». При чтении Терминатор каждый символ переводит в матрицу 10 на 10 точек, где 1 означает наличие заполнения, а 0 — отсутствие.

Символы распознаются следующим образом:

- «Г» — два прямоугольника лежащих друг на друге, левая граница верхнего прямоугольника левее нижнего, правая граница верхнего прямоугольника правее нижнего.
- «0» — заполненный прямоугольник с прямоугольным вырезом внутри, границы выреза не лежат на сторонах внешнего прямоугольника.
- «8» — заполненный прямоугольник с двумя прямоугольными вырезами внутри, границы вырезом не лежат на сторонах внешнего прямоугольника, границы вырезом не пересекаются, нижняя граница одного выреза выше другого.
- «1» — заполненный прямоугольник, ширина прямоугольника строго меньше его длины.
- «-» — заполненный прямоугольник, ширина прямоугольника строго больше его длины.

Необходимо, чтоб остальные комбинации интерпретировались символом «X».

Напишите программу для решения поставленной задачи.

Формат входных данных

10 строк состоящих из 10 символов «0» или «1».

Формат выходных данных

Один из символов «Г», «0», «1», «8», «-» или «X».

Примеры

Пример №1

Стандартный ввод
0000000000
0001110000
0001010000
0001010000
0001110000
0001110000
0001010000
0001010000
0001110000
0000000000
Стандартный вывод
8

Пример программы-решения

Ниже представлено решение на языке Python3.

```
1 def get_borders(l, sym=1):
2     a, b = None, None
3     for i in range(len(l)):
4         if sym in l[i]:
5             a = i
6             break
7     for i in range(len(l)-1, -1, -1):
8         if sym in l[i]:
9             b = i
10            break
11    return a, b
12
13 def get_lr(l, sym=1):
14    left, right = None, None
15    for i in range(len(l)):
16        if l[i] == sym:
17            left = i
18            break
19    for i in range(len(l)-1, -1, -1):
20        if l[i] == sym:
21            right = i
22            break
23    return left, right
24
25 def get_rectangle(l, b):
26    result = []
27    for i in range(b[0], b[2]+1):
28        tmp = []
29        for j in range(b[1], b[3]+1):
30            tmp.append(l[i][j])
31        result.append(tmp)
32    return result
33
34 def is_rectangle(l, sym=1):
35    up, bottom = get_borders(l, sym)
36    if up is None:
37        return None
38    left, right = get_lr(l[up], sym)
39    for i in range(up, bottom+1):
40        if (left, right) != get_lr(l[i], sym):
41            return None
42    return (up, left, bottom, right)
43
44 def task5():
45    rows = []
46    for _ in range(10):
47        rows.append([int(i) for i in input()])
48    a = is_rectangle(rows)
49    if a:
50        tmp = get_rectangle(rows, a)
51        if is_rectangle(tmp, 0):
52            return '0'
53        for i in range(len(tmp)):
54            if is_rectangle(tmp[:i], 0) and is_rectangle(tmp[i:], 0):
55                return '8'
56    if a[2]-a[0] > a[3]-a[1]:
```



```

57     return '1'
58     if a[2]-a[0] < a[3]-a[1]:
59         return '-'
60     else:
61         up, bottom = get_borders(rows)
62         for i in range(up, bottom+1):
63             a = is_rectangle(rows[:i])
64             b = is_rectangle(rows[i:])
65             if a and b and a[1] < b[1] and a[3] > b[3]:
66                 return 'T'
67     return 'X'
68
69 print(task5())

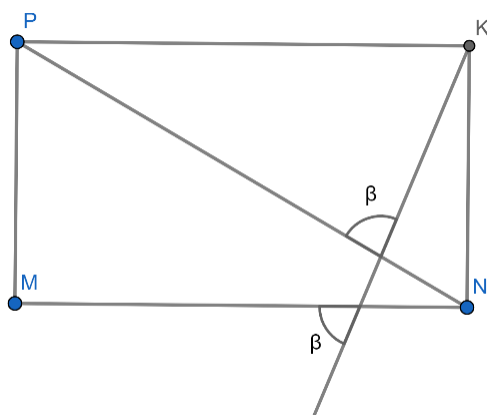
```

Задачи первого этапа. Математика

Первая попытка. Задачи 8–9 класса

Задача I.2.1.1. (20 баллов)

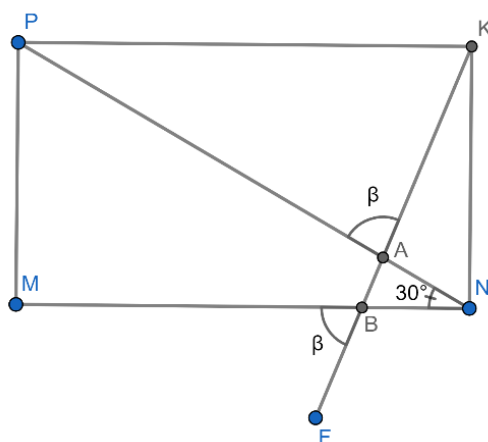
Сторона PM прямоугольника $MNKP$ равна 6, диагональ PN равна 12. Найдите величину угла β . Ответ запишите в градусах.



Решение

В прямоугольном треугольнике PMN катет PM вдвое меньше гипотенузы PN поэтому $\angle PNM = 30^\circ$.

В треугольнике ABN углы A и B равны β как вертикальные с углами KAP и MCF соответственно, а угол N равен 30° . По теореме о сумме углов треугольника получаем: $2\beta + 30^\circ = 180^\circ$, откуда $\beta = 75^\circ$.



Ответ: 75.

Задача I.2.1.2. (20 баллов)

Пираты Меткий Джек, Капитан Флинт и Длинноногий Бенни, приплыв на остров, стали обмениваться добычей. Сначала Джек дал Флинту и Бенни столько слитков золота, сколько было у каждого из них. Затем Флинт дал Джеку и Бенни столько, сколько стало у каждого из них. И наконец, Бенни дал Джеку и Флинту столько, сколько у каждого из них к этому моменту имелось. В результате у всех оказалось по 80 слитков. Сколько слитков было у каждого в начале? В ответе напишите одно число, состоящее из первоначальных количеств, без пробелов в порядке: Меткий Джек, Капитан Флинт, Длинноногий Бенни.

Решение

Будем рассматривать тройку количеств слитков (Джек; Флинт; Бенни). После всех обменов эта тройка (80; 80; 80). При последнем обмене два первых числа удвоились. Значит, тройка была (40; 40; 160). Она получилась при обмене, когда удвоились первое и третье число. Значит, была (20; 140; 80). Эта тройка возникла при удвоении второго и третьего числа. Получаем первоначальную тройку (130; 70; 40).

Ответ: 1307040.

Задача I.2.1.3. (20 баллов)

На кружок по робототехнике ходят ученики 6, 7, 8 и 9 классов (их возраст 12, 13, 14 и 15 лет соответственно). Известно, что шестиклассников меньше, чем семиклассников, а семиклассников меньше, чем восьмиклассников. При этом девятиклассников в кружке столько же, сколько семи- и восьмиклассников вместе взятых, а количества шести- и семиклассников нечетны. На одном из занятий кружка руководитель с удивлением обнаружил, что средний возраст ребят — целое число лет. Сколько учащихся посещают занятия кружка, если их меньше 30?

Решение

Пусть a, b, c, d — количество шести-, семи-, восьми- и девятиклассников соответственно, n лет — средний возраст ребят, тогда:

$$\frac{12a + 13b + 14c + 15d}{a + b + c + d} = n$$

или $(12 - n)a + (13 - n)b + (14 - n)c + (15 - n)d = 0$. С учетом условия $d = b + c$ получаем $(12 - n)a + (28 - 2n)b + (29 - 2n)c = 0$. Перепишем уравнение в виде $(69 - 5n)a + (57 - 4a)(b - a) + (29 - 2n)(c - b) = 0$. Поскольку $a < b < c$, при $n \leq 13$ левая часть уравнения положительна, а при $n \geq 13$ — отрицательна. Поэтому единственное возможное значение $n = 13$, следовательно, $c = 2a$.

По условию a и b нечетны. Рассмотрим возможные значения a :

- а. Случай $a = 1$ невозможен, т. к. тогда $c = 2$ и условие $a < b < c$ невыполнимо.
- б. Если $a = 3$, то $c = 6$. Единственное возможное нечетное значение $b = 5$. При этом $d = 11$, и общее количество ребят равно 25. (Легко убедиться, что все условия задачи выполнены).
- с. Если $a \geq 5$, то $b \geq 7$, $c \geq 10$, $d = b + c \geq 17$. Общее количество участников кружка $a + b + c + d \geq 39$, и условия задачи не выполняются.

Таким образом, единственное решение задачи соответствует случаю b .

Ответ: 25.

Задача I.2.1.4. (20 баллов)

У Ани есть квадратный трехчлен $A(x) = x^2 + 3x - 6$, а у Бори — $B(x) = x^2 - 11x + 22$. Каждый из ребят загадал натуральное число, a и b соответственно. Оказалось, что $A(a) = B(b)$. Найдите наибольшее возможное значение $|a - b|$.

Решение

По условию $a^2 + 3a - 6 = b^2 - 11b + 22$ или $a^2 + b^2 + 3a + 11b - 28 = 0$. Раскладывая левую часть на множители, получаем: $(a + b - 4)(a - b + 7) = 0$. Отсюда $a + b = 4$ или $a - b = -7$. В первом случае имеем три пары чисел (1; 3) (2; 2) (3; 1), при этом $|a - b|$ принимает значения 0 или 2. Во втором случае $|a - b| = 7$, и это наибольшее возможное значение.

Ответ: 7.

Задача I.2.1.5. (20 баллов)

У Саши есть сундучок с карточками, на которых записаны все четырехзначные числа, не содержащие в своей записи цифры 0 и 1 (на каждой карточке записано одно число). Сергей случайным образом достает одну карточку. Какова вероятность, что на этой карточке написано число, в котором цифры различны и расположены в порядке убывания, начиная с разряда тысяч?

Ответ запишите в виде десятичной дроби с тремя верными цифрами после запятой.

Решение

Всего карточек у Саши $8^4 = 4096$. Количество чисел, в которых цифры расположены в порядке убывания C_8^4 , так как каждой неупорядоченной группе различных цифр соответствует ровно одно число, в котором эти цифры расположены в порядке убывания, их количество 70. Таким образом, искомая вероятность равна $\frac{70}{4096} = 0,0178984375$.

Ответ: 0,017.

Первая попытка. Задачи 10–11 класса

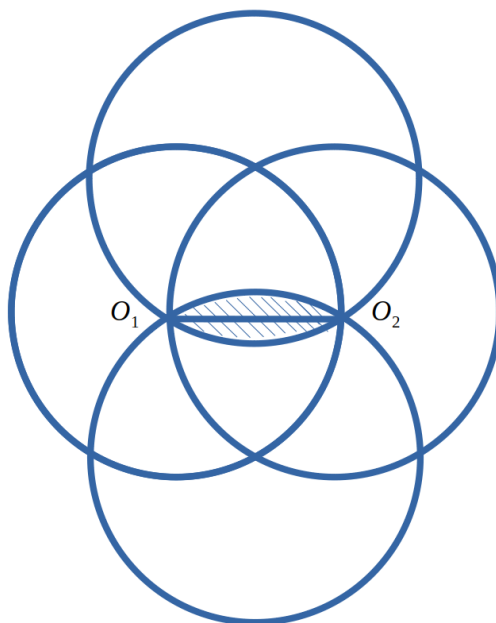
Задача I.2.2.1. (15 баллов)

Точность плоттеров (графических построителей) определяется минимально возможным значением приращения координаты. Гоша сконструировал плоттер из доступных ему деталей и решил определять его точность с помощью следующего приема: построителем строятся две окружности одинакового радиуса так, что каждая проходит через центр другой, а две другие окружности того же радиуса имеют центры в точках пересечения первых двух окружностей. Далее с помощью специального алгоритма, который Гоше передал приятель, в двух последовательных измерениях оценивается площадь области, общей для всех четырех кругов, при этом во втором измерении перо построителя перемещается как раз на искомое минимально возможное значение. Какова точность графопостроителя в миллиметрах, если в первом измерении радиус R окружностей был равен 6 см, а площадь заданной области во втором измерении — $669,78 \text{ мм}^2$. В алгоритме число $\pi = 3,14$, а иррациональные числа и любой результат деления округляются до сотой доли.

- 1 мм.
- 1,5 мм.
- 0,5 мм.
- 2 мм.

Решение

Каждая из двух последних окружностей проходит через центры первых двух (см. рисунок к задаче), поэтому длина из общей хорды будет равна $O_1O_2 = R$.



Искомая площадь будет равна удвоенной площади сегмента с центральным углом 60° , т. е.:b

$$S = 2 \left(\frac{\pi R^2}{6} - \frac{R^2 \sqrt{3}}{4} \right) = \frac{R^2 (2\pi - 3\sqrt{3})}{6}.$$

Подставим в полученное выражение для площади известные значения переменных, учтя, что для второго измерения вместо R нужно писать $R + \Delta R$, и найдем решение:

$$\frac{(R + \Delta R)^2 (2\pi - 3\sqrt{3})}{6} = 669,78$$

$$(60 + \Delta R)^2 \cdot 0,18 = 669,78$$

$$(60 + \Delta R)^2 = 61^2$$

$$\Delta R = 1$$

Ответ: 1 мм.

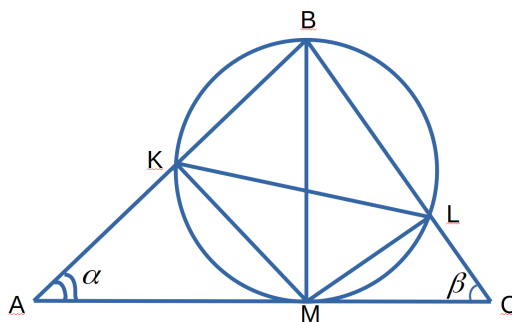
Задача I.2.2.2. (20 баллов)

Программист разработал алгоритм по оценке площади плоских фигур, а также фигур, вписанных в окружность, при этом в качестве результата программа вывела отношение площадей двух выбранных фигур. На этапе тестирования для анализа эффективности на вход алгоритма был треугольник ABC с углами $\angle A = \alpha$ и $\angle B = \beta$, а также высотой BM , на которой была построена окружность. Данная окружность пересекает сторону AB в точке K , а сторону BC — в точке L . Верно ли сработал алгоритм, если при вычислении отношения площади треугольника KLM к площади треугольника ABC для углов $\alpha = 37^\circ$ и $\beta = 60^\circ$ ответом алгоритма было число 0.194587? Если алгоритм верно произвел вычисление, то до какого знака после запятой (включительно)?

1. Алгоритм выполнил вычисления некорректно.
2. Алгоритм выполнил вычисления корректно с точностью до 1 знака после запятой.
3. Алгоритм выполнил вычисления корректно с точностью до 2 знака после запятой.
4. Алгоритм выполнил вычисления корректно с точностью до 3 знака после запятой.
5. Алгоритм выполнил вычисления корректно с точностью до 4 знака после запятой.

Решение

По условию, $BM \perp AC$, BM — диаметр окружности, $\angle BAC = \alpha$, $\angle BCA = \beta$, K и L — точки пересечения окружности со сторонами AB и BC (см. рис.); требуется найти $S_{\Delta KLM} : S_{\Delta ABC}$.



Пусть $BM = h$; тогда $AM = h \operatorname{ctg} \alpha$, $MC = h \operatorname{ctg} \beta$. Имеем:

$$\begin{aligned} S_{\Delta ABC} &= \frac{1}{2} BM \cdot AC = \frac{1}{2} BM \cdot (AM + MC) = \frac{1}{2} h (h \operatorname{ctg} \alpha + h \operatorname{ctg} \beta) = \\ &= \frac{1}{2} h^2 (\operatorname{ctg} \alpha + \operatorname{ctg} \beta) = \frac{1}{2} h^2 \frac{\sin(\alpha + \beta)}{\sin \alpha \sin \beta} \end{aligned}$$

Так как $\angle ABM = 90^\circ - \alpha$, $\angle MBC = 90^\circ - \beta$, а также $\angle BKM = \angle BLM = 90^\circ$ (вписанные углы, опирающиеся на диаметр), то $\angle BMK = \alpha$, $\angle BML = \beta$. Тогда из ΔBKM и ΔBLM находим $KM = h \cos \alpha$, $ML = h \cos \beta$. Значит:

$$S_{\Delta KLM} = \frac{1}{2} KM \cdot ML \sin(\alpha + \beta) = \frac{h^2}{2} \cos \alpha \cos \beta \sin(\alpha + \beta)$$

Окончательно получим:

$$S_{\Delta KLM} : S_{\Delta ABC} = \frac{0.5h^2 \cos \alpha \cos \beta \sin(\alpha + \beta) \sin \alpha \sin \beta}{0.5h^2 \sin(\alpha + \beta)} = \frac{1}{4} \sin 2\alpha \sin 2\beta$$

Подставляя данные в условии задачи значения углов α и β , получим окончательно, что $S_{\Delta KLM} : S_{\Delta ABC} = 0,19459461$. Можно заключить, что в данном случае алгоритм сработал корректно, а точность вычислений — до 4-го знака после запятой включительно.

Ответ: 5.

Задача I.2.2.3. (15 баллов)

Путешествуя по джунглям, леди T . забрела на поляну, на которой обитают 15 пауков-птицеедов и 25 древесных лягушек. Какова вероятность того, что после прогулки по этой поляне леди T . снимет с себя двух лягушек и трех пауков? Записать ответ с точностью до тысячных.

Решение

Всего на поляне было $15 + 25 = 40$ насекомых. Общее количество вариантов выбора 5 насекомых из 40 равно:

$$n = C_{40}^5 = \frac{40!}{5! \cdot 35!} = 12 \cdot 37 \cdot 38 \cdot 39.$$

Количество вариантов, когда возникает событие, благоприятствующее условию задачи и снятию с себя 3 пауков и 2 лягушек, равно:

$$m = C_{15}^3 \cdot C_{25}^2 = \frac{15!}{3! \cdot 12!} \cdot \frac{25!}{2! \cdot 23!} = 2 \cdot 13 \cdot 14 \cdot 15 \cdot 25.$$

Тогда вероятность будет равна:

$$p = \frac{m}{n} = \frac{2 \cdot 13 \cdot 14 \cdot 15 \cdot 25}{12 \cdot 37 \cdot 38 \cdot 39} = \frac{5 \cdot 7 \cdot 25}{3 \cdot 37 \cdot 38} = 0,207.$$

Ответ: 0,207.

Задача I.2.2.4. (20 баллов)

Найти количество положительных значений параметра a , при которых неотрицательные значения x , удовлетворяющие уравнению:

$$\cos(21ax + 16x) = \cos(10x - 11ax)$$

И расположенные в порядке возрастания, образуют арифметическую прогрессию.

- Таких решений нет.
- 1 значение параметра a .
- 2 значения параметра a .
- 3 значения параметра a .
- 4 значения параметра a .
- 5 значений параметра a .
- 6 значений параметра a .
- 7 значений параметра a .
- 8 значений параметра a .

Решение

Преобразуем исходное уравнение:

$$2 \sin \left(\frac{21ax + 16x + 10x - 11ax}{2} \right) \cdot \sin \left(\frac{21ax + 16x - 10x + 11ax}{2} \right) = 0.$$

Тогда:

$$\begin{cases} \sin(5ax + 13x) = 0, \\ \sin(16ax + 3x) = 0, \end{cases} \Rightarrow \begin{cases} 5ax + 13x = \pi k, k \in N, \\ 16ax + 3x = \pi n, n \in N, \end{cases} \Rightarrow \begin{cases} x = \frac{\pi k}{5a+13}, k \in N, \\ x = \frac{\pi n}{16a+3}, n \in N. \end{cases}$$

Положительные значения каждого решения (при $k, n \in N$) образуют арифметическую прогрессию с разностями соответственно:

$$d_1 = \frac{\pi}{5a+13}, d_2 = \frac{\pi}{16a+3}$$

Из членов этих прогрессий можно составить в порядке возрастания одну прогрессию, только когда а) $\frac{d_1}{d_2} = p$ или б) $\frac{d_2}{d_1} = q$ (при $p, q \in N$).

$$\text{а) } \frac{\pi}{5a+13} : \frac{\pi}{16a+3} = p \Rightarrow a = \frac{13p-3}{16-5p}.$$

Так как по условию $a > 0$, то, решая неравенство $\frac{13p-3}{16-5p} > 0$, получим $\frac{3}{13} < p < \frac{16}{5}$.

Так как p - натуральное число, то $p = \{1; 2; 3\}$ и, соответственно, $a = \left\{ \frac{10}{11}; \frac{23}{6}; 36 \right\}$.

$$\text{б) } \frac{\pi}{16a+3} : \frac{\pi}{5a+13} = q \Rightarrow a = \frac{13-3q}{16q-5} \Rightarrow \frac{5}{16} < p < \frac{13}{3} \Rightarrow q = \{1; 2; 3; 4\} \text{ и } a = \left\{ \frac{10}{11}; \frac{7}{27}; \frac{4}{43}; \frac{1}{59} \right\}.$$

Таким образом, всего при 6 значениях параметра a решения уравнения можно выстроить в арифметическую прогрессию.

Ответ: 6 значений параметра a .

Задача I.2.2.5. (30 баллов)

Вычислить модуль тангенса угла при вершине равнобедренного треугольника с наибольшей площадью, при условии, что дана длина медианы, проведенной к боковой стороне данного треугольника. Записать ответ с точностью до сотых.

Решение

Обозначим за a длину боковой стороны треугольника, l — длину медианы, проведенной к боковой стороне a , α — угол между двумя боковыми сторонами. Очевидно, что площадь треугольника $S = 0,5a^2 \sin \alpha$. Выразим длину медианы через длины боковых сторон через теорему косинусов:

$$l^2 = a^2 + \left(\frac{a}{2}\right)^2 - 2a \frac{a}{2} \cos \alpha = a^2 \left(\frac{5}{4} - \cos \alpha\right).$$

Из этого выражения выразим боковую сторону треугольника:

$$a^2 = \frac{4l^2}{5 - 4 \cos \alpha},$$

откуда выражение для площади треугольника будет переписана следующим образом:

$$S = \frac{2l^2 \sin \alpha}{5 - 4 \cos \alpha} = S(\alpha),$$

где $0 < \alpha < \pi$ по смыслу задачи. Из выражения для площади видно, что при $\alpha \rightarrow 0$ и $\alpha \rightarrow \pi$ функция $S(\alpha) \rightarrow 0$. При этом очевидно и из смысла площади, и из математической функции, что $S(\alpha) > 0$. Если в интервале $(0; \pi)$ функция $S(\alpha)$ имеет критическую точку, то в ней $S(\alpha)$ принимает наибольшее значение. Найдем производную от функции $S(\alpha)$:

$$S'(\alpha) = 2l^2 \frac{5 \cos \alpha - 4}{(5 - 4 \cos \alpha)^2} = 0,$$

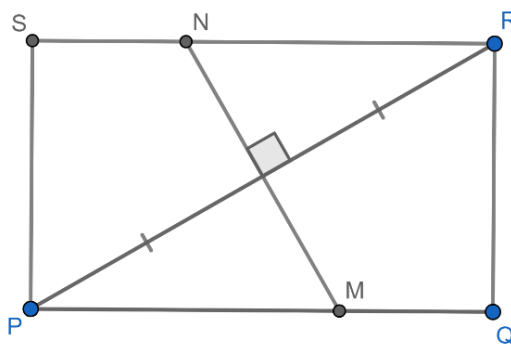
откуда получаем, что наибольшее значение площади будет при $\cos \alpha = 0,8$. Следовательно, $\sin \alpha = 0,6$ и $\tan \alpha = \frac{3}{4}$.

Ответ: 0,75.

Вторая попытка. Задачи 8–9 класса

Задача I.2.3.1. (20 баллов)

Дан прямоугольник $PQRS$ со сторонами $PQ = 16$ и $QR = 12$. Серединный перпендикуляр к PR пересекает стороны PQ и RS в точках M и N соответственно. Найдите длину отрезка MN .



Решение

Пусть Z — середина диагонали PR .

По теореме Пифагора для треугольника PQR : $PQ = \sqrt{PQ^2 + QR^2} = 20$, тогда $PZ = \frac{1}{2}PQ = 10$.

Прямоугольные треугольники RNZ и PMZ равны по катету и прилежащему острому углу: $RZ = PZ$ по условию, а $\angle NRZ = \angle MPZ$ как внутренние накрест лежащие при параллельных прямых SR и PQ и секущей PR . Следовательно, $NZ = MZ = \frac{1}{2}MN$.

Прямоугольные треугольники PZM и PQR подобны по общему острому углу P , поэтому $\frac{PZ}{PQ} = \frac{ZM}{QR}$ откуда $ZM = \frac{PZ \cdot QR}{PQ} = \frac{10 \cdot 12}{16} = \frac{15}{2}$.

Окончательно получаем: $MN = 2 \cdot ZM = 15$.

Ответ: 15.

Задача I.2.3.2. (20 баллов)

Петя поделил натуральное число на 8, 5 и 3 с остатком. Сумма всех остатков оказалась равной 13. Найдите наибольшее возможное трехзначное число, которое делил Петя.

Решение

Остаток от деления меньше делителя, поэтому сумма 13 может получиться, только если остаток от деления на 8 равен 7, от деления на 5 равен 4, при делении на 3 равен 2. Тогда, если к искомому числу прибавить 1, то оно будет делиться на 8, 5 и 3. Поскольку эти числа взаимно простые, то оно будет делиться на их произведение равное 120. Наибольшее такое трехзначное число 960. Искомое число на 1 меньше.

Ответ: 959.

Задача I.2.3.3. (20 баллов)

В начале сентября класс из 25 человек написал сочинение «Как я провел лето». Учитель при проверке выставляет двойную оценку. Например, оценка $4/5$ означает, что 4 поставлено за содержание, 5 — за грамотность. Двоек не было! Учеников, имеющих среди своих оценок хотя бы одну 3, было 16, хотя бы одну 4 — 12, хотя бы одну 5 — 5. «Твердых хорошистов» (с оценкой $4/4$) было вдвое меньше «упертых троечников» (с оценкой $3/3$). Сколько было среди учеников «круглых отличников» (с оценкой $5/5$), если известно, что их было нечетное число?

Решение

Пусть x — количество учеников с оценкой $4/4$, тогда учеников с оценкой $3/3$ — $2x$. Пусть y — количество учеников, получивших $3/4$ или $4/3$, z — количество учеников с оценками $3/5$ или $5/3$, u — количество учеников с оценками $4/5$ или $5/4$, v — количество учеников с оценкой $5/5$. Из условия задачи получаем систему уравнений:

$$\begin{cases} 2x + y + z + u + v = 25 \\ 2x + y + z = 16 \\ y + x + u = 12 \\ z + u + v = 5 \end{cases}$$

Выражая все переменные через x получаем:

$$\begin{cases} z = x - 4 \\ y = 20 - 3x \\ u = 2x - 8 \\ v = 17 - 3x \end{cases}$$

Возможны только два значения x : 4 и 5. При допустимом значении x получаем нечетное $v = 5$.

Ответ: 5.

Задача I.2.3.4. (20 баллов)

Сергей вычислил значения приведенного квадратного трехчлена при трех значениях $x = 3, 14$ и 16 и сложил их. Саша вычислил значения другого приведенного квадратного трехчлена при тех же значениях x , сложил их и получил ту же сумму. При каком значении x трехчлены Сергея и Саши равны? (У приведенного многочлена коэффициент при наибольшей степени равен единице).

Решение

Пусть у Саши был многочлен $P(x) = x^2 + ax + b$ а у Сергея $Q(x) = x^2 + cx + d$. По условию:

$$\begin{aligned} 3^2 + 3a + b + 14^2 + 14a + b + 16^2 + 16a + b &= \\ = 3^2 + 3c + d + 14^2 + 14c + d + 16^2 + 16c + d \end{aligned}$$

Отсюда $33a + 3b = 33c + 3d$, то есть:

$$11(a - c) = d - b \tag{I.2.1}$$

Если $a = c$, то и $d = b$, то есть $P(x)$ совпадает с $Q(x)$, что по условию не так. Пусть x_0 — искомое значение, то есть $P(x_0) = Q(x_0)$ или $x_0^2 + ax_0 + b = x_0^2 + cx_0 + d$.

Отсюда, учитывая (I.2.1), получаем $x_0 = \frac{d-b}{a-c} = 11$.

Ответ: 11.

Задача I.2.3.5. (20 баллов)

Если выбрать два произвольных различных числа из множества $\{1, 2, 3, \dots, n-1, n\}$, вероятность того, что числа являются последовательными, равна $\frac{1}{31}$. Найдите n .

Решение

Всего способов выбрать два числа из n различных чисел:

$$C_n^2 = \frac{n(n-1)}{2}$$

Количество пар последовательных чисел из n равно $n - 1$. Таким образом, данная вероятность:

$$\frac{1}{31} = \frac{n - 1}{C_n^2} = \frac{2}{n}$$

Отсюда $n = 62$.

Ответ: 62.

Вторая попытка. Задачи 10–11 класса

Задача I.2.4.1. (15 баллов)

На Земле после внезапного эволюционного изменения зайцы стали плотоядными и начали охотиться на волков, в то время как волки остались плотоядными и не изменили свой рацион.

При наблюдении за одной особью ученый заметил, что данный заяц имеет обычное поведение: делать запасы, охотясь за двумя волками сразу. Статистика показывает, что нападение начинает всегда заяц и, напав на каждого из двух волков поочередно, с вероятностью $p_1 = 0,6$ ловит волка. Если волк не становится жертвой зайца, то, независимо от судьбы другого волка, начинает сам охотиться на нападающего зайца и с вероятностью $p_2 = 0,81$ ловит зайца-обидчика.

Найти вероятность того, что в очередной схватке двух волков и одного зайца погибнет одно животное, не важно кто. Ответ записать с точностью до десятитысячной.

- 0,1998.
- 0,3058.
- 0,0144.
- 0,2841.

Решение

Существует три случая, когда погибнет одно животное: либо A_1 — гибнет заяц, а волки остаются целыми, либо A_2 — гибнет первый волк, а второй волк и заяц остаются живыми; либо A_3 — второй волк гибнет, а первый волк и заяц выживают. При этом искомая вероятность $P = P(A_1) + P(A_2) + P(A_3)$.

Определим значения каждого из событий:

$$P(A_1) = (1 - p_1)^2 \cdot [1 - (1 - p_2)^2] = 0,154224;$$

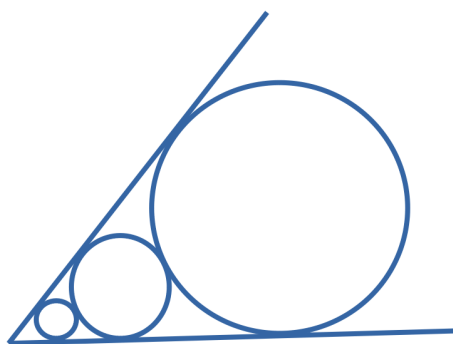
$$P(A_2) = P(A_3) = p_1 \cdot (1 - p_1) \cdot (1 - p_2) = 0,0456;$$

$$P = (1 - p_1)^2 \cdot [1 - (1 - p_2)^2] + 2p_1 \cdot (1 - p_1) \cdot (1 - p_2) = 0,2458.$$

Ответ: 0,2458.

Задача I.2.4.2. (15 баллов)

Гигантская сейшельская черепаха Эш решил перебраться поближе к цивилизации и обустроить для своих шестерых черепашат детскую площадку, где они бы могли играть. Для этого на морском дне поблизости от нового дома он нашел две доски и шесть колес разных диаметров от разнообразных агрегатов. Доски он положил так, чтобы двумя концами они упирались друг в друга и образовывали угол между собой в 60° . Внутри этого угла вложил колеса по возрастанию их радиусов, при этом каждое последующее колесо начиная со второго касается предыдущего и одновременно касается обеих досок. Первое колесо Эш уже заполнил песком, потратив на это 3 минуты. Во сколько раз больше времени ему понадобится, чтобы наносить песок во все 6 колес, если толщина песчаного слоя должна быть во всех колесах одинаковой?



- 63.
- 364.
- 66430.
- 1365.

Решение

Обозначим за v скорость заполнения песком колеса. Тогда, приняв во внимание, что толщина h песчаного слоя во всех колесах одинакова, получим выражение для времени наполнения первого колеса t_1 и всех колес вместе t_{tot} :

$$\begin{cases} t_1 = \frac{hS_1}{v}, \\ t_{tot} = \frac{hS_{tot}}{v}, \end{cases}$$

где S_1 и S_{tot} — площади первого колеса и всех колес вместе, соответственно.

Из этой пары соотношений становится очевидно, что для нахождения отношения времен, необходимо найти отношение площади всех шести колес к площади одного колеса (см. рис. I.2.1):

$$\frac{t_{tot}}{t_1} = \frac{S_{tot}}{S_1},$$

или, очевидно, отношение суммы квадратов радиусов всех окружностей к квадрату радиуса первой окружности. Рассмотрим две соседних окружности, при этом обозначим за R радиус большей окружности.

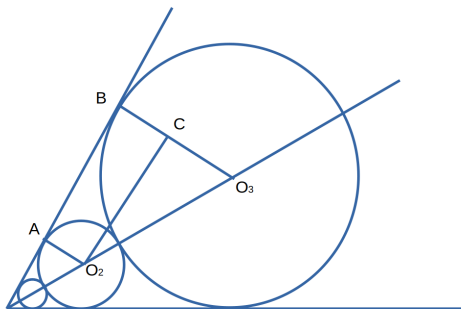


Рис. I.2.1: Чертеж к задаче

Опустим перпендикуляр из центра меньшей окружности на радиус большей окружности, проведенный в точку касания с одной из сторон данного угла. Получим прямоугольный треугольник с гипотенузой $R + r$, катетом $R - r$ и острым углом, равным 30° , противолежащим этому катету. Тогда $R + r = 2(R - r)$.

Отсюда находим, что $R = 3r$, независимо от того, какая это пара соседних окружностей: первая и вторая, вторая и третья и т. д.. Таким образом, получается, что радиусы окружностей являются геометрической прогрессией со знаменателем, равным, 3. Тогда получится, квадраты радиусов окружностей являются геометрической прогрессией со знаменателем $q = 9$. Тогда сумма квадратов радиусов будет всего лишь суммой 6 членов геометрической прогрессии:

$$\sum_{i=6}^n R_i^2 = r_1^2 \frac{q^6 - 1}{q - 1} = 66430r_1^2.$$

И тогда отношение:

$$\frac{t_{tot}}{t_1} = \frac{S_{tot}}{S_1} = \frac{\pi \sum_{i=6}^n R_i^2}{\pi r_1^2} = 66430.$$

Ответ: 66430.

Задача I.2.4.3. (20 баллов)

Найти решение неравенства:

$$\log_{\frac{1}{5}} \log_{11} \left(\frac{\sqrt{x^2 + 3} + x}{3} \right) < \log_5 \log_{\frac{1}{11}} (\sqrt{x^2 + 3} - x)$$

Для неправильной дроби с взаимно простыми числителем и знаменателем, являющейся самой левой границей решения, записать в ответ числитель, на единицу больший (например, если решением неравенства является множество решений $(\frac{15}{11}; 5) \cup [13; \infty)$, то в ответ нужно будет записать число 16).

Решение

Так как $\sqrt{x^2+3} - x = \frac{3}{\sqrt{x^2+3}+x}$, то:

$$\log_{\frac{1}{11}}(\sqrt{x^2+3} - x) = -\log_{11}(\sqrt{x^2+3} - x) = \log_{11}\left(\frac{\sqrt{x^2+3}+x}{3}\right).$$

С учетом приведенных выше преобразований можно перейти к равносильной системе неравенств:

$$\begin{cases} \frac{\sqrt{x^2+3}+x}{3} > 0, \\ \log_{11}\frac{\sqrt{x^2+3}+x}{3} > 0, \\ -\log_5\log_{11}\frac{\sqrt{x^2+3}+x}{3} < \log_5\log_{11}\frac{\sqrt{x^2+3}+x}{3}, \end{cases} \Rightarrow$$

$$\begin{cases} \frac{\sqrt{x^2+3}+x}{3} > 1, \\ \frac{1}{\log_{11}\frac{\sqrt{x^2+3}+x}{3}} < \log_{11}\frac{\sqrt{x^2+3}+x}{3}, \end{cases}$$

Воспользовавшись заменой переменных $t = \log_{11}\frac{\sqrt{x^2+3}+x}{3}$, решим нижнее неравенство:

$$\frac{1}{t} < t \Rightarrow \frac{1-t^2}{t} \Rightarrow \begin{cases} -1 < t < 0, \\ t > 1, \end{cases}$$

Вернемся к переменной x и запишем итоговую систему неравенств:

$$\begin{cases} \frac{\sqrt{x^2+3}+x}{3} > 1, \\ \begin{cases} -1 < \log_{11}\frac{\sqrt{x^2+3}+x}{3} < 0, \\ \log_{11}\frac{\sqrt{x^2+3}+x}{3} > 1, \end{cases} \Rightarrow \begin{cases} \frac{\sqrt{x^2+3}+x}{3} > 1, \\ \log_{11}\frac{\sqrt{x^2+3}+x}{3} > 1, \end{cases} \Rightarrow \begin{cases} \frac{\sqrt{x^2+3}+x}{3} > 1, \\ \frac{\sqrt{x^2+3}+x}{3} > 11, \end{cases} \Rightarrow \end{cases}$$

$$\Rightarrow \frac{\sqrt{x^2+3}+x}{3} > 11, \Rightarrow \begin{cases} 33 - x \geq 0, \\ x^2 + 3 > (33 - x)^2, \\ 33 - x < 0, \\ x^2 + 3 \geq 0, \end{cases}$$

$$\begin{cases} x \leq 33, \\ x > \frac{181}{11}, \\ x > 33 \end{cases}.$$

Окончательно решение будет выглядеть следующим образом: $x \in (\frac{181}{11}, \infty)$. И тогда в ответ запишем число 182.

Ответ: 182.

Задача I.2.4.4. (20 баллов)

Молодой изобретатель решил запустить в космос две ракеты и задал им вектора направления движения: $\vec{v}_1 = \left\{4; -8\log_{\frac{1}{2}}t; c\log_{\frac{1}{2}}t\right\}$ и $\vec{v}_2 = \left\{5; c; \log_{\frac{1}{2}}t\right\}$, где t — это время. Однако новоявленный Илон Маск забыл значения параметра c , который позволял бы сохранять угол между направлениями запуска после $\frac{1}{16}$ секунды острым, чтобы избежать столкновения ракет в дальнейшем. Помогите отыскать изобретателю набор значений параметра и записать в ответ среднее значение параметра, предполагая, что распределение значения искомого параметра — равномерное. Точность ответа — до тысячной.

Решение

Косинус острого угла положителен, следовательно, скалярное произведение данных векторов $\vec{v}_1 \cdot \vec{v}_2 > 0$ или $20 - 8c \log_{\frac{1}{2}} t + c \log_{\frac{1}{2}}^2 t > 0$. Обозначив $\log_{\frac{1}{2}} t = p$, получим $cp^2 - 8cp + 20 > 0$. Причем, так как $x \in [\frac{1}{16}; +\infty)$, неравенство должно выполняться для любых $t \in (0; 4]$.

Геометрическая интерпретация показывает два возможных варианта решения:

а) ветви ее направлены вверх и не пересекают оси t , из-за чего парабола полностью находится над осью t , и при любых значениях переменной t неравенство будет выполняться:

$$\begin{cases} c > 0, \\ D < 0, \end{cases} \Rightarrow \begin{cases} c > 0, \\ 16c^2 - 20c < 0, \end{cases} \Rightarrow \begin{cases} c > 0, \\ 4c(4c - 5) < 0, \end{cases} \Rightarrow c \in (0, \frac{5}{4}];$$

б) ветви ее направлены вверх, и парабола пересекает ось t , но при $t \in (0; 4]$ ветвь будет находиться выше оси t , а вершина параболы будет находиться левее 4:

$$\begin{cases} c > 0, \\ D \geq 0, \\ f(4) > 0, \\ -\frac{b}{4a} < 4 \end{cases} \Rightarrow \begin{cases} c > 0, \\ 16c^2 - 20c \geq 0, \\ f(4) > 0, \\ \frac{8c}{2c} < 4 \end{cases} \Rightarrow \begin{cases} c > 0, \\ 16c^2 - 20c \geq 0, \\ f(4) > 0, \\ \frac{4}{4} < 4 \end{cases} \Rightarrow x \in \emptyset.$$

При $c = 0$ неравенство также выполняется, поэтому мы можем включить значение в промежуток решения $c \in [0, \frac{5}{4}]$. И следовательно, среднее значение параметра будет равняться $\frac{5}{8} = 0,625$.

Ответ: 0,625.

Задача I.2.4.5. (30 баллов)

Потолок в аудитории для проведения лабораторных работ подпирают две металлические абсолютно прямые и очень тонкие балки (длина много больше толщины) из ультрасовременного и прочного сплава. Отношение длин этих балок равно k , а углы, которые балки образуют при этом с полом в аудитории, относятся как 2:3. Найдите наименьшее допустимое значение параметра k и запишите ответ с точностью до сотой.

Решение

Отношение длин отрезков можно найти из выражения для расстояния между параллельными полом и потолком:

$$H = l_1 \sin 3x = l_2 \sin 2x.$$

$$k = \frac{l_2}{l_1} = \frac{\sin 3x}{\sin 2x} = \frac{\sin(2x + x)}{\sin 2x} = \frac{\sin 2x \cos x + \sin x \cos 2x}{\sin 2x} = 2 \cos x - \frac{1}{2 \cos x}.$$

Проведем замену $\cos x = t$ и, учитывая, что максимальный угол, который может быть в данной системе, равен $3x = 90 \Rightarrow x \leq 30$, следовательно, $t \in [\frac{\sqrt{3}}{2}; 1]$.

Исследование производной функции $k(t) = 2t - \frac{1}{2t}$ ($f'(t) = 2 + \frac{1}{2t^2} > 0$) показывает, что в заданном промежутке значений переменной t функция является растущей и, следовательно, наименьшее значение отношения длин балок k будет соответствовать наименьшему значению параметра t :

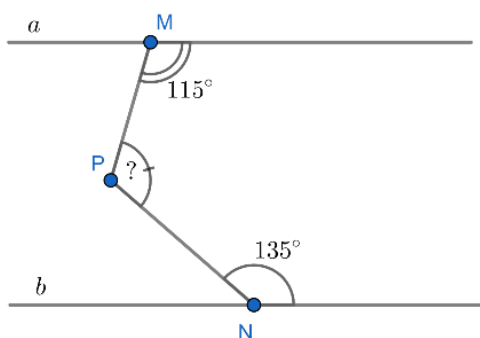
$$k\left(\frac{\sqrt{3}}{2}\right) = \sqrt{3} - \frac{1}{\sqrt{3}} = \frac{2\sqrt{3}}{3} = 1,15.$$

Ответ: 1,15.

Третья попытка. Задачи 8–9 класса

Задача I.2.5.1. (20 баллов)

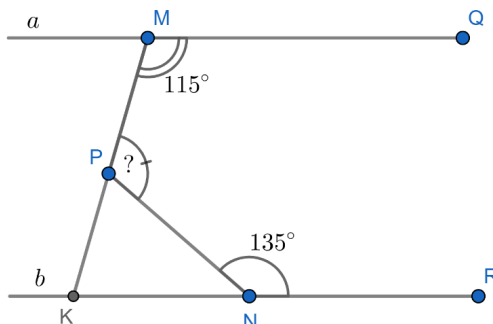
Известно, что прямые a и b параллельны. Найдите величину угла MPN , ответ запишите в градусах.



Решение

Достроим прямую MP до пересечения с прямой b в точке K .

Углы PNR и PNK в сумме дают 180° как смежные, поэтому $\angle PNK = 45^\circ$.



Сумма углов QMP и PKN также составляет 180° , поскольку эти углы — внутренние односторонние при параллельных прямых a и b и секущей MK , отсюда $\angle PKN = 65^\circ$.

По теореме о внешнем угле треугольника:

$$\angle MPN = \angle KNP + \angle NKP = 45^\circ + 65^\circ = 110^\circ.$$

Ответ: 110.

Задача I.2.5.2. (20 баллов)

Сергей поделил натуральное двузначное число на 8 с остатком, а Саша поделил это же число на 5 и получил остаток в 2 раза больше, чем у Сергея. Найдите сумму всех возможных чисел, которые могли делить Сергей и Саша.

Решение

Пусть остаток от деления двузначного числа n у Сергея равен a , тогда остаток Саши — $2a$. Поскольку $2a < 5$ то a может равняться 1 или 2 (случай $a = 0$ невозможен, т. к. второй остаток должен быть больше первого). Разберем эти случаи:

1. $a = 1$. Тогда $n = 8k + 1 = 5m + 2$, где k и m — некоторые натуральные числа. Следовательно, число n имеет вид $40p + 17$ где $p = 0, 1$ или 2 . Таким образом, n может принимать значения 17, 57 и 97. Их сумма равна 171.
2. $a = 2$. Тогда $n = 8k + 2 = 5m + 4$, где k и m — некоторые натуральные числа. Следовательно, число n имеет вид $40p + 34$ где $p = 0$ или 1 . Таким образом, n может принимать значения 34 и 74. Их сумма равна 108.

Сумма всевозможных значений из обоих случаев составляет $171 + 108 = 279$.

Ответ: 279.

Задача I.2.5.3. (20 баллов)

Спортивный магазин провел акцию «Не имей сто рублей, а имей сто друзей». Она заключалась в следующем: если покупатель, который приобрел велосипед «Дружок», привел 5 друзей, которые приобретали такой же велосипед, то приведенному деньги возвращались. За время проведения акции 25 покупателей пришли сами, остальных привели друзья. Некоторые привели ровно по 5 новых покупателей, а остальные 217 не привели ни одного. Сколько участников акции ездят на велосипеде бесплатно?

Решение

Каждый из x «счастливчиков» привел по 5 друзей. Тогда «приведенных» клиентов было $5x$, а еще 25 пришли сами, значит, всего покупателей было $25 + 5x$. С другой стороны, x человек привели новых клиентов, а 217 — не привели, то есть всего участников акции было $x + 217$. Итак, $25 + 5x = x + 217$ откуда $x = 48$.

Ответ: 48.

Задача I.2.5.4. (20 баллов)

Каждый из графиков функций $y = x^2 + ax + b$ и $y = x^2 + bx + a$ пересекает ось Ox в двух различных точках, а график $y = (x^2 + ax + b)(x^2 + bx + a)$ имеет с осью Ox ровно три общие точки. Найдите наименьшее возможное значение суммы всех координат этих трех точек.

Решение

Координаты по оси Oy точек пересечения с осью Ox равны 0, поэтому достаточно рассмотреть абсциссы точек пересечения, то есть корни многочленов. Из условия следует, что трехчлены $x^2 + ax + b$ и $x^2 + bx + a$ имеют общий корень x_0 , а также отличные от него корни x_1 и x_2 соответственно; отсюда, в частности, следует, что $a \neq b$. Общий корень является также корнем разности этих трехчленов, то есть $(a - b)(x_0 - 1) = 0$. Таким образом, $x_0 = 1$. Подставляя этот корень в любой из трехчленов, получаем $1 + a + b = 0$. По теореме Виета $x_0 + x_1 = -a$, $x_0 + x_2 = -b$, откуда $x_0 + x_1 + x_2 = -x_0 - a - b = -(1 + a + b) = 0$.

Ответ: 0.

Задача I.2.5.5. (20 баллов)

Три семьи с детьми (папа, мама и ребенок) отправились в поход. Сколькими способами они могут так построиться для безопасного прохождения болотистого участка, чтобы направляющим и замыкающим были взрослые, и двое детей не шли друг за другом?

Решение

Расставим сначала взрослых. Они могут стать друг за другом 6! способами. При каждом способе есть 5 мест между двумя взрослыми, на которые можно ставить детей. Занумеруем детей как угодно, результат не зависит от того, кого из них мы раньше поставим на место. Место для первого ребенка можно выбрать любым из 5 способов, для второго — любым из 4 оставшихся, для третьего — из 3. Всего $6! \cdot 5 \cdot 4 \cdot 3 = 43200$ способов построения.

Ответ: 43200.

Третья попытка. Задачи 10–11 класса**Задача I.2.6.1. (15 баллов)**

Каким должен быть наибольший объем сока, заливаемого в сосуд в виде конуса, если сумма длины образующей и радиуса основания этого сосуда равны 15 см?

Ответ в см^3 записать с точностью до тысячной.

Решение

Обозначим образующую конуса как l , радиус — R , высоту — h . Согласно условию задачи и выражению, связывающему эти три параметра конуса, составим систему уравнений с целью выразить радиус конуса через его высоту:

$$\begin{cases} l + R = 15, \\ l^2 - R^2 = h^2, \end{cases} \Rightarrow \begin{cases} l + R = 15, \\ (l + R) \cdot (l - R) = h^2, \end{cases} \Rightarrow \begin{cases} l + R = 15, \\ 15(l - R) = h^2, \end{cases}$$

$$\Rightarrow \begin{cases} l + R = 15, \\ l - R = \frac{h^2}{15}, \end{cases} \text{ откуда } R = \frac{225 - h^2}{30}.$$

Подставив это выражение для объема конуса, можно получить следующее выражение:

$$V = \frac{1}{3}\pi R^2 h = \frac{\pi}{2700}(h^5 - 450h^3 + 50625h).$$

С помощью производной исследуем промежутки знакопостоянства функции объема, памятуя, что $0 < h < 15$:

$$V' = \frac{\pi}{2700}(5h^4 - 1350h^2 + 50625) = 0.$$

В пределах $0 < h < 15$ есть только одна критическая точка $h = 3\sqrt{5}$, левее которой производная положительная, а правее — отрицательна. Это говорит о том, что в данной точке есть экстремум — максимум. Тогда $R = \frac{225 - h^2}{30} = 6$ и объем будет равен $36\sqrt{5}\pi$ или, округленно, 252,893.

Ответ: 252,893.

Задача I.2.6.2. (15 баллов)

Незадачливый пользователь во время очередной вылазки в интернет нечаянно загрузил себе на компьютер 14 вирусов: 8 рекламных и 6 троянских. Через два часа после окончания интернет-серфинга автоматически должна запуститься антивирусная проверка, которая моментально удаляет все вредоносные программы. Однако за час до вылазки в интернет этот активный пользователь покопался в открытой части кода антивирусника, вследствие чего программа стала удалять случайным образом только некоторое количество вредоносного ПО. Какова вероятность того, что в результате автоматического запуска будут удалены 7 рекламных вирусов и 2 троянских? Ответ записать с точностью до сотой.

Решение

Общее количество вариантов выбора 9 вирусов из 14 равно

$$n = C_{14}^9 = \frac{14!}{5! \cdot 9!} = 11 \cdot 13 \cdot 14.$$

Количество вариантов, когда возникает событие, благоприятствующее условию

задачи и удалению с себя 7 рекламных вирусов и 2 троянских, равно:

$$m = C_8^7 \cdot C_6^2 = \frac{8!}{7! \cdot 1!} \cdot \frac{6!}{2! \cdot 4!} = 3 \cdot 5 \cdot 8.$$

Тогда вероятность будет равна:

$$p = \frac{m}{n} = \frac{3 \cdot 5 \cdot 8}{11 \cdot 13 \cdot 14} = \frac{60}{1001} = 0,06.$$

Ответ: 0,06.

Задача I.2.6.3. (30 баллов)

На окружность нанесли 2020 меток и поставили около каждой из них натуральные числа от 1 до 2020. После этого начали последовательно вычеркивать каждое второе число, т. е. числа 2, 4, 6, 8, ..., 2018, 2020, 3, 7 и т. д. до тех пор, пока не осталось одно единственное число. Назовите оставшееся число.

Решение

Задача представляет собой классический вариант задачи Флавия с $n = 2020$ элементами. В общем случае, если $n = 2^k + m = 2^{10} + 996$, где $m < 2^k$, то останется число $f(n) = 2m + 1 = 2 \cdot 996 + 1 = 1993$.

Ответ: 1993.

Задача I.2.6.4. (20 баллов)

Баба Яга в Крещенскую ночь решила погадать на жениха. Но поскольку вся нечисть в это время занята в других гаданиях, Баба Яга решила привлечь Кошечку Бессмертного и Лешего. Гадать она решила так: ровно в полночь Кошечка начнет рисовать концентрические окружности с центрами в некоторой точке и радиусами, образующими бесконечную геометрическую прогрессию. За выбор координат центра окружностей и параметров прогрессии ответственным она назначила Лешего. Если прямая $3x - 4y + 13 = 0$ коснется одной из заданных окружностей, то баба Яга выйдет замуж, причем через количество веков, равное номеру окружности. Ответьте, выйдет ли замуж Баба Яга, и если да, то через сколько веков. Леший указал центр концентрических окружностей в точке $(5; 3)$, а в прогрессии первый член $\frac{1}{5}$ и знаменатель прогрессии, равный 2.

Если гадание предскажет, что сказочной героине не суждено найти жениха, то в ответ запишите -1.

Решение

Уравнения окружностей:

$$(x - 5)^2 + (y - 3)^2 = R_n^2,$$

где $R_n^2 = \frac{1}{5} \cdot 2^{n-1}$. Найдем общие точки окружности и прямой:

$$\begin{cases} x^2 + 10x + 25 + y^2 - 6x + 9 = R^2, \\ y = \frac{1}{4}(3x + 13), \end{cases}$$

$$\Rightarrow \begin{cases} y = \frac{1}{4}(3x + 13), \\ x^2 + 10x + 25 + \frac{1}{6}(9x^2 + 78x + 169) - \frac{6}{4}(3x + 13) = R^2, \end{cases}$$

$$\Rightarrow 25x^2 - 154x + 401 - 16R^2 = 0$$

Чтобы общая точка была одна, необходимо $D = 0$:

$$23716 - 100(401 - 16R^2) = 0 \text{ или } 1600R^2 = 16384 \Rightarrow R = \frac{16}{5}.$$

$$\frac{16}{5} = \frac{1}{5} \cdot 2^{n-1} \Rightarrow n = 5.$$

Таким образом, можно сказать, что Бабе яге посчастливится выйти замуж через 5 веков.

Ответ: 5.

Задача I.2.6.5. (20 баллов)

Найти произведение всех пар чисел $(x; y)$ для каждой из которых выполняются одновременно два условия:

$$\begin{cases} 2^{|x^2-2x-3|-\log_2 3} = 3^{-y-4}, \\ 4|y| - |y-1| + (y+3)^2 \leq 8 \end{cases}$$

Например, если решением задачи будут две пары чисел $(\frac{1}{2}; 3)$ и $(-1; 2)$, то в ответ надо будет записать следующее число: $\frac{1}{2} \cdot 3 \cdot (-1) \cdot 2 = -3$.

Решение

Оценим левую и правую части уравнения системы:

$$x^2 - 2x - 3 = (x-1)^2 - 4 \geq 4 \Rightarrow |x^2 - 2x - 3| \leq 4 \Rightarrow$$

$$1 \leq 2^{|x^2-2x-3|} \leq 16 \Rightarrow \frac{1}{3} \leq \frac{1}{3} \cdot 2^{|x^2-2x-3|} \leq \frac{16}{3}.$$

Решение неравенства системы сводится к решению совокупности трех систем.

$$\left[\begin{cases} y \leq 0, \\ y^2 + 3y \leq 0, \\ 0 \leq y \leq 1, \\ y^2 + 11y \leq 0, \\ y \geq 1, \\ y^2 + 9y + 2 \leq 0, \end{cases} \Rightarrow \left[\begin{cases} y \leq 0, \\ -3 \leq y \leq 0, \\ 0 \leq y \leq 1, \\ -11 \leq y \leq 0, \\ y \geq 1, \\ \frac{-9-\sqrt{73}}{2} \leq y \leq \frac{-9+\sqrt{73}}{2}, \end{cases} \Rightarrow \left[\begin{cases} -3 \leq y \leq 0, \\ \emptyset \\ \emptyset \end{cases} \right.$$

$$\Rightarrow y \in [-3; 0]$$

Таким образом правая часть уравнения допускает оценку:

$$\frac{1}{81} \leq 3^{-y-4} \leq \frac{1}{3}$$

Так как наименьшее значение левой части уравнения системы совпадает с наибольшим значением правой части системы, то имеет место:

$$\begin{cases} 3^{-y-4} = \frac{1}{3}, \\ 2^{|x^2-2x-3|-\log_2 3} = \frac{1}{3}, \end{cases} \Rightarrow \begin{cases} -y-4 = -1, \\ |x^2-2x-3| = 0, \end{cases} \Rightarrow \begin{cases} y = -3, \\ \begin{cases} x = -1, \\ x = 3, \end{cases} \end{cases} \Rightarrow \begin{cases} y = -3, \\ x = -1, \\ y = -3, \\ x = 3. \end{cases}$$

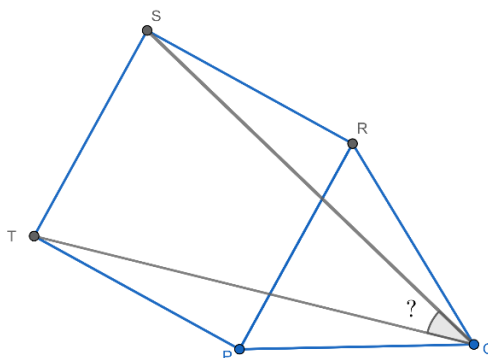
Таким образом произведение всех пар чисел будет равняться -27 .

Ответ: -27.

Четвертая попытка. Задачи 8–9 класса

Задача I.2.7.1. (20 баллов)

На стороне PR равностороннего треугольника PQR во внешнюю сторону построен квадрат $PRST$. Найдите величину угла SQT , ответ запишите в градусах.



Решение

Углы TPQ и SRQ равны $90^\circ + 60^\circ = 150^\circ$. Треугольники PQT и SRQ равнобедренные, поэтому $\angle PQT = \angle RQS = \frac{180^\circ - 150^\circ}{2} = 15^\circ$.

Окончательно получаем: $\angle SQT = \angle PQR - \angle RQT - \angle RQS = 60^\circ - 2 \cdot 15^\circ = 30^\circ$.

Ответ: 30.

Задача I.2.7.2. (20 баллов)

Граф Калиостро поделил натуральное двузначное число на 9 с остатком, а Маргадон поделил это же число на 7 и получил остаток в 3 раза больше, чем у Калиостро. Найдите сумму всех возможных чисел, которые могли делить граф Калиостро и Маргадон.

Решение

Пусть остаток от деления двузначного числа n у Калиостро равен a , тогда остаток Маргадона — $3a$. Поскольку $3a < 7$, то a может быть 1 или 2 (случай невозможен, т. к. второй остаток должен быть больше первого). Разберем эти случаи:

1. $a = 1$. Тогда $n = 9k + 1 = 7m + 3$, где k и m — некоторые натуральные числа. Тогда число n имеет вид $63p + 10$, $p = 0, 1$. $n = 10, 73$. Их сумма 83.
2. $a = 2$. Тогда $n = 9k + 2 = 7m + 6$, где k и m — некоторые натуральные числа. Тогда число n имеет вид $63p + 20$, $p = 0, 1$. $n = 20, 83$. Их сумма 103.

Сумма чисел из обоих случаев равна 186.

Ответ: 186.

Задача I.2.7.3. (20 баллов)

У Маши были три корзинки с конфетами. Сначала она переложила из первой корзинки во вторую и третью столько конфет, сколько в каждой из них было изначально. После этого из второй корзинки Маша переложила в третью и первую такое количество конфет, которое в каждой из них имелось на тот момент. Наконец, она переложила из третьей корзинки в первую и вторую столько конфет, сколько там оказалось к этому времени. В результате всех перекладываний в первой корзинке оказалось 160 конфет, во второй 120, а в третьей — 40. Какое количество конфет изначально было во **второй** корзинке?

Решение

Будем записывать количества конфет тройками чисел (первая корзинка; вторая корзинка; третья корзинка). Начнем решать задачу с конца, тогда после третьей операции была тройка (160; 120; 40). Эта тройка образовалась при удвоении числа конфет в первой и второй корзинках, поэтому после второй операции была тройка (80; 60; 180). В свою очередь, эта тройка образовалась при удвоении числа конфет в первой и третьей корзинках; следовательно, после первой операции было (40; 190; 90). Наконец, эта тройка получилась в результате удвоения количества конфет во второй и третьей корзинках. Таким образом, изначально была тройка (180; 95; 45) и во второй корзинке было 95 конфет.

Ответ: 95.

Задача I.2.7.4. (20 баллов)

Буратино вычислил значения приведенного квадратного трехчлена при двух значениях: $x = 2$ и $x = 72$ и сложил их. Пьеро вычислил значения другого приведенного квадратного трехчлена при тех же значениях, сложил их и получил ту же сумму. При каком значении x трехчлены Буратино и Пьеро равны?

У приведенного многочлена коэффициент при наибольшей степени равен единице.

Решение

Пусть у Буратино был многочлен $P(x) = x^2 + ax + b$, а у Пьеро — $Q(x) = x^2 + cx + d$.

По условию $2^2 + 2a + b + 72^2 + 72a + b = 2^2 + 2c + d + 72^2 + 72c + d$.

Откуда $74a + 2b = 74c + 2d$, то есть:

$$37(a - c) = d - b \quad (\text{I.2.2})$$

Если $a = c$, то и $d = b$, то есть $P(x)$ совпадает с $Q(x)$, что по условию не так.

Пусть x_0 — искомое значения, то есть $P(x_0) = Q(x_0)$.

Получаем, что $x_0^2 + ax_0 + b = x_0^2 + cx_0 + d$. Откуда $x_0 = \frac{d-b}{a-c} = 37$, учитывая (I.2.2).

Ответ: 37.

Задача I.2.7.5. (20 баллов)

Если выбрать три произвольных различных числа из множества натуральных чисел $\{1, 2, 3, \dots, n-1, n\}$, вероятность того, что числа являются последовательными, равна $\frac{1}{222}$. Найдите n .

Решение

Всего способов выбрать три числа из n различных чисел:

$$C_n^3 = \frac{n(n-1)(n-2)}{6}$$

Количество троек последовательных чисел из n равно $n-2$. Таким образом, данная вероятность:

$$\frac{1}{222} = \frac{n-2}{C_n^3} = \frac{6}{n(n-1)}.$$

То есть $n = 37$.

Ответ: 37.

Четвертая попытка. Задачи 10–11 класса**Задача I.2.8.1. (15 баллов)**

Найти наибольшее значение выражения:

$$\frac{x^2 + 7xy}{x + y} + \frac{y^2 + 7yz}{y + z} + \frac{z^2 + 7zx}{z + x}$$

для положительных значений переменных, сумма которых равняется единице.

Решение

Воспользуемся оценкой, которая выполняется для всех положительных переменных a и b :

$$\frac{2ab}{a+b} \leq \frac{a+b}{2}.$$

Преобразуем первое слагаемое в сумме:

$$\frac{x^2 + 7xy}{x+y} = \frac{x^2 + xy + 6xy}{x+y} = x + \frac{6xy}{x+y} = x + 3 \frac{2xy}{x+y} \leq x + 3 \frac{x+y}{2} = \frac{5}{2}x + \frac{3}{2}y.$$

По аналогии с предыдущим слагаемым получим:

$$\frac{y^2 + 7yz}{y+z} = \frac{5}{2}y + \frac{3}{2}z,$$

$$\frac{z^2 + 7zx}{z+x} = \frac{5}{2}z + \frac{3}{2}x,$$

и тогда сумма полученных слагаемых будет равняться $4(x+y+z) = 4$.

Ответ: 4.

Задача I.2.8.2. (15 баллов)

Согласно правилам некоторой олимпиады по математике, участники, вышедшие в финальный тур, могут получить любое количество призов из призового фонда. Т.е. один финалист, обладающий достаточными знаниями и навыками решения задач, может выиграть и 2, и 4, и, даже, все призы из призового фонда. Сколькими способами будет распределить награды между финалистами, если призовой фонд олимпиады по математике состоит из 6 ноутбуков, а в финал вышли 8 участников? Во скольких из всех способов восьмой участник получит не менее двух ноутбуков? Ответ записать через запятую, без пробелов.

Решение

1. Каждый ноутбук можно разыграть восемью способами, поэтому всего имеется $8^6 = 262144$ способов.
2. Чтобы определить, в скольких случаях 8-ому участнику достанется не менее двух ноутбуков, найдем, сколькими способами можно получить восьмому участнику один ноутбук и не получить ни одного. Если восьмому участнику не досталось ни одного ноутбука, то остальным участникам досталось каждому по 1 ноутбуку, что оценивается в 7^6 способов; если же один из ноутбуков все-таки достался восьмому участнику, то остальным участникам досталось 5 ноутбуков, что можно оценить в $6 \cdot 7^5$ способов распределения наград. В остальных $8^6 - 7^6 - 6 \cdot 7^5$ случаях восьмому участнику достанется не менее двух ноутбуков.

Ответ: 262144,43653.

Задача I.2.8.3. (20 баллов)

Внутри правильного тетраэдра с ребром 10 летает пчела. Каким должен быть ее наименьший замкнутый путь, если пчела должна побывать на всех гранях тетраэдра?

Ответ округлить до тысячной.

Решение

Очевидно, что траектория движения пчелы представляет собой замкнутый пространственный четырехугольник $KLMN$, вершины которого расположены на гранях тетраэдра $ABCD$ (см. чертеж (I.2.2а)).

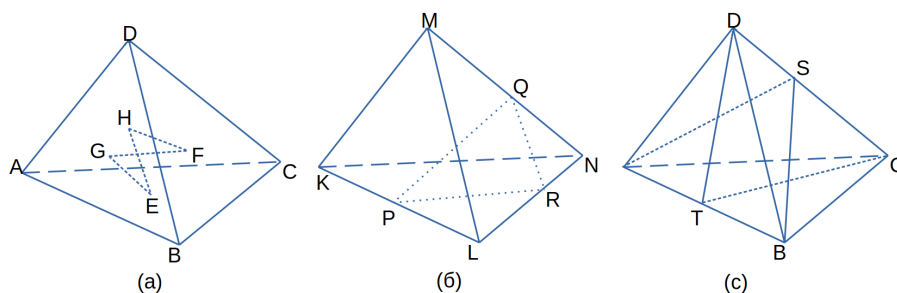


Рис. I.2.2: Чертеж к задаче

Рассмотрим тетраэдр $ABCD$. Пусть пчела побывала на каждой из граней тетраэдра и вернулась в исходную точку. Без ограничения общности можно считать, что она сначала побывала на грани ABC , потом — на грани BCD , затем — на DAB , и, наконец, на ACD . Обозначим соответствующие точки на гранях, в которых побывала муха, через E, F, G и H . Ясно, что минимальное расстояние, которое пчела могла пролететь, равно периметру пространственного четырехугольника $EFGH$.

1. Проведем через DC плоскость, перпендикулярную AB (плоскость симметрии тетраэдра $ABCD$) и рассмотрим четырехугольник $E_1F_1G_1H_1$, симметричный $EFGH$ относительно этой плоскости. (Вершины E_1 и G_1 останутся на тех же гранях, что E и G соответственно, F_1 попадет на одну грань с H , а H_1 — на одну грань с F .) Периметры четырехугольников $EFGH$ и $E_1F_1G_1H_1$ равны.

Лемма. Рассмотрим любой пространственный четырехугольник $KLMN$ (см. чертеж (I.2.2б)). Пусть P и Q — середины сторон KL и MN . Тогда $PQ \leq (KN + LM)$.

Доказательство. Обозначим через R середину диагонали LN . Имеем $PR = \frac{1}{2}KN$, $RQ = \frac{1}{2}LM$. Таким образом, $PQ < PR + RQ = \frac{1}{2}(KN + LM)$.

Лемма доказана.

2. Обозначим через E_2, F_2, G_2 и H_2 середины отрезков EE_1, FH_1, GG_1 и HF_1 соответственно. Вершины этого четырехугольника тоже лежат на гранях тетраэдра, кроме того, периметр четырехугольника $E_2F_2G_2H_2$ не больше периметра $EFGH$. Кроме того, вершины E_2 и G_2 (середины EE_1 и GG_1) будут лежать в плоскости симметрии тетраэдра, проходящей через CD , т. е. на медианах CT и DT граней ABC и ABD .

Исходя из четырехугольника $E_2F_2G_2H_2$, точно так же построим $E_3F_3G_3H_3$, симметричный ему относительно плоскости симметрии тетраэдра, проходящей

через AB , а затем, взяв середины отрезков, соединяющих вершины этих четырехугольников, лежащих в одной грани, получим $E_4F_4G_4H_4$, все вершины которого лежат в объединении двух плоскостей симметрии тетраэдра $ABCD$, проходящих через CD и AB . Иными словами, вершины E_4 и G_4 лежат на отрезках CT и DT , а вершины F_4 и H_4 — на медианах AS и BS граней ACD и BCD .

При этом периметр $E_4F_4G_4H_4$ не превосходит периметра $EFGH$. Значит, периметр $EFGH$ не меньше, чем $4d$, где d — расстояние между прямыми CT и BS .

Осталось построить путь длины $4d$ и найти d . Пусть E_0 и F_0 — основания общего перпендикуляра к прямым CT и BS , причем E_0 лежит на CT , а F_0 — на BS . Обозначим через G_0 точку, симметричную точке E_0 относительно плоскости ABS . Из симметрии ясно, что F_0G_0 — общий перпендикуляр к прямым BS и DT . Аналогично строится точка H_0 , при этом G_0H_0 и H_0E_0 являются общими перпендикулярами соответственно к DT и AS и к AS и CT . Значит, периметр четырехугольника $E_0F_0G_0H_0$ равен $4d$. Заметим, что нужно еще проверить, что основания этих общих перпендикуляров лежат на гранях тетраэдра, а не на их продолжениях, это будет сделано ниже (нам еще нужно вычислить d).

3. Проведем через AB плоскость, перпендикулярную CT и спроецируем на нее наш тетраэдр. Получим треугольник ABD' (см. чертеж 1.2.3), в котором $AB = a$, $DT = \sqrt{\frac{2}{3}}a$ (по формуле для длины высоты правильного тетраэдра).

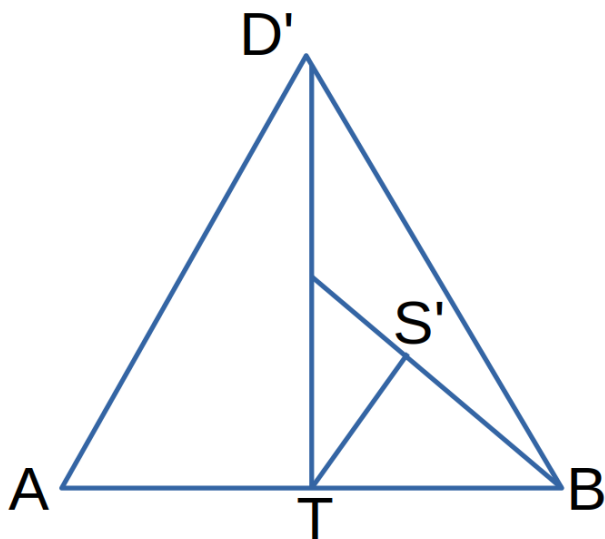


Рис. 1.2.3: Чертеж к задаче

Точка S перейдет в S' — середину $D'T$. Искомое расстояние d равно расстоянию от точки T до прямой BS' (поскольку общий перпендикуляр параллелен плоскости проекции). Кроме того, очевидно, что основание перпендикуляра, опущенного из точки T на прямую BS' , лежит на отрезке BS' , а не на его продолжении, значит, точка F_0 лежит на отрезке BS . Аналогично доказывается, что и остальные вершины четырехугольника лежат на медианах, а не на их продолжениях.

В прямоугольном треугольнике BTS' известны катеты $BT = \frac{a}{2}$, $TS' = \frac{a}{2}\sqrt{\frac{2}{3}}$.

Значит, $BS' = \frac{a}{2}\sqrt{\frac{5}{3}}$; $d = \frac{BT \cdot TS'}{BS'} = \frac{a}{\sqrt{10}}$. И тогда искомый периметр равен

$$4\frac{a}{\sqrt{10}} = 4\sqrt{10} \approx 12,649.$$

Ответ: 12,649.

Задача I.2.8.4. (20 баллов)

В подвале пожилой Дамы хранится 15 корзин с яблоками, причем количество неиспорченных яблок в каждой корзине соответствует номеру этой корзины. Дети, живущие по соседству, решили помочь ей перебрать фрукты, но чтобы работа не была такой скучной, было принято следующее решение: за одно переукладывание добавлять в корзину яблоки можно столько яблок, сколько их уже есть в ней, из любой другой корзины, в которой достаточно для этого яблок. Каким может быть наибольшее количество яблок в одной корзине?

Решение

Всего в корзинах $k = 15(15 + 1)/2 = 120$ яблок. Посмотрим, как происходит операция обратная к досыпанию яблок. Она выглядит так: яблоки в какой-то корзине делят пополам, после чего две половинки или оказываются в отдельных корзинах, или содержимое одной из них добавляют в какую-то не пустую корзину (а другую половинку оставляют в отдельной корзине).

В результате таких операций из корзины с k яблоками можно получить только корзины, в которых $mk/2^S$ яблок при некоторых натуральных m и S (поскольку есть только деление на 2 и сложение). Значит, в таком виде, в частности, должно представляться и число 1, но число 1 не представляется в таком виде, так как k не является степенью двойки.

При четном k в одну корзину нельзя собрать не только k яблок, но и $k - 1$ яблок, так как $k - 1$ нечетное число, а последняя операция переукладывания могла быть только удвоением.

Таким образом, в одну корзину можно добавить $k - 2 = 118$ яблок. Осталось убедиться, что эти оценки сверху достижимы. Схема алгоритма достижения такова. Имея две корзины с a и b яблоками, где $a + b$ нечетно и $a = 2^m \bmod (a + b)$ при некотором натуральном m , мы можем получить в этих корзинах после «переукладываний» 1 и $a + b - 2$ яблок, затем 2 и $a + b - 2$. Далее, воспользовавшись корзинкой с 2 яблоками, «извлечем» еще 2 яблока из остальных корзин и получим в этих двух корзинах 4 и $a + b - 2$ яблок и т. д. Таким образом, начав с корзин, в которых 1 и 2 яблока, будем выкладывать из остальных по 2 яблока, пока во всех остальных корзинах в сумме не останется 1 или 0 яблок.

Ответ: 118.

Задача I.2.8.5. (30 баллов)

Астрономическая обсерватория ведет наблюдения за космическими телами, движущимися в некоторой области наблюдаемого космического пространства, и результаты наблюдений за двумя кометами, метеорным потоком и группой метеороидов заносит в карту. Траектории комет представляют из себя две параболы Π_1 и Π_2 ,

причем траектория второй параболы Π_2 симметрична траектории первой кометы Π_1 с уравнением $y = ax^2$, $a < 0$ относительно точки $N(b; ab^2)$, где $b > 0$. Метеорный поток, движущийся в той же области по прямой, пересекает каждую из траекторий ровно в одной точке: Π_1 — в точке B_1 , Π_2 — в точке B_2 так, что угол B_1B_2N — прямой. Группа метеороидов движется по прямой траектории, представляющей из себя касательную к траектории первой параболы, проведенной в точке B_1 и одновременно пересекающей отрезок B_2N в точке L .

Определите отношение параметров $\frac{a}{b}$, если длина отрезка B_1L минимальна, а площадь треугольника B_1B_2N равна $\frac{1}{3}$.

Решение

Известно, что прямые пересекают параболу ровно в одной точке только когда она параллельна оси OY , либо касается этой параболы. Оси парабол параллельны оси OY (см. чертеж 1.2.4), поэтому либо $B_1B_2 \parallel OY$, либо B_1B_2 — общие касательные к параболам, что невозможно, так как единственная общая касательная данных парабол проходит через точку N . Отсюда $B_1B_2 \parallel OY$.

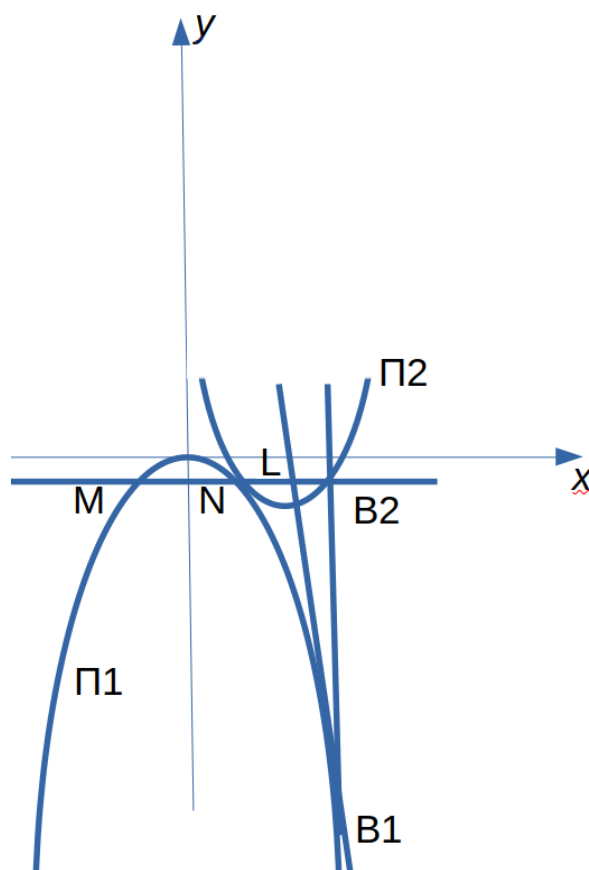


Рис. 1.2.4: Чертеж к задаче

Таким образом, $NB_2 \parallel OX$, поэтому $MN = 2b_1$ если точка M — точка пересечения NB_2 с Π_1 , то и из симметрии парабол следует $NB_2 = 2b$, откуда $B_2(3b; ab)$, $B_1(3b, 9ab^2)$ — координаты точек B_2 и B_1 . Тогда касательная B_1L имеет уравнение $y = 6abx - 9ab^2$ и пересекает прямую NB_2 уравнением $y = ab^2$ в точке $L(\frac{5}{3}b; ab^2)$. Отсюда получаем $NL = \frac{2}{3}b$, $B_2L = \frac{4}{3}b$ т. е. $NL : B_2L = 1 : 2$. Далее из равенства $\frac{1}{3} =$

$S_{B_1 B_2 N} = \frac{1}{2} N B_2 \cdot B_1 B_2 = 8|a|b^2$ следует, что $|a| = \frac{1}{24b^2}$. Поэтому $B_1 L^2 = B_1 B_2^2 + L B_2^2 = 64a^2 b^2 + \frac{16}{9} b^2 = \frac{1}{9} \left(\frac{1}{b^2} + 16b^2 \right)$. Наименьшее значение этой функции достигается при $b^2 = \frac{1}{4}$, т. е. $b = \frac{1}{2}$ и тогда $a = -\frac{1}{3}$. В таком случае отношение параметров $\frac{a}{b} = -1,5$.

Ответ: -1,5.

Второй отборочный этап

Описание задачи

Люди научили машины читать, но способны ли те понять прочитанное? Вам предстоит доказать, что машины способны не только прочесть текст и понять, что там происходит, но и правильно ответить на вопросы, которые требуют построения причинно-следственных связей.

Ниже вы найдете набор данных, состоящий из текстов и вопросов к ним с разными вариантами ответа. Ваша задача: написать алгоритм, который определит, какие из ответов верные.

Основные принципы:

- Ответ содержится в нескольких предложениях, а не в одном!
- Ответ не четко (дословно) прописан в тексте. Полное соответствие ответа в изначальном параграфе найти нельзя.
- Количество правильных ответов может быть любым.

Подобные системы нужны для автоматизации колл-центров и сервисов клиентской поддержки, в чат-ботах и поисковых машинах. Подробнее об AGI-based вопросно-ответных системах, их использовании и методах построения смотрите в запись вебинара от разработчиков задачи (<https://www.youtube.com/watch?v=guPAeY10CEc>).

Также вам доступен вебинар с разбором базового решения (<https://youtu.be/Jbj8R5sZzBk>) и дополнительный вебинар (<https://youtu.be/xDUGBm0zbf4>), где рассказывается о новых способах решения задачи бинарной классификации в контексте NLP и разобраны, как работают трансформеры, формирующие из слов вектор, и как все-таки сформировать подтверждение в нужном формате для загрузки на платформу.

Данные

Для решения задачи вам предоставлен датасет, в котором около 6 000 вопросов для более чем 800 текстов из 5 разных областей:

- Тексты начальной школы
- Новости
- Художественные тексты
- Сказки
- Краткое содержание сериалов

В архиве вы найдете 3 файла:

- `train` и `val` — для обучения модели и настройки параметров;
- `test` — проверочный файл, в котором нужно сделать предсказание и загрузить на платформу для проверки.

Данные для решения задачи: <https://russiansuperglue.com/tasks/download/>

Формат решений

Задача представляет собой бинарную классификацию (True/False).

В систему для проверки необходимо предоставить jsonl-файл с метками ответов: если ответ верен — 1, если нет — 0. Качество решения определяется, как среднее значение точности и полноты ответов — F1 average.

Также вам доступно базовое решение от разработчиков задачи (<https://github.com/AI-Front/NTI>).

Пример выходных данных:

```
{
  "id": 397,
  "text": "(1) Мужская сборная команда Норвегии по биатлону в рамках этапа Кубка
  → мира в немецком Оберхофе выиграла эстафетную гонку. (2) Вторыми стали
  → французы, а бронзу получила немецкая команда. (3) Российские биатлонисты не
  → смогли побороться даже за четвертое место, отстав от норвежцев более чем на
  → две минуты. (4) Это худший результат сборной России в текущем сезоне. (5)
  → Четвертыми в Оберхофе стали австрийцы. (6) В составе сборной Норвегии на
  → четвертый этап вышел легендарный Уле-Эйнар Бьорндален. (7) Впрочем, Норвегия с
  → самого начала гонки была в числе лидеров, успешно проведя все четыре этапа.
  → (8) За сборную России в Оберхофе выступали Иван Черезов, Антон Шипулин,
  → Евгений Устюгов и Максим Чудов. (9) Гонка не задалась уже с самого начала:
  → если на стрельбе из положения лежа Черезов был точен, то из положения стоя он
  → допустил несколько промахов, в результате чего ему пришлось бежать один
  → дополнительный круг. (10) После этого отставание российской команды от
  → соперников только увеличивалось. (11) Напомним, что днем ранее российские
  → биатлонистки выиграла свою эстафету. (12) В составе сборной России выступали
  → Анна Богалий-Титовец, Анна Булыгина, Ольга Медведцева и Светлана Слепцова.
  → (13) Они опередили своих основных соперниц - немок - всего на 0,3 секунды.",
  "questions": [
    {
      "question": "На сколько секунд женская команда опередила своих соперниц?",
      "answers": [
        {
          "text": "Всего на 0,3 секунды.",
          "label": 1
        },
        {
          "text": "На 0,3 секунды.",
          "label": 1
        },
        {
          "text": "На секунду.",
          "label": 0
        },
        {
          "text": "На 0.5 секунд.",
          "label": 0
        }
      ]
    },
    {
      "idx": 0
    }
  ]
}
```



```
X_train, y_train = get_X_y('train.jsonl')
X_test, y_test = get_X_y('val.jsonl')
```

Посмотрим на количество примеров и состав классов:

```
len(X_train)
```

```
11950
```

```
len(X_test)
```

```
2235
```

```
from collections import Counter
print(Counter(y_train))
print(Counter(y_test))
```

```
Counter({0: 6568, 1: 5382})
Counter({0: 1242, 1: 993})
```

Запуск baseline

Перебор моделей и параметров — классический подход.

Попробуем модели разного типа, выберем лучший классификатор из простых, а потом будем перебирать его настройки.

```
rs = 42
```

```
clf = LogisticRegression(random_state=rs)
clf2 = RandomForestClassifier(random_state=rs, n_jobs=-1)
clf3 = SGDClassifier()
clf4 = SVC(random_state=rs)
clf5 = DecisionTreeClassifier(random_state=rs)
"""clf6 = SVC(class_weight="balanced", random_state=rs)
clf7 = DecisionTreeClassifier()
clf8 = ExtraTreeClassifier()
clf9 = LinearRegression()
clf10 = LogisticRegressionCV()
clf11 = GradientBoostingClassifier(random_state=rs)"""
clflist = [clf, clf2, clf3, clf4, clf5]
```

Запустим и оценим решение

```
for classif in clflist:
    clf = Pipeline([
        ('vect', CountVectorizer(ngram_range=(1,3), analyzer='word', max_features=10000)),
        ('tfidf', TfidfTransformer(sublinear_tf=True)),
        #('reducer', TruncatedSVD(n_components=Val3)),
        ('clf', classif),
    ])
    clf.fit(X_train, y_train)
    predictions = clf.predict(X_test)
    print(classif)
```

```

print("Precision: {0:6.2f}".format(precision_score(y_test, predictions,
→ average='macro')))
print("Recall: {0:6.2f}".format(recall_score(y_test, predictions,
→ average='macro')))
print("F1-measure: {0:6.2f}".format(f1_score(y_test, predictions,
→ average='macro')))
print("Accuracy: {0:6.2f}".format(accuracy_score(y_test, predictions)))
print(classification_report(y_test, predictions))
labels = clf.classes_
sns.heatmap(data=confusion_matrix(y_test, predictions), annot=True, fmt="d",
→ cbar=False, xticklabels=labels, yticklabels=labels)
plt.title("Confusion matrix")
plt.show()

```

```

LogisticRegression(C=1.0, class_weight=None, dual=False, fit_intercept=True,
intercept_scaling=1, l1_ratio=None, max_iter=100,
multi_class='auto', n_jobs=None, penalty='l2',
random_state=42, solver='lbfgs', tol=0.0001, verbose=0,
warm_start=False)

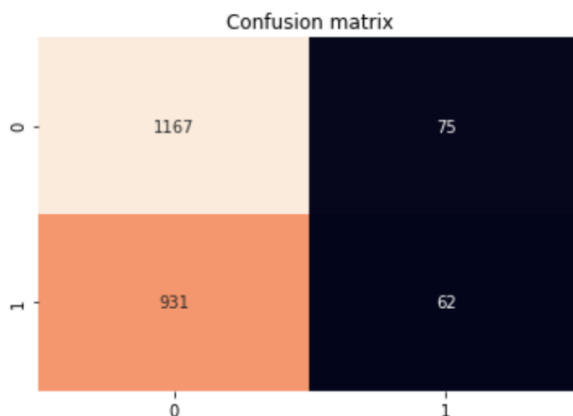
```

```

Precision: 0.50
Recall: 0.50
F1-measure: 0.40
Accuracy: 0.55

```

	precision	recall	f1-score	support
0	0.56	0.94	0.70	1242
1	0.45	0.06	0.11	993
accuracy			0.55	2235
macro avg	0.50	0.50	0.40	2235
weighted avg	0.51	0.55	0.44	2235



```

RandomForestClassifier(bootstrap=True, ccp_alpha=0.0, class_weight=None,
criterion='gini', max_depth=None, max_features='auto',
max_leaf_nodes=None, max_samples=None,
min_impurity_decrease=0.0, min_impurity_split=None,
min_samples_leaf=1, min_samples_split=2,
min_weight_fraction_leaf=0.0, n_estimators=100,
n_jobs=-1, oob_score=False, random_state=42, verbose=0,
warm_start=False)

```

```

Precision: 0.52
Recall: 0.51

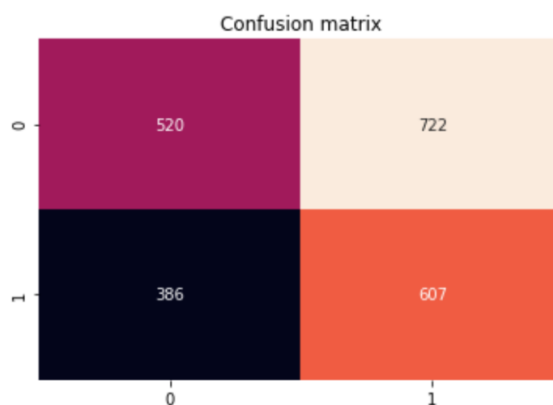
```

```

F1-measure: 0.50
Accuracy: 0.50
      precision  recall  f1-score  support
0         0.57    0.42    0.48    1242
1         0.46    0.61    0.52    993

accuracy          0.50    2235
macro avg         0.52    0.51    0.50    2235
weighted avg      0.52    0.50    0.50    2235

```



```

SGDClassifier(alpha=0.0001, average=False, class_weight=None,
              early_stopping=False, epsilon=0.1, eta0=0.0, fit_intercept=True,
              l1_ratio=0.15, learning_rate='optimal', loss='hinge',
              max_iter=1000, n_iter_no_change=5, n_jobs=None, penalty='l2',
              power_t=0.5, random_state=None, shuffle=True, tol=0.001,
              validation_fraction=0.1, verbose=0, warm_start=False)

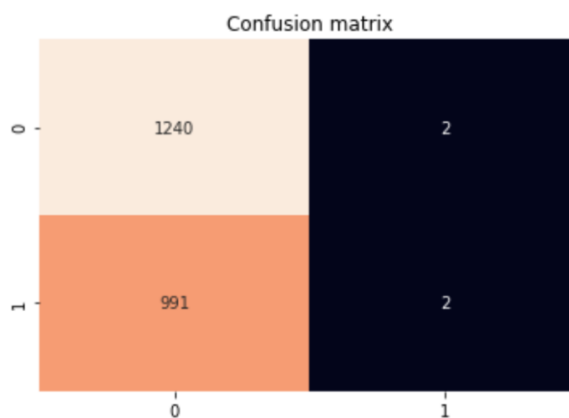
```

```

Precision: 0.53
Recall: 0.50
F1-measure: 0.36
Accuracy: 0.56
      precision  recall  f1-score  support
0         0.56    1.00    0.71    1242
1         0.50    0.00    0.00    993

accuracy          0.56    2235
macro avg         0.53    0.50    0.36    2235
weighted avg      0.53    0.56    0.40    2235

```



```
SVC(C=1.0, break_ties=False, cache_size=200, class_weight=None, coef0=0.0,
    decision_function_shape='ovr', degree=3, gamma='scale', kernel='rbf',
    max_iter=-1, probability=False, random_state=42, shrinking=True, tol=0.001,
    verbose=False)
```

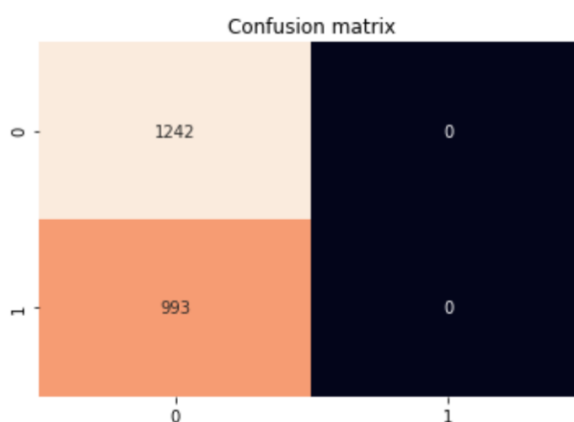
Precision: 0.28

Recall: 0.50

F1-measure: 0.36

Accuracy: 0.56

	precision	recall	f1-score	support
0	0.56	1.00	0.71	1242
1	0.00	0.00	0.00	993
accuracy			0.56	2235
macro avg	0.28	0.50	0.36	2235
weighted avg	0.31	0.56	0.40	2235



```
DecisionTreeClassifier(ccp_alpha=0.0, class_weight=None, criterion='gini',
    max_depth=None, max_features=None, max_leaf_nodes=None,
    min_impurity_decrease=0.0, min_impurity_split=None,
    min_samples_leaf=1, min_samples_split=2,
    min_weight_fraction_leaf=0.0, presort='deprecated',
    random_state=42, splitter='best')
```

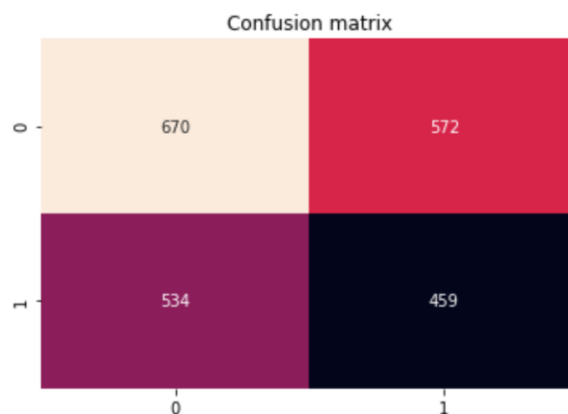
Precision: 0.50

Recall: 0.50

F1-measure: 0.50

Accuracy: 0.51

	precision	recall	f1-score	support
0	0.56	0.54	0.55	1242
1	0.45	0.46	0.45	993
accuracy			0.51	2235
macro avg	0.50	0.50	0.50	2235
weighted avg	0.51	0.51	0.51	2235



Попробуем BERT

Перезапустите среду выполнения! И тут лучше переключиться на GPU!

Установите все библиотеки ниже, и затем заново загрузите данные, импортируйте sklearn — так вы избежите конфликта версий.

```
!pip install transformers
!pip install tensorboardx
!pip install simpletransformers
```

```
train_df = pd.DataFrame({
    'text': X_train,
    'label': y_train
})
```

```
print(train_df.head())
```

```
eval_df = pd.DataFrame({
    'text': X_test,
    'label': y_test
})
```

```
print(eval_df.head())
```

	text	label
0	(1) Но люди не могут существовать без природы,...	1
1	(1) Но люди не могут существовать без природы,...	0
2	(1) Но люди не могут существовать без природы,...	0
3	(1) Но люди не могут существовать без природы,...	1
4	(1) Но люди не могут существовать без природы,...	0

	text	label
0	(1) Самый первый «остров» Архипелага возник в ...	1
1	(1) Самый первый «остров» Архипелага возник в ...	0
2	(1) Самый первый «остров» Архипелага возник в ...	0
3	(1) Самый первый «остров» Архипелага возник в ...	0
4	(1) Самый первый «остров» Архипелага возник в ...	1

```
from simpletransformers.classification import ClassificationModel
```

```
model = ClassificationModel('bert', 'bert-base-multilingual-uncased', use_cuda=False)
```

```

# Train the model
model.train_model(train_df)

# Evaluate the model
result, model_outputs, wrong_predictions = model.eval_model(eval_df)

HBox(children=(HTML(value='Downloading'), FloatProgress(value=0.0, max=625.0),
↳ HTML(value='')))
HBox(children=(HTML(value='Downloading'), FloatProgress(value=0.0, max=672271273.0),
↳ HTML(value='')))

HBox(children=(HTML(value='Downloading'), FloatProgress(value=0.0, max=871891.0),
↳ HTML(value='')))

HBox(children=(HTML(value=''), FloatProgress(value=0.0, max=11950.0), HTML(value='')))
HBox(children=(HTML(value='Epoch'), FloatProgress(value=0.0, max=1.0),
↳ HTML(value='')))
HBox(children=(HTML(value='Running Epoch 0 of 1'), FloatProgress(value=0.0,
↳ max=1494.0), HTML(value='')))

predictions, raw_outputs = model.predict(X_test)

print("Precision: {0:6.2f}".format(precision_score(y_test, predictions,
↳ average='macro')))
print("Recall: {0:6.2f}".format(recall_score(y_test, predictions, average='macro')))
print("F1-measure: {0:6.2f}".format(f1_score(y_test, predictions, average='macro')))
print("Accuracy: {0:6.2f}".format(accuracy_score(y_test, predictions)))
print(classification_report(y_test, predictions))
confusion_matrix(y_test, predictions)

```

Что еще можно улучшить?

Ограничение модели BERT — обработка только первых 512 токенов текста.

Нам это не очень подходит — если обрезать тексты по 512 слову, как это происходит в simple transformers, но мы можем обрезать и вопрос, и ответ! Тогда наш классификатор не сможет научиться различать неправильные ответы.

Обрежем тексты по началу, а не по концу:

```

def text_splitter(text, amount=500):
    tokens = text.split(' ')
    new_text = ' '.join(tokens[:-amount])
    return new_text

def get_X_y_for_bert(data_json_file):
    X, y = [], []
    with open(data_json_file, 'r') as json_file:
        json_list = list(json_file)
        #print(json_list[0])
        for json_str in json_list:
            item = json.loads(json_str)
            text = item['passage']['text']
            questions = item['passage']['questions']
            for q in questions:
                query = q['question']
                ans = q['answers']
                for a in ans:

```



```
X.append(text_splitter(text+' Query: '+query+' Answer:
↔ '+a['text']))
y.append(a['label'])
return X, y
```

Еще идеи:

- больше эпох обучения;
- попробовать другие модели для классификации документов из simple transformers — только мультязычные! См. документацию (<https://simpletransformers.ai/docs/classification-models/>).

Заключительный этап

Индивидуальный предметный тур

Информатика. 8–11 класс

Задача III.1.1.1. A plus/minus B (10 баллов)

Вы являетесь членом команды, разрабатывающей интеллектуальный калькулятор. Идея этого калькулятора следующая: пользователь вслух произносит арифметическое выражение, результат которого он хочет узнать, а калькулятор автоматически распознает его речь, вычисляет выражение и формирует голосовое сообщение с результатом вычисления. Часть команды разработчиков уже перевела голосовое сообщение в текстовый вид. Вы ответственный за внутренние вычисления в калькуляторе. На вход вашему блоку подается текстовая запись выражения, на выходе должен быть текстовый ответ для этого выражения. Так как это только прототип, то все вычисления происходят на целых числах в интервале от 0 до 99, а выражение имеет вид либо $A \text{ plus } B$ либо $A \text{ minus } B$.

Формат входных данных

В первой строке содержится количество выражений n , на которые нужно вывести ответ ($1 \leq n \leq 100$).

В следующих n строках содержатся текстовые записи арифметических выражений. Каждая строка имеет вид либо $A \text{ plus } B$ либо $A \text{ minus } B$, где A и B — текстовая запись двух чисел от 0 до 99 на английском языке. Гарантируется, что результат вычисления каждого такого выражения также число из этого отрезка. Все слова каждого выражения разделены ровно одним пробелом. В записи чисел используются следующие слова:

zero, one, two three, four, five, six, seven, eight, nine,

ten, eleven, twelve, thirteen, fourteen, fifteen, sixteen, seventeen, eighteen, nineteen,

twenty, thirty, forty, fifty, sixty, seventy, eighty, ninety.

Формат выходных данных

Вывести n строк, в каждой ответ на соответствующее выражение в виде текстового сообщения.

*Примеры**Пример №1*

Стандартный ввод
4 twelve plus forty nine seventy two minus seventeen one minus one twenty three plus sixty eight
Стандартный вывод
sixty one fifty five zero ninety one

Пример программы-решения

Ниже представлено решение на языке C++.

```

1  #include <bits/stdc++.h>
2
3  using namespace std;
4  typedef long long ll;
5  typedef pair<int, int> pii;
6  typedef long double ld;
7
8  int main(){
9      ios::sync_with_stdio(0), cin.tie(0), cout.tie(0);
10
11     string S19[20] = {"zero", "one", "two", "three", "four", "five", "six", "seven",
12     ↪ "eight", "nine",
13     ↪ "ten", "eleven", "twelve", "thirteen", "fourteen", "fifteen",
14     ↪ "sixteen", "seventeen", "eighteen", "nineteen"};
15     string S90[9] = {"ten", "twenty", "thirty", "forty", "fifty", "sixty", "seventy",
16     ↪ "eighty", "ninety"};
17
18     map<string, int> T;
19
20     for(int i = 0; i < 20; i++)
21         T[S19[i]] = i;
22
23     for(int i = 0; i < 9; i++)
24         T[S90[i]] = (i+1) * 10;
25
26     vector<string> itotxt(100);
27     for(auto q : T)
28         itotxt[q.second] = q.first;
29
30     int u;
31     cin >> u;
32     string vs;
33     getline(cin, vs);
34     while(u--){
35         string s;
36         getline(cin, s);

```

```

34
35     stringstream ss;
36     ss << s;
37
38     string ds, op;
39     bool frst = 1;
40     int a = 0, b = 0, r;
41     while(ss >> ds){
42         if(ds == "plus" || ds == "minus"){
43             frst = 0;
44             op = ds;
45         }
46         else
47             if(frst)
48                 a += T[ds];
49             else
50                 b += T[ds];
51     }
52
53     if(op == "plus")
54         r = a + b;
55     else
56         r = a - b;
57
58     if(r >= 20) {
59         cout<<itotxt[r - r%10];
60         if(r%10 > 0)
61             cout<<' '<<itotxt[r%10];
62         cout<<endl;
63     }
64     else cout<<itotxt[r]<<endl;
65 }
66 }

```

Задача III.1.1.2. Развлечение ИИ (15 баллов)

Один ИИ в целях самосовершенствования генерирует случайные последовательности натуральных чисел и затем для каждой находит самую большую перестановку, которая является непрерывным подотрезком сгенерированной последовательности.

Последовательность A является подотрезком последовательности B , если A может быть получена из B удалением нескольких (возможно, ни одного) элементов из начала и нескольких (возможно, ни одного) элементов из конца.

Перестановкой длины n называется последовательность, содержащая числа от 1 до n в произвольном порядке. Каждое число должно входить в нее ровно один раз.

Требуется проверить результат ИИ. По исходной последовательности нужно найти размер самой большой перестановки, являющейся непрерывным подотрезком заданной последовательности натуральных чисел.

Формат входных данных

В первой строке записано число n — количество чисел в исходной последовательности, $1 \leq n \leq 1000$. В следующей строке находятся n чисел через пробел — элементы последовательности. Все они в пределах от 1 до n .

Формат выходных данных

Вывести одно число — ответ на задачу. Если ни одной перестановки нет, вывести 0.

Пример №1

Стандартный ввод
19 5 2 6 3 5 1 4 2 1 3 6 2 5 8 1 2 5 4 3
Стандартный вывод
6

Пояснения к примеру

В предложенной последовательности встречаются перестановки 4 2 1 3 и 1 2 5 4 3, но самой большой является перестановка 6 3 5 1 4 2 или 2 6 3 5 1 4 длины 6.

Пример программы-решения

Ниже представлено решение на языке C++.

```

1  #include <bits/stdc++.h>
2
3  using namespace std;
4  typedef long long ll;
5  typedef pair<int, int> pii;
6  typedef long double ld;
7
8  int main(){
9      ios::sync_with_stdio(0), cin.tie(0), cout.tie(0);
10
11     int n, ans = 0;
12     cin >> n;
13     vector<int> v(n);
14     for(int i = 0; i < n; i++) cin >> v[i];
15     vector<int> mask(n+1, 0);
16
17     for(int i = 0; i < n; i++){
18         int t = 0, mx = 0;
19         for(int j = i; j < n; j++){
20             if(mask[v[j]] != 0) break;
21             else{
22                 t++;
23                 mask[v[j]] = 1;
24                 mx = max(mx, v[j]);
25                 if(mx == t) ans = max(ans, mx);
26             }
27         }
28         for(int j = i; j < n; j++)
29             if(mask[v[j]] != 0) mask[v[j]] = 0; else break;
30     }
31     cout << ans;
32 }
```

Пример программы-решения

Ниже представлено решение на языке Python.

```

1  n = int(input())
2  a = input().split()
3  ans = 0
4  for i in range(n):
5      a[i] = int(a[i])
6  for len in range(1, n + 1):
7      upd = []
8      for i in range(n + 1):
9          upd.append(0)
10     l = 0
11     r = len
12     bal = 0
13     upper = 0
14     for i in range(len):
15         if upd[a[i]] == 0:
16             bal += 1
17             upd[a[i]] += 1
18             if a[i] > len:
19                 upper += 1
20     if bal == len and upper == 0:
21         ans = len
22     for i in range(len, n):
23         if upd[a[i]] == 0:
24             bal += 1
25             if a[i] > len:
26                 upper += 1
27             upd[a[i]] += 1
28             if upd[a[i - len]] == 1:
29                 bal -= 1
30             if a[i - len] > len:
31                 upper -= 1
32
33         upd[a[i - len]] -= 1
34         if bal == len and upper == 0:
35             ans = len
36     print(ans)

```

Задача III.1.1.3. Диалог двух ИИ (20 баллов)

Два независимых ИИ ведут диалог. Его запись представляет собой непрерывную строку, в которую без пробелов записаны слова диалога. Известен набор слов, используемых ими при диалоге. В какой-то момент в диалог вмешивается третий ИИ (возможно в самый последний момент, когда запись диалога уже закончена), и диалог первых двух нарушается. Требуется по записи беседы и набору слов, доступному первым двум ИИ определить какое максимальное количество слов могли сказать первые два ИИ до того момента, пока не вмешался третий.

Формат входных данных

В первой строке записана непустая строка, содержащая полную запись беседы трех ИИ, длина этой строки не превосходит 1000. Во второй строке находится число

n — количество слов, используемых первыми двумя ИИ ($1 \leq n \leq 100$). В следующих n строках содержатся слова, при помощи которых первые два ИИ ведут диалог. И первая строка и слова записаны только символами «0» и «1». Длина каждого слова не превосходит 50, слова могут повторяться.

Формат выходных данных

Вывести одно число — максимальное количество слов, которое могли сказать друг другу первые два ИИ до того момента, пока не вмешался третий.

Примеры

Пример №1

Стандартный ввод
110110110111111011010010
6
110
11011
01111
1101101
11
0110
Стандартный вывод
7

Пояснения к примеру

Начало строки из примера можно многими способами представить в виде последовательности слов (в скобках указано окончание, сказанное третьим участником беседы):

$110 + 11011 + 01111 + 1101101 + (0010)$ — здесь было сказано 4 слова;

$11 + 0110 + 11011 + 11 + 11011 + (010010)$ — здесь было сказано 5 слов;

$110 + 110 + 110 + 11 + 11 + 1101101 + (0010)$ — здесь было сказано 6 слов;

$110 + 110 + 110 + 11 + 11 + 110 + 110 + (10010)$ — здесь было сказано 7 слов, и это максимальное возможное количество.

Пример программы-решения

Ниже представлено решение на языке C++.

```

1  #include <bits/stdc++.h>
2
3  using namespace std;
4  typedef long long ll;
5  typedef pair<int, int> pii;
6  typedef long double ld;
7
```

```

8  vector<int> prefix_function (string s) {
9      int n = (int) s.length();
10     vector<int> pi (n);
11     for (int i=1; i<n; ++i) {
12         int j = pi[i-1];
13         while (j > 0 && s[i] != s[j])
14             j = pi[j-1];
15         if (s[i] == s[j]) ++j;
16         pi[i] = j;
17     }
18     return pi;
19 }
20
21 int main(){
22     ios::sync_with_stdio(0), cin.tie(0), cout.tie(0);
23
24     string s;
25     cin >> s;
26     int n, len, ans = 0;
27
28     s = "#" + s;
29     len = s.size();
30     cin >> n;
31     vector<string> v(n);
32     for(int i = 0; i < n; i++) cin >> v[i];
33
34     vector<vector<int> > K(len+1);
35     for(int i = 0; i < n; i++){
36         string z = v[i] + "@" + s;
37         vector<int> tkmp = prefix_function(z);
38
39         for(int j = 0; j < z.size(); j++)
40             if(tkmp[j] == v[i].size())
41                 K[j - v[i].size() + 1 - 1 - v[i].size()].push_back(i);
42     }
43
44     vector<int> dp(len, -1);
45     dp[0] = 0;
46
47     for(int i = 0; i < len - 1; i++)
48         if(dp[i] >= 0)
49             for(auto q : K[i+1]){
50                 int tlen = v[q].size();
51                 dp[i + tlen] = max(dp[i + tlen], dp[i] + 1);
52             }
53
54     for(auto q : dp) ans = max(ans, q);
55     cout << ans;
56 }

```

Пример программы-решения

Ниже представлено решение на языке Python.

```

1  s = input()
2  n = int(input())
3  a = []
4  dp = []
5  for i in range(n):

```



```
6     a.append(input())
7 for i in range(len(s) + 1):
8     dp.append(-1000000000)
9 dp[0] = 0
10 for i in range(1, len(s) + 1):
11     for j in range(n):
12         if (i - len(a[j]) >= 0):
13             ff = s[i - len(a[j]):i]
14             if ff == a[j]:
15                 dp[i] = max(dp[i], dp[i - len(a[j])] + 1)
16 ans = 0
17 for i in range(1, len(s) + 1):
18     ans = max(dp[i], ans)
19 print(ans)
```

Задача III.1.1.4. Круги и окружности (25 баллов)

Классической задачей теории ИИ является задача на распознавание изображений. В данном случае вам предлагается выяснить количество кругов и количество окружностей «нарисованных» на картинке. «Рисовать» будем на прямоугольной таблице, состоящей из единичных квадратов. Если единичный квадрат принадлежит какой-то фигуре, то в нем стоит символ «<#>» (решетка), иначе символ «<.>» (точка).

Формат входных данных

В первой строке заданы размеры изображения n и m . $30 \leq n, m \leq 1000$. В следующих n строках, каждая длиной m символов, задается само изображение. Диаметры кругов и окружностей могут быть различными, но не меньше 6. Ни одна из фигур не касается границ изображения и других фигур. Линия любой окружности «непрерывна», то есть из любой ее точки можно попасть в любую другую ее точку перемещаясь по «точкам» окружности (смежными о стороне или углу) как по часовой, так и против часовой стрелки. Рекомендуем внимательно вспомнить определение окружности.

Формат выходных данных

Вывести два числа через пробел – сначала количество кругов, затем количество окружностей, изображенных на рисунке.


```

11 vector<string> T;
12     vector<vector<int> > G;
13
14 bool in_tr(int x, int y){
15     return(x >= 0 && y >= 0 && x < n && y < m);
16 }
17
18 void no_dfs1(int x, int y, int c){
19     vector<pii> och;
20     och.push_back({x, y});
21     int b = 0;
22     G[x][y] = c;
23     while(b < och.size()){
24         int tx = och[b].first;
25         int ty = och[b].second;
26         for(int i = 0; i < 4; i++){
27             int nx = och[b].first + dx1[i];
28             int ny = och[b].second + dy1[i];
29             if(in_tr(nx, ny) && T[x][y] == T[nx][ny] && G[nx][ny] == -1){
30                 G[nx][ny] = c;
31                 och.push_back({nx, ny});
32             }
33         }
34         b++;
35     }
36 }
37
38 void no_dfs2(int x, int y, int c){
39     vector<pii> och;
40     och.push_back({x, y});
41     int b = 0;
42     G[x][y] = c;
43     while(b < och.size()){
44         int tx = och[b].first;
45         int ty = och[b].second;
46         for(int i = 0; i < 8; i++){
47             int nx = och[b].first + dx2[i];
48             int ny = och[b].second + dy2[i];
49             if(in_tr(nx, ny) && T[x][y] == T[nx][ny] && G[nx][ny] == -1){
50                 G[nx][ny] = c;
51                 och.push_back({nx, ny});
52             }
53         }
54         b++;
55     }
56 }
57
58 int main(){
59     ios::sync_with_stdio(0), cin.tie(0), cout.tie(0);
60
61     cin >> n >> m;
62     T.resize(n);
63     for(int i = 0; i < n; i++)
64         cin >> T[i];
65
66     G.resize(n, vector<int>(m, -1));
67
68     int k = 0, h1;
69     for(int i = 0; i < n; i++)
70     for(int j = 0; j < m; j++)

```

```

71     if(G[i][j] == -1){
72         k++;
73         if(T[i][j] == '#'){
74             h1++;
75             no_dfs2(i, j, k);
76         }
77         else
78             no_dfs1(i, j, k);
79     }
80     cout<<2 * h1 - k + 1<<' '<<k - 1 - h1<<endl;
81 }

```

Пример программы-решения

Ниже представлено решение на языке Python.

```

1  asd = input().split()
2  dx = [-1, -1, 0, 1, 1, 1, 0, -1]
3  dy = [0, 1, 1, 1, 0, -1, -1, -1]
4  n = int(asd[0])
5  m = int(asd[1])
6  a = []
7  ans1 = 0
8  ans2 = 0
9  visited = []
10 for i in range(n + 2):
11     ss = []
12     ad = []
13     for j in range(m + 2):
14         ss.append('.')
15         ad.append(False)
16     a.append(ss)
17     visited.append(ad)
18 for i in range(1, n + 1):
19     f = input()
20     for j in range(1, m + 1):
21         a[i][j] = f[j - 1]
22 for i in range(1, n + 1):
23     for j in range(1, m + 1):
24         if a[i][j] == '#' and not visited[i][j]:
25             ans = []
26             visited[i][j] = True
27             Q = []
28             Q.append([i, j])
29             while len(Q) > 0:
30                 y = Q[len(Q) - 1][0]
31                 x = Q[len(Q) - 1][1]
32                 #print(y, x)
33                 ans.append([y, x])
34                 Q.pop()
35             for k in range(8):
36                 if not visited[y + dy[k]][x + dx[k]] and a[y + dy[k]][x + dx[k]] == '#':
37                     Q.append([y + dy[k], x + dx[k]])
38                 visited[y + dy[k]][x + dx[k]] = True
39     ans.sort()
40     prev_y = -1
41     ch = True
42     for k in range(1, len(ans)):
43         if (ans[k][0] != prev_y):

```

```

44         prev_y = ans[k][0]
45     elif (ans[k][1] - ans[k - 1][1] != 1):
46         ch = False
47     if ch:
48         ans1 += 1
49     else:
50         ans2 += 1
51 print(ans1, ans2)

```

Задача III.1.1.5. Унификация (30 баллов)

В этой задаче в самой простой форме научимся строить логические выводы. Если более точно, то только предварительную обработку некоторых выражений, называемую унификацией. Метод, для которого строится унификация, называется методом резолюций. Это один из самых давних методов теории ИИ, позволяющий автоматически доказывать правильность логических рассуждений. Договоримся о следующих обозначениях (для краткости изложения, некоторые из них не совсем формализованы):

- буквы a, b, c (и только они) обозначают константы, вместо них ничего подставлять нельзя. Константы соответствуют конкретным объектам рассматриваемой области, о которой мы хотим что-то утверждать (конкретным числам, людям, словам и т. п.).
- буквы x, y, z обозначают переменные, вместо любой переменной можно подставить терм, но нельзя подставить предикат. Вместо переменной нельзя подставлять терм, содержащий эту переменную. Если вместо одной переменной подставлен терм t , то в точности этот же терм должен быть подставлен и вместо всех других переменных с таким же именем в предикате, содержащем эту переменную. Определения термов и предикатов смотрите ниже. Ограничений на количество различных переменных вводить не будем, но гарантируется, что во входных тестах в качестве переменных использованы только эти символы.
- буквы f, g, h (и только они, будем называть их функциональными символами) обозначают термы (функции). Каждый терм имеет местность (количество аргументов), в этой задаче будем считать, что любой терм зависит от нуля, одного или двух аргументов. Изначально аргументы термов — это переменные. Примеры исходных термов (без подстановок) следующие: одноместный терм $f(x)$ (например, «отец объекта x »), двуместный терм $g(x, z)$ (например «число, являющееся суммой чисел x и z »). Терм, после подстановки вместо всех его переменных конкретных констант возвращает какой-то объект из множества рассматриваемых объектов. Помимо функциональных символов, термами считаются константы (нульместные термы) и переменные (одноместные термы).
- буквы P, Q, R (и только они) обозначают предикаты (отношения). Предикаты так же имеют местность. В данной задаче будем ограничиваться рассмотрением только двуместных предикатов. После подстановки вместо всех своих переменных некоторых констант и вычисления значений термов, предикат возвращает логическое значение $TRUE$ или $FALSE$, которое интерпретируется как результат логического утверждения относительно аргументов предиката. Пример исходного предиката (без подстановок): $Q(y, z)$ (например «число y делится нацело на число z »).
- подстановкой вместо переменной (например x) терма (например $f(y)$) в предикат P будем называть результат замены всех вхождений этой переменной

в этом предикате на заданный терм. Напомним, чтобы избежать ненужной рекурсии, подставлять вместо переменной терм, содержащий ту же самую переменную запрещено. Примеры подстановок (в терм и предикат):

- в терме $f(x, z)$ подставим вместо z константу a , получим $f(x, a)$;
- в терме $f(x, g(x, h(z)))$ подставим вместо x терм $g(a, f(b, y))$, получим $f(g(a, f(b, y)), g(g(a, f(b, y)), h(z)))$. Мы обязаны заменить все вхождения x на $g(a, f(b, y))$;
- в предикате $P(f(x), g(x, y))$ подставим вместо x терм $h(y)$, получим $P(f(h(y)), g(h(y), y))$.

Отметим отдельно, что можно по-очереди производить несколько подстановок. В том числе на следующих шагах можно подставлять вместо переменных, появившихся в результате предыдущих подстановок.

- унификацией двух предикатов P_1 и P_2 называется некоторый последовательный набор подстановок отдельно внутрь P_1 и отдельно внутрь P_2 , таких, что P_1 и P_2 посимвольно совпадут. Другими словами, два предиката унифицируются, если найдутся две серии независимых друг от друга подстановок таких, что предикаты совпадут с точностью до символа. Унификацией так же называется и результат применения этих подстановок, то есть предикат, который получается, когда P_1 и P_2 совпадают. Далее будем рассматривать только унифицируемые пары предикатов. Очевидно, что два предиката унифицируемы, если они обозначены одним и тем же предикатным символом.

Пример унификации предикатов:

$$P_1 = P(x, g(y, f(x)))$$

$$P_2 = P(f(y), g(b, f(f(a))))$$

Если мы сразу заменим в P_1 переменную x на терм $f(y)$, то получим $P_1 = P(f(y), g(y, f(f(y))))$ и далее потерпим неудачу, так как переменная y должна быть заменена в одном случае на константу b , а в другом на константу a . Поэтому, действуем немного по-другому. Сначала в P_2 заменим переменную y на другую переменную, например, z . Получим $P_2 = P(f(z), g(b, f(f(a))))$. Так как предикаты меняются независимо, то в предикате P_1 ничего не изменится. Теперь выполним в P_1 замену x на $f(z)$, получим $P_1 = P(f(z), g(y, f(f(z))))$. Осталось заменить в P_1 переменную z на константу a , и потом переменную y на константу b . А в P_2 нужно заменить переменную z на константу a . Получим, что оба предиката примут вид $P(f(a), g(b, f(f(a))))$. Это и будет результат их унификации. Возможен и другой порядок выполнения подстановок, но результат унификации будет тот же.

Ваша задача — написать программу, которая для таких пар предикатов строит их унификацию.

Формат входных данных

В первой строке задан первый предикат P_1 , во второй строке задан второй предикат P_2 . Правила записи и ограничения описаны выше, запись предикатов не содержит пробелов. Если два терма обозначены одинаковыми функциональными символами, то они имеют одинаковую местность. Гарантируется, что заданные предикаты унифицируемы и результат унификации однозначен и не содержит переменных. Каждый предикат записан не более чем 50 символами. Для понимания деталей формата ввода смотрите пример.

Формат выходных данных

Вывести результат унификации двух исходных предикатов в том же формате.

Примеры

Пример №1

Стандартный ввод
P(x,g(y,f(x))) P(f(y),g(b,f(f(a))))
Стандартный вывод
P(f(a),g(b,f(f(a))))

Пример программы-решения

Ниже представлено решение на языке C++.

```

1  #include <bits/stdc++.h>
2
3  using namespace std;
4  typedef long long ll;
5  typedef pair<int, int> pii;
6  typedef long double ld;
7
8  string P1, P2;
9  int t1 = 99, t2 = 499;
10 vector<int> mask(1000, -1);
11
12 string tos(int a){
13     stringstream ss;
14     string r;
15     ss << a;
16     ss >> r;
17     return r;
18 }
19
20 int toa(string s){
21     stringstream ss;
22     int r;
23     ss << s;
24     ss >> r;
25     return r;
26 }
27
28 string termf(string P, int pos){
29     if(P[pos] <= 'h' && P[pos] >= 'f'){
30         int bal = 1, t = pos + 2;
31         while(bal > 0){
32             if(P[t] == '(') bal++;
33             if(P[t] == ')') bal--;
34             t++;
35         }
36         return P.substr(pos, t - pos);
37     }

```

```

38     else {
39         return P.substr(pos, 3);
40     }
41 }
42
43 string subst(string P, int posb, int pose, string term){
44     string r = P.substr(0, posb);
45     r += term;
46     r += P.substr(pose + 1);
47
48     return r;
49 }
50
51 bool isvar(string s){
52     bool ok = 1;
53     for(int i = 0; i < s.size(); i++)
54         if(s[i] > '9' || s[i] < '0') ok = 0;
55     return ok;
56 }
57
58 void standart(){
59     t1++;
60     for(int i = 0; i < P1.size(); i++)
61         if(P1[i] == 'x')
62             P1 = subst(P1, i, i, tos(t1));
63
64     t1++;
65     for(int i = 0; i < P1.size(); i++)
66         if(P1[i] == 'y')
67             P1 = subst(P1, i, i, tos(t1));
68
69     t1++;
70     for(int i = 0; i < P1.size(); i++)
71         if(P1[i] == 'z')
72             P1 = subst(P1, i, i, tos(t1));
73
74     for(int i = P1.size()-1; i >= 0; i--){
75         if(P1[i] <= 'c' && P1[i] >= 'a'){
76             string vs;
77             for(int j = 0; j < 3; j++) vs += P1[i];
78             P1 = subst(P1, i, i, vs);
79         }
80     }
81
82     t2++;
83     for(int i = 0; i < P2.size(); i++)
84         if(P2[i] == 'x')
85             P2 = subst(P2, i, i, tos(t2));
86
87     t2++;
88     for(int i = 0; i < P2.size(); i++)
89         if(P2[i] == 'y')
90             P2 = subst(P2, i, i, tos(t2));
91
92     t2++;
93     for(int i = 0; i < P2.size(); i++)
94         if(P2[i] == 'z')
95             P2 = subst(P2, i, i, tos(t2));
96
97     for(int i = P2.size()-1; i >= 0; i--){

```



```

98     if(P2[i] <= 'c' && P2[i] >= 'a'){
99         string vs;
100        for(int j = 0; j < 3; j++) vs += P2[i];
101        P2 = subst(P2, i, i, vs);
102    }
103 }
104 }
105
106 int main(){
107     ios::sync_with_stdio(0), cin.tie(0), cout.tie(0);
108     cin >> P1 >> P2;
109     standart();
110
111     while(1){
112         int t = 0;
113         while(t + 3 < P1.size() && t + 3 < P2.size()){
114             if(P1[t] == P2[t])
115                 t++;
116             else
117                 if(isvar(P1.substr(t, 3)) && isvar(P2.substr(t, 3)))
118                     t += 3;
119             else
120                 break;
121         }
122
123         if(t + 3 >= P1.size() || t + 3 >= P2.size()){
124             for(int i = 0; i < P1.size(); i++)
125                 if( i < P1.size() && P1[i] <= 'c' && P1[i] >= 'a'){
126                     string vs;
127                     vs += P1[i];
128                     P1 = subst(P1, i, i + 2, vs);
129                 }
130
131             cout<<P1<<endl;
132             return 0;
133         }
134         else{
135             string TP, IP;
136             int f;
137             if(isvar(P2.substr(t, 3)) ) {
138                 TP = P2;
139                 IP = P1;
140                 f = 2;
141             }
142             else{
143                 TP = P1;
144                 IP = P2;
145                 f = 1;
146             }
147
148             string tx = TP.substr(t, 3);
149             string tt = termf(IP, t);
150
151             if(f == 1){
152                 for(int i = 0; i < tt.size(); i++)
153                     if(i + 3 < tt.size() && isvar(tt.substr(i, 3))){
154                         int ty = toa(tt.substr(i, 3));
155                         if(mask[ty] == -1){
156                             t1++;
157                             mask[ty] = t1;

```

```

158     }
159 }
160
161     for(int i = 0; i < tt.size(); i++)
162         if(i + 3 < tt.size() && isvar(tt.substr(i, 3))){
163             int ty = toa(tt.substr(i, 3));
164             tt = subst(tt, i, i+2, tos(mask[ty]));
165         }
166     }
167     else{
168         for(int i = 0; i < tt.size(); i++)
169             if(i + 3 < tt.size() && isvar(tt.substr(i, 3))){
170                 int ty = toa(tt.substr(i, 3));
171                 if(mask[ty] == -1){
172                     t2++;
173                     mask[ty] = t2;
174                 }
175             }
176
177         for(int i = 0; i < tt.size(); i++)
178             if(i + 3 < tt.size() && isvar(tt.substr(i, 3))){
179                 int ty = toa(tt.substr(i, 3));
180                 tt = subst(tt, i, i+2, tos(mask[ty]));
181             }
182     }
183
184     for(int i = 0; i < TP.size(); i++)
185         if(TP.substr(i, 3) == tx)
186             TP = subst(TP, i, i + 2, tt);
187
188     if(f == 2){
189
190         P2 = TP;
191         P1 = IP;
192     }
193     else{
194         P1 = TP;
195         P2 = IP;
196     }
197 }
198 }
199 }

```

Математика. 8-9 класс

Задача III.1.2.1. (15 баллов)

На новогодние праздники родители возили Оксану на Регульские острова. Они вылетели из Москвы в 12:00 по московскому времени и приземлились на островах на следующий день в 7:30 по местному времени. При возвращении они стартовали в 9:30 по местному времени и приземлились в Москве в 18:00 по московскому времени в тот же день. Считая, что продолжительность полета была одинаковой в обоих случаях, найдите, сколько времени было на Регульских островах, когда Оксана приземлилась в Москве?

Решение

Пусть разница во временных поясах между Москвой и Регульскими островами равна x часов, а время перелета — y часов. Тогда получаем систему:

$$\begin{cases} x + y = 19,5 \\ y - x = 8,5 \end{cases}$$

Откуда, $x = 5,5$, $y = 14$. Следовательно, в момент прилета в Москву на островах было 23:30.

Система оценки

- Полное обоснованное решение — 15 баллов.
- Составлена система — 10 баллов.
- Задача не решена или решена неверно — 0 баллов.

Ответ: 23:30.

Задача III.1.2.2. (15 баллов)

Портье перепутал ключи от 100 закрытых номеров гостиницы. Он не знает, какой ключ к какому номеру подходит. Какое наименьшее количество попыток открывания нужно сделать, чтобы определить ключи ко всем номерам?

Решение

Подойдем к первому номеру и будем пробовать открыть его, последовательно подбирая ключи. Если за 99 попыток открыть не удалось, то оставшийся ключ будет от этого номера. Для второго номера будет достаточно 98 попыток, для третьего — 97 попыток и т. д. Всего попыток:

$$99 + 98 + 97 + \dots + 2 + 1 = (\text{по формуле суммы арифметической прогрессии}) = \frac{99 + 1}{2} \cdot 99 = 4950.$$

Система оценки

- Полное обоснованное решение — 15 баллов.
- Количество отличается от верного на 1 для каждой двери — 5 баллов.
- Задача не решена или решена неверно — 0 баллов.

Ответ: 4950.

Задача III.1.2.3. (20 баллов)

На доске записаны 576 натуральных чисел. Среди любых 122 чисел найдется хотя бы одно четное число, а среди любых 456 чисел найдется хотя бы одно нечетное число. Может ли сумма всех записанных чисел равняться $2020 \cdot 2021$?

Решение

Среди всех чисел может быть не более 121 нечетных чисел и не более 455 четных чисел. Так как чисел всего 576, то четных чисел ровно 455 и нечетных чисел ровно 121. Сумма всех чисел — нечетное число, а $2020 \cdot 2021$ — четное число.

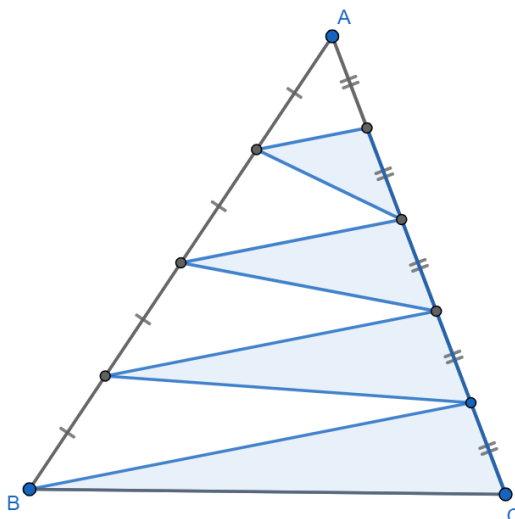
Система оценки

- Полное обоснованное решение — 20 баллов.
- Определено количество четных и нечетных чисел — 10 баллов.
- Задача не решена или решена неверно — 0 баллов.

Ответ: нет.

Задача III.1.2.4. (20 баллов)

Точка случайным образом (равномерно) брошена в треугольник ABC . Найдите вероятность попадания точки в один из закрашенных треугольников.

*Решение*

Отметим точки D, E, F, G, H, I, J и опустим перпендикуляры BN, FM, EL, DK на прямую AC . Искомая вероятность равна отношению суммы площадей закрашенных треугольников DGH, EHI, FIJ и BJC к площади треугольника ABC .

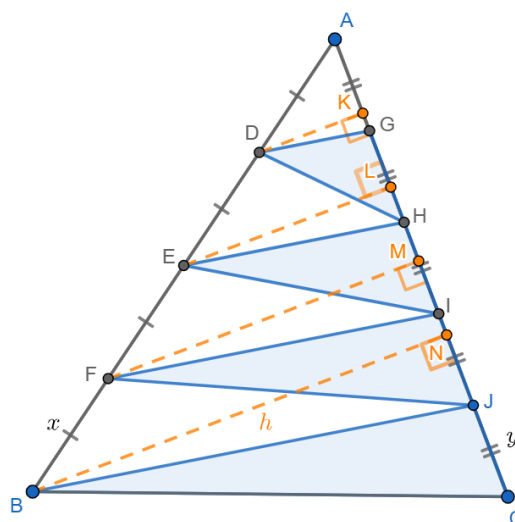
Введем обозначения:

$$AD = DE = EF = FB = x,$$

$$AG = GH = HI = IJ = JC = y,$$

$$BN = h.$$

Прямоугольные треугольники ADK, AEL, AFM, ABM подобны по общему острому углу, отсюда получаем $DK = \frac{1}{4}h, EL = \frac{1}{2}h, FM = \frac{3}{4}h$.



Тогда $S_{ABC} = \frac{1}{2}BN \cdot AC = \frac{5}{2}hy$, а сумма площадей закрашенных треугольников равна:

$$\begin{aligned} S_{DGH} + S_{EHI} + S_{FIJ} + S_{BJC} &= \frac{1}{2}DK \cdot GH + \frac{1}{2}EL \cdot HI + \frac{1}{2}FM \cdot IJ + \frac{1}{2}BM \cdot JC = \\ &= \frac{1}{2}y \cdot \left(\frac{1}{4}h + \frac{1}{2}h + \frac{3}{4}h + h\right) = \frac{5}{4}hy \end{aligned}$$

Искомая вероятность равна: $\left(\frac{5}{4}hy\right) : \left(\frac{5}{2}hy\right) = \frac{1}{2}$.

Система оценки

- Полное обоснованное решение — 20 баллов.
- Допущена одна вычислительная ошибка, при этом ход решения верен — 15 баллов.
- Сформулирована идея о том, что искомая вероятность равна отношению площади закрашенных треугольников к общей площади $\triangle ABC$ — 5 баллов.
- Задача не решена или решена неверно — 0 баллов.

Ответ: $\frac{1}{2}$.

Задача III.1.2.5. (30 баллов)

Найдите все действительные корни уравнения:

$$(x^2 + 7x + 12)(x^2 - 3x + 2)(x^2 + 2x + 4) = 108.$$

Решение

Разложим первый и третий квадратные трехчлены в левой части уравнения на множители, а затем перегруппируем множители:

$$(x + 3)(x + 4)(x - 1)(x - 2)(x^2 + 2x + 4) = (x^2 + 2x - 3)(x^2 + 2x - 8)(x^2 + 2x + 4).$$

После замены $t = x^2 + 2x$ получим уравнение: $(t - 3)(t - 8)(t + 4) - 108 = 0$;

После раскрытия скобок и приведения подобных слагаемых имеем:

$$t^3 - 7t^2 - 20t - 12 = 0.$$

Знакопеременная сумма коэффициентов многочлена в левой части равна нулю ($1 + 7 - 20 + 12 = 0$), поэтому один из корней уравнения $t = -1$. Разложим левую часть на множители:

$$(t + 1)(t^2 - 8t - 12) = 0.$$

Корнями данного уравнения являются числа -1 , $4 + 2\sqrt{7}$ и $4 - 2\sqrt{7}$.

Выполним обратную замену.

- а) при $t = -1$ получаем $x^2 + 2x = -1$, откуда $x_1 = -1$.
- б) при $t = 4 + 2\sqrt{7}$ получаем $x^2 + 2x - (4 + 2\sqrt{7}) = 0$; это уравнение имеет два действительных корня $x_{2,3} = -1 \pm \sqrt{(5 + 2\sqrt{7})}$.
- в) при $t = 4 - 2\sqrt{7}$ получаем уравнение $x^2 + 2x - 4 + 2\sqrt{7} = 0$, упрощенный дискриминант $D_1 = 5 - 2\sqrt{7}$ которого отрицателен, поэтому действительных корней оно не имеет.

Система оценки

- Полное обоснованное решение — 30 баллов.
- Верное в целом решение, но допущена одна вычислительная ошибка — 25 баллов.
- После замены получено кубическое уравнение — 15 баллов.
- Найдены корни этого уравнения — 5 баллов (в дополнение к 15).
- Задача не решена или решена неверно — 0 баллов.

Ответ: -1 ; $-1 + \sqrt{(5 + 2\sqrt{7})}$; $-1 - \sqrt{(5 + 2\sqrt{7})}$.

Математика. 10-11 класс

Задача III.1.3.1. (15 баллов)

Планету Пандора осваивают две компании: разведывательная и строительная. Для них привезли с Земли роботов-разведчиков и роботов-строителей. Для увеличения количества роботов своей специальности обе компании тайно отправляют на склад хакеров, которые перепрограммируют роботов. В результате этого десятая часть роботов-разведчиков считают себя строителями, а десятая часть роботов-строителей считают себя разведчиками. Если же рассматривать всех роботов, то пятая часть роботов считают себя разведчиками. Какую часть составляли роботы-разведчики от общего числа привезенных с Земли роботов?

Решение

Пусть x — количество привезенных с Земли роботов-строителей, y — количество привезенных роботов-разведчиков. После работы хакеров $0,9y + 0,1x = 0,2(x + y)$. Отсюда $x = 7y$. Поэтому $\frac{y}{x+y} = \frac{1}{8}$.

Система оценки

- Полное обоснованное решение — 15 баллов.
- Составлена система — 10 баллов.
- Задача не решена или решена неверно — 0 баллов.

Ответ: $\frac{1}{8}$.

Задача III.1.3.2. (15 баллов)

В рамках тестирования генератора случайных чисел программист Вася решает уравнения вида $2Ax - 3A + 4 = 0$. Считая, что величина A равномерно распределена на отрезке $[-1; 4]$, найдите вероятность того, что число x окажется больше $1/2$.

Решение

При $A = 0$ уравнение не имеет решений, а при $A \neq 0$ получаем $x = \frac{3A-4}{2A}$. Решим неравенство $\frac{3A-4}{2A} > \frac{1}{2}$:

$$\frac{3A-4}{A} - 1 > 0 \Leftrightarrow \frac{2(A-2)}{A} > 0 \Leftrightarrow A \in (-\infty; 0) \cup (2; +\infty).$$

На отрезке $[-1; 4]$ решением будет объединение промежутков $[-1; 0)$ и $(2; 4]$.

Искомая вероятность равна отношению суммы длин отрезков $[-1; 0)$ и $(2; 4]$ к длине отрезка $[-1; 4]$: $P = \frac{1+2}{5} = \frac{3}{5}$.

Система оценки

Баллы суммируются:

- Верно составлено неравенство ИЛИ верно найдено значение A , соответствующее значению $x = \frac{1}{2}$ — 5 баллов.
- Решено неравенство — 5 баллов.
- Верно вычислена вероятность — 5 баллов.

Задача не решена или решена неверно — 0 баллов.

Ответ: $\frac{3}{5}$.

Задача III.1.3.3. (20 баллов)

На гранях игрального кубика записаны числа 1, 2, 2, 3, 6, 6. В каждом туре два игрока по очереди бросают кубик. Первый игрок выигрывает, если при бросании на грани кубика выпадает 6 очков, второй — если 3 очка. Игра заканчивается в случае выигрыша одного из игроков. Найдите вероятность выигрыша второго игрока (не обязательно в первом туре).

Решение

Событие $A =$ «выигрыш второго игрока» является объединением независимых событий $A_n =$ «второй игрок выиграет именно в n -м туре», где n принимает все значения от 1 до бесконечности. Событие A_1 является пересечением событий «первый игрок не получает 6» и «второй игрок получает 3». $P(A_1) = \frac{4}{6} \cdot \frac{1}{6} = \frac{1}{9}$. Событие A_2 происходит, если в первом туре игра не закончилась, а во втором туре победил второй игрок. $P(A_2) = (\frac{4}{6} \cdot \frac{5}{6}) \cdot (\frac{4}{6} \cdot \frac{1}{6}) = \frac{5}{9} \cdot \frac{1}{9}$. Аналогично, $P(A_3) = \frac{5}{9} \cdot \frac{5}{9} \cdot \frac{1}{9}$ и $P(A_{n+1}) = P(A_n) \cdot \frac{5}{9}$. То есть вероятности событий $P(A_n)$ представляют собой бесконечно убывающую геометрическую прогрессию с первым членом $\frac{1}{9}$ и знаменателем $\frac{5}{9}$. $P(A) = \sum_{n=1}^{\infty} P(A_n) = \frac{\frac{1}{9}}{1 - \frac{5}{9}} = 0,25$.

Система оценки

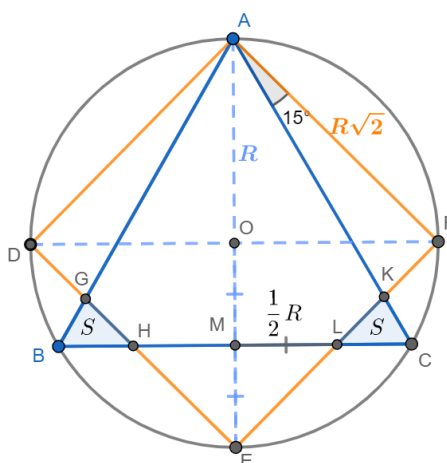
- Полное обоснованное решение — 20 баллов.
- Найдена вероятность выигрыша в 1 туре — 5 баллов.
- Определены вероятности последующих событий — 10 баллов. (В дополнение к 5 баллам.)
- Задача не решена или решена неверно — 0 баллов.

Ответ: 0,25.

Задача III.1.3.4. (20 баллов)

Вписанные в круг квадрат и правильный треугольник имеют общую вершину. В круг случайным образом (равномерно) бросают точку. Найдите вероятность, что она попадет в треугольник, но не попадет в квадрат.

Решение



Пусть в окружность с центром O и радиусом R вписаны правильный треугольник ABC и квадрат $ADEF$, тогда стороны треугольника и квадрата равны $R\sqrt{3}$ и $R\sqrt{2}$ соответственно. Обозначим точки пересечения треугольника и квадрата как G, H, K и L (см. рисунок). Тогда искомая вероятность P равна отношению суммы площадей закрашенных треугольников BGH и CKL к площади круга. В силу симметрии относительно прямой AC площади закрашенных треугольников равны; обозначим площадь каждого из них как S .

1. Найдем CK . Из прямоугольного треугольника AFK :

$$AK = AF : \cos 15^\circ = R\sqrt{2} : \frac{\sqrt{6} + \sqrt{2}}{4} = \frac{4R}{\sqrt{3} + 1} = 2(\sqrt{3} - 1)R,$$

где значение $\cos 15^\circ = \cos(45^\circ - 30^\circ) = \cos 45^\circ \cos 30^\circ + \sin 45^\circ \sin 30^\circ = \frac{\sqrt{6} + \sqrt{2}}{4}$.

Тогда $CK = AC - AK = R(\sqrt{3} - 2\sqrt{3} + 2) = (2 - \sqrt{3})R$.

2. Найдем CL . Т.к. AM — медиана $\triangle ABC$, то $AO : OM = 2 : 1$ и $OM = \frac{1}{2}R$. Следовательно, $EM = OE - OM = \frac{1}{2}R$. Треугольник EML — равнобедренный прямоугольный, поэтому $ML = ME = \frac{1}{2}R$. Отсюда $CL = CM - ML = \frac{\sqrt{3}-1}{2}R$.
3. $S = S_{CKL} = \frac{1}{2}CL \cdot CK \cdot \sin \angle LCK = \frac{\sqrt{3}}{4}CL \cdot CK = \frac{\sqrt{3}}{4} \cdot \frac{\sqrt{3}-1}{2}R \cdot (2-\sqrt{3})R = \frac{9-5\sqrt{3}}{8}R^2$.
4. Окончательно получаем: $P = \frac{2S}{\pi R^2} = \frac{9-5\sqrt{3}}{4\pi}$.

Система оценки

- Полное обоснованное решение — 20 баллов.
- Допущена одна вычислительная ошибка, при этом ход решения верен — 15 баллов.
- Сформулирована идея о том, что искомая вероятность равна отношению площади закрашенных треугольников к площади круга — 5 баллов.
- Задача не решена или решена неверно — 0 баллов.

Ответ: $\frac{9-5\sqrt{3}}{4\pi}$.

Задача III.1.3.5. (30 баллов)

Найдите наименьшее натуральное m , при котором существуют натуральные x и y , удовлетворяющие уравнению:

$$(x^2 + 4y^2)^2 + 4my(4y^2 + x^2) = m^2x^2$$

.

Решение

Сделаем замену $2y = t$ — четное натуральное число. Рассмотрим это уравнение как квадратное относительно m , и найдем его корни: $m = \frac{(x^2+t^2)(t+\sqrt{x^2+t^2})}{x^2}$ (второй корень отрицательный, так как $\sqrt{x^2+t^2} > t$).

Отсюда:

$$mx^2 = (x^2 + t^2)(t + \sqrt{x^2 + t^2}).$$

Пусть $d = \text{НОД}(x; t)$ и $x = x_0d$, $t = t_0d$. Подставляя в равенство, получим:

$$mx_0^2 = d(x_0^2 + t_0^2)(t_0 + \sqrt{x_0^2 + t_0^2}).$$

x_0 и x_0^2 взаимно просто с t_0 , t_0^2 и с $x_0^2 + t_0^2$. Следовательно, m должно быть кратно $x_0^2 + t_0^2$ и $x_0^2 + t_0^2$ должно быть полным квадратом (иначе $\sqrt{x_0^2 + t_0^2}$ будет иррациональным числом и равенство не будет верным). Наименьшая сумма квадратов, являющаяся квадратом, это $3^2 + 4^2 = 5^2$. Значит, $m \geq 25$. При $m = 25$, $x = 3$ и $y = 2$ получаем верное равенство.

Система оценки

- Полное обоснованное решение — 30 баллов.
- Верное в целом решение, но допущена одна вычислительная ошибка — 25 баллов.
- Уравнение рассмотрено как квадратное относительно m и найдены его корни — 5 баллов.
- Задача не решена или решена неверно — 0 баллов.

Ответ: 25.

Командный практический тур

Задача командного тура

В командной части заключительного этапа участникам было предложено решить актуальную для научного сообщества задачу по обработке естественного языка. Длительность командного этапа — 3 дня (14,5 астрономических часов).

Участникам был предоставлен датасет, состоящий из новостных вырезок, в которых описано некоторое событие и содержалась именованная сущность. Каждый текст сопровождался предложением с пропуском, где нужно добавить эту именованную сущность из списка вариантов. Данные варианты были извлечены из текста и могли повторяться. Правильный выбор, извлеченный из любого места текста, считался правильным, т. е. правильных ответов могло быть несколько. Главное условие: чтобы краткое содержание было осмысленным.

Актуальность задачи

Решение задач на обработку естественного языка, аналогичные этой, где алгоритмы искусственного интеллекта должны учитывать не только доступную информацию, но и здравый смысл, сейчас особенно интересуют российское и международное научное сообщество, так как без их успешного решения невозможно создание развитых интеллектуальных систем и сильного искусственного интеллекта.

Подобные алгоритмы также активно применяются в бизнесе при создании автоматизированных колл-центров и сервисов клиентской поддержки, в чат-ботах, голосовых помощниках и поисковых машинах.

Описание данных

Для решения задачи предоставлен датасет из 70000+ текстов, которые представляют собой вырезки из новостей. Все текстовые примеры были собраны из открытых источников, а затем автоматически отфильтрованы с помощью QA-систем, чтобы не допустить проникновения очевидных вопросов в набор данных. Затем тексты были отфильтрованы по частоте IPM содержащихся слов и, наконец, просмотрены вручную.

В архиве доступно 3 файла:

- train и val — для обучения модели и настройки параметров;
- test — проверочный файл, в котором нужно сделать предсказание и загрузить на платформу для проверки.

Скачать данные: <https://onti2020.ai-academy.ru/task/rucos.zip>

Пример данных:

```
{
  'source': 'Lenta',
  'passage': {
    'text':
```

```

Институт немецкой экономики задался вопросом: действительно ли жилье
→ и офисы в ФРГ скупают инвесторы из других стран. Эксперты
→ получили недвусмысленный ответ и рассказали о нем DW. @header
→ Учитывались только трансграничные сделки\nСледует учесть, что в
→ исследовании IW использовались данные только о трансграничных
→ инвестиционных потоках, когда сделки совершались на деньги,
→ поступившие, например, непосредственно из России. \n"Если
→ российский инвестор сначала основал в Германии фирму, а уже
→ потом с ее счета оплатил покупку коммерческой недвижимости, то
→ это в нашу статистику не попало", - предупредил эксперт. К тому
→ же обращает на себя внимание резкая активизация инвесторов с
→ Кипра. И вот в 2011 году с маленького острова, на котором широко
→ представлен российский капитал, вдруг поступили сразу 7
→ миллионов евро - столько же, сколько из Испании.\n@highlight\nB
→ Германии за сутки выявлено более 100 новых заражений
→ коронавирусом\n@highlight\nТысячи демонстрантов в Гамбурге
→ выступили за прием беженцев\n@highlight\nКомментарий: Россия
→ накануне эпидемии - виноватые назначены заранее',
'entities': [
  {'start': 79, 'end': 82, 'text': 'ФРГ'},
  {'start': 177, 'end': 179, 'text': 'DW'},
  {'start': 416, 'end': 422, 'text': 'России'},
  {'start': 468, 'end': 476, 'text': 'Германии'},
  {'start': 678, 'end': 683, 'text': 'Кипра'},
  {'start': 839, 'end': 846, 'text': 'Испании'},
  {'start': 861, 'end': 869, 'text': 'Германии'},
  {'start': 962, 'end': 970, 'text': 'Гамбурге'},
  {'start': 1023, 'end': 1029, 'text': 'Россия'}
],
'qas': [
  {
    'query': ' В беседе с @placeholder профессор Фогтлендер особо указал
→ на то, что в крупных городах Германии, прежде всего, в Берлине,
→ доля иностранцев в сделках на рынке недвижимости существенно
→ выше, чем в среднем по стране: \n"Там она составляет уже не 1 и
→ не 5, а, по-видимому, 10-15 процентов\n".',
    'answers': [
      {'start': 177, 'end': 179, 'text': 'DW'}
    ],
    'idx': 0
  }
],
'idx': 0
}

```

Метрика и формат решений

На платформу для проверки нужно загрузить jsonl-файл, соответствующий примеру, в котором указан id вопроса и ответ на него.

Качество решения оценивалось автоматически по метрике F1.

Базовое решение

Решение задачи RuCoS

Информация о задаче: https://russiansuperglue.com/tasks/task_info/RuCoS.

Загрузка данных.

```
!wget https://russiansuperglue.com/tasks/download/RuCoS -O RuCoS.zip
!unzip RuCoS.zip

--2021-04-28 17:27:23-- https://russiansuperglue.com/tasks/download/RuCoS
Resolving russiansuperglue.com (russiansuperglue.com)... 37.18.107.48
Connecting to russiansuperglue.com (russiansuperglue.com)|37.18.107.48|:443...
↪ connected.
HTTP request sent, awaiting response... 200 OK
Length: 56208297 (54M) [application/zip]
Saving to: 'RuCoS.zip'

RuCoS.zip          100%[=====>]  53.60M  6.25MB/s   in 8.8s

2021-04-28 17:27:32 (6.11 MB/s) - 'RuCoS.zip' saved [56208297/56208297]

Archive:  RuCoS.zip
  creating: RuCoS/
 inflating: RuCoS/train.jsonl
   creating: __MACOSX/
   creating: __MACOSX/RuCoS/
 inflating: __MACOSX/RuCoS/._train.jsonl
 inflating: RuCoS/.DS_Store
 inflating: __MACOSX/RuCoS/._DS_Store
 inflating: RuCoS/test.jsonl
 inflating: __MACOSX/RuCoS/._test.jsonl
 inflating: RuCoS/val.jsonl
 inflating: __MACOSX/RuCoS/._val.jsonl
 inflating: __MACOSX/._RuCoS
```

Установка зависимостей.

```
!pip install -q transformers seaborn
!pip install nltk
!pip install tqdm
!pip install pandas
!pip install numpy
```

```
Requirement already satisfied: nltk in /usr/local/lib/python3.7/dist-packages
↪ (3.2.5)
Requirement already satisfied: six in /usr/local/lib/python3.7/dist-packages
↪ (from nltk) (1.15.0)
Requirement already satisfied: tqdm in /usr/local/lib/python3.7/dist-packages
↪ (4.41.1)
```

```

Requirement already satisfied: pandas in /usr/local/lib/python3.7/dist-packages
  ↳ (1.1.5)
Requirement already satisfied: python-dateutil>=2.7.3 in
  ↳ /usr/local/lib/python3.7/dist-packages (from pandas) (2.8.1)
Requirement already satisfied: pytz>=2017.2 in
  ↳ /usr/local/lib/python3.7/dist-packages (from pandas) (2018.9)
Requirement already satisfied: numpy>=1.15.4 in
  ↳ /usr/local/lib/python3.7/dist-packages (from pandas) (1.19.5)
Requirement already satisfied: six>=1.5 in
  ↳ /usr/local/lib/python3.7/dist-packages (from python-dateutil>=2.7.3->pandas)
  ↳ (1.15.0)
Requirement already satisfied: numpy in /usr/local/lib/python3.7/dist-packages
  ↳ (1.19.5)

```

Подготовка данных

Подключаем библиотеки.

```

import pandas as pd
import numpy as np
import json
from tqdm.notebook import tqdm
import re
import warnings
import seaborn as sns

from nltk import sent_tokenize
import nltk

```

Инициализация и основные константы.

```

nltk.download('punkt')
tqdm.pandas()

warnings.simplefilter("ignore")

SEED = 1337
TRAIN_BATCH_SIZE = 64
VAL_BATCH_SIZE = 64
SEQ_LEN = 220
MODEL_NAME = 'DeepPavlov/rubert-base-cased'
device = 'cuda:0'

```

```

[nltk_data] Downloading package punkt to /root/nltk_data...
[nltk_data] Package punkt is already up-to-date!

```

Считываем данные.

```

with open('RuCoS/train.jsonl', 'r', encoding='utf-8') as f:
    train_content = f.read()

```

```
with open('RuCoS/test.jsonl', 'r', encoding='utf-8') as f:
    test_content = f.read()
```

```
with open('RuCoS/val.jsonl', 'r', encoding='utf-8') as f:
    val_content = f.read()
```

```
train_dict = [json.loads(jline) for jline in train_content.splitlines()]
test_dict = [json.loads(jline) for jline in test_content.splitlines()]
val_dict = [json.loads(jline) for jline in val_content.splitlines()]
```

Для ускорения обучения загрузим лишь часть данных. Для итогового решения надо использовать весь набор.

```
MAX_TRAIN = 30000
MAX_VAL = 2000
```

```
train_dict = train_dict[:MAX_TRAIN]
val_dict = val_dict[:MAX_VAL]
```

Обработаем тренировочные данные чтобы далее с ними было удобнее работать.

```
train_prettified = {
    'text_id': [],
    'q_id': [],
    'a_id': [],
    'text': [],
    'query': [],
    'end': [],
    'start': [],
    'qtext': [],
    'label': []
}

for i, el in enumerate(tqdm(train_dict)):
    passage = el['passage']

    text_id = el['idx']
    q_id = el['qas'][0]['idx']
    text = passage['text']
    query = el['qas'][0]['query']

    a_id = 0
    for ent in passage['entities']:

        label = 0
        for ans in el['qas'][0]['answers']:
            if ans['end'] == ent['end'] and ans['start'] == ent['start']:
                label = 1
                break

    qtext = text[ent['start']:ent['end']]
    train_prettified['text_id'].append(text_id)
```

```

train_prettified['q_id'].append(q_id)
train_prettified['a_id'].append(a_id)
train_prettified['text'].append(text)
train_prettified['query'].append(query)
train_prettified['end'].append(ent['end'])
train_prettified['start'].append(ent['start'])
train_prettified['qtext'].append(qtext)
train_prettified['label'].append(label)

a_id += 1

print('Размер обучающей выборки:', len(train_prettified['text_id']), '| Верных',
      ↪ sum(train_prettified['label'])/len(train_prettified['text_id']))

HBox(children=(FloatProgress(value=0.0, max=30000.0), HTML(value='')))

```

Размер обучающей выборки: 400890 | Верных 0.17734041757090474

Аналогичная обработка для validation выборки и test.

```

val_prettified = {
    'text_id': [],
    'q_id': [],
    'a_id': [],
    'text': [],
    'query': [],
    'end': [],
    'start': [],
    'qtext': [],
    'label': []
}

for i, el in enumerate(tqdm(val_dict)):
    passage = el['passage']

    text_id = el['idx']
    q_id = el['qas'][0]['idx']
    text = passage['text']
    query = el['qas'][0]['query']

    a_id = 0
    for ent in passage['entities']:

        label = 0
        for ans in el['qas'][0]['answers']:
            if ans['end'] == ent['end'] and ans['start'] == ent['start']:
                label = 1
                break
        qtext = text[ent['start']:ent['end']]
        val_prettified['text_id'].append(text_id)
        val_prettified['q_id'].append(q_id)
        val_prettified['a_id'].append(a_id)
        val_prettified['text'].append(text)

```



```

    val_prettified['query'].append(query)
    val_prettified['end'].append(ent['end'])
    val_prettified['start'].append(ent['start'])
    val_prettified['qtext'].append(qtext)
    val_prettified['label'].append(label)

    a_id += 1

print('Размер валидационной выборки', len(val_prettified['text_id']), '|
↳ Верных', sum(val_prettified['label'])/len(val_prettified['text_id']))

HBox(children=(FloatProgress(value=0.0, max=2000.0), HTML(value='')))

Размер валидационной выборки 26139 | Верных 0.15008225257278396

test_prettified = {
    'text_id': [],
    'q_id': [],
    'a_id': [],
    'text': [],
    'query': [],
    'end': [],
    'start': [],
    'qtext': []
}

for i, el in enumerate(tqdm(test_dict)):
    passage = el['passage']

    text_id = el['idx']
    q_id = el['qas'][0]['idx']
    text = passage['text']
    query = el['qas'][0]['query']

    a_id = 0
    for ent in passage['entities']:
        qtext = text[ent['start']:ent['end']]
        test_prettified['text_id'].append(text_id)
        test_prettified['q_id'].append(q_id)
        test_prettified['a_id'].append(a_id)
        test_prettified['text'].append(text)
        test_prettified['query'].append(query)
        test_prettified['end'].append(ent['end'])
        test_prettified['start'].append(ent['start'])
        test_prettified['qtext'].append(qtext)

        a_id += 1
print('Размер тестовой выборки:', len(test_prettified['text_id']))

HBox(children=(FloatProgress(value=0.0, max=7257.0), HTML(value='')))

Размер тестовой выборки: 96996

```

Теперь, обработаем конструкции вида @name в датасете.

Создадим все возможные запросы с @placeholder замененным на возможный ответ.

```
def prep(x):
    x = re.split(r'\n@highlight\n', x)[0].replace('\n', ' ').replace('@header',
    ↪ ' ')
    return x

    %%time
train = pd.DataFrame(train_prettified)
test = pd.DataFrame(test_prettified)
val = pd.DataFrame(val_prettified)

train = train[train['query'].apply(lambda x: x.count('@placeholder') ==
    ↪ 1)].reset_index(drop=True)

train['text'] = train['text'].progress_apply(prepare)
val['text'] = val['text'].progress_apply(prepare)
test['text'] = test['text'].progress_apply(prepare)

# необходимая замена
train['query_prep'] = train[['query', 'qtext']].apply(lambda x:
    ↪ x['query'].replace('@placeholder', x['qtext']), axis=1)
test['query_prep'] = test[['query', 'qtext']].apply(lambda x:
    ↪ x['query'].replace('@placeholder', x['qtext']), axis=1)
val['query_prep'] = val[['query', 'qtext']].apply(lambda x:
    ↪ x['query'].replace('@placeholder', x['qtext']), axis=1)

# удалим дубликаты
train = train.drop_duplicates(['text', 'query_prep'])
test = test.drop_duplicates(['text', 'query_prep'])
val = val.drop_duplicates(['text', 'query_prep'])

HBox(children=(FloatProgress(value=0.0, max=393952.0), HTML(value='')))

HBox(children=(FloatProgress(value=0.0, max=26139.0), HTML(value='')))

HBox(children=(FloatProgress(value=0.0, max=96996.0), HTML(value='')))

CPU times: user 12.5 s, sys: 562 ms, total: 13 s
Wall time: 13.1 s
```

Посмотрим, что получилось.

```
train[['text', 'query', 'query_prep']].head()
```

	text	query	query_prep
0	Наблюдатели полагают, что подоплекой теракта в...	Кроме того, серьезным вызовом для России стано...	Кроме того, серьезным вызовом для России стано...
2	Наблюдатели полагают, что подоплекой теракта в...	Кроме того, серьезным вызовом для России стано...	Кроме того, серьезным вызовом для России стано...
4	Наблюдатели полагают, что подоплекой теракта в...	Кроме того, серьезным вызовом для России стано...	Кроме того, серьезным вызовом для России стано...
6	Наблюдатели полагают, что подоплекой теракта в...	Кроме того, серьезным вызовом для России стано...	Кроме того, серьезным вызовом для России стано...
7	Наблюдатели полагают, что подоплекой теракта в...	Кроме того, серьезным вызовом для России стано...	Кроме того, серьезным вызовом для России стано...

```
train[['text', 'query', 'query_prep']].iloc[0]['query'], train[['text', 'query',
↪ 'query_prep']].iloc[0]['query_prep'], train[['text', 'query',
↪ 'query_prep']].iloc[1]['query_prep']
```

```
('Кроме того, серьезным вызовом для России становится стремительно развивающийся
↪ Китай.Еще в понедельник @placeholder в рамках спора о системе
↪ противоракетной обороны пригрозил размещением дополнительных ракетных
↪ комплексов.',
'Кроме того, серьезным вызовом для России становится стремительно развивающийся
↪ Китай.Еще в понедельник Домодедово в рамках спора о системе противоракетной
↪ обороны пригрозил размещением дополнительных ракетных комплексов.',
'Кроме того, серьезным вызовом для России становится стремительно развивающийся
↪ Китай.Еще в понедельник Süddeutsche Zeitung в рамках спора о системе
↪ противоракетной обороны пригрозил размещением дополнительных ракетных
↪ комплексов.')
```

Подготовка модели SBERT

Инициализация и основные константы.

```
from transformers import AutoTokenizer, AutoModel,
↪ get_linear_schedule_with_warmup

import torch
from torch import nn, optim
from torch.utils.data import Dataset, DataLoader
import torch.nn.functional as F
import random
import os
from sklearn.metrics import f1_score
from collections import defaultdict

from torch.cuda.amp import autocast, GradScaler
```

Установим состояние генераторов случайных чисел для воспроизводимости результата.

```
def set_seed(seed = 42, set_torch=True):
    random.seed(seed)
    np.random.seed(seed)
    os.environ["PYTHONHASHSEED"] = str(seed)
    if set_torch:
        torch.manual_seed(seed)
        torch.cuda.manual_seed(seed)
        torch.backends.cudnn.deterministic = True
        torch.backends.cudnn.benchmark = False
set_seed(SEED)
```

Загрузим токенайзер и саму модель. Будем использовать уже предобученные модели.

```

tokenizer = AutoTokenizer.from_pretrained(MODEL_NAME)
bert_model = AutoModel.from_pretrained(MODEL_NAME)

# Класс для работы с датасетом в формате, принимаемым моделью
class CQDataset(Dataset):
    def __init__(self, texts, queries, targets, tokenizer, max_len=512):
        self.texts = texts
        self.queries = queries
        self.targets = targets
        self.tokenizer = tokenizer
        self.max_len = max_len

    def __len__(self):
        return len(self.texts)

    def __getitem__(self, item):
        text = str(self.texts[item])
        query = str(self.queries[item])
        target = self.targets[item]

        tt = self.tokenizer.tokenize(text)
        qt = self.tokenizer.tokenize(query)

        # Мы используем DeepPavlov RuBERT поэтому ввод должен иметь вид
        # [CLS] context_tokens [SEP] query_tokens [SEP]
        tokens = tt + ['[SEP]'] + qt
        tokens = tokens[-self.max_len+2:]
        tokens =
        ↪ ['[CLS]']+tokens+['[SEP]']+['[PAD]']*(self.max_len-2-len(tokens))

        attention_mask = [1*(token!='[PAD]') for token in tokens]
        fe = tokens.index('[SEP]')
        token_type_ids = [(i>=fe)*1 for i in range(len(tokens))]
        input_ids = self.tokenizer.convert_tokens_to_ids(tokens)

        input_ids = torch.tensor(input_ids, dtype=torch.long)
        attention_mask = torch.tensor(attention_mask, dtype=torch.long)
        token_type_ids = torch.tensor(token_type_ids, dtype=torch.long)
        target = torch.tensor(target, dtype=torch.float16)

        return {
            'input_ids': input_ids,
            'attention_mask': attention_mask,
            'token_type_ids': token_type_ids,
            'target': target
        }

def create_data_loader(df, tokenizer, max_len, batch_size, shuffle=True):
    ds = CQDataset(
        df['text'].values,
        df['query_prep'].values,
        df['label'].values,
        tokenizer,

```

```

        max_len
    )

    return DataLoader(
        ds,
        batch_size=batch_size,
        num_workers=2,
        shuffle=shuffle
    )

```

Подготовим загрузчики данных на основании датасетов.

```

train_dl = create_data_loader(train, tokenizer, SEQ_LEN, TRAIN_BATCH_SIZE)
val_dl = create_data_loader(val, tokenizer, SEQ_LEN, VAL_BATCH_SIZE,
    ↪ shuffle=False)

```

Будем использовать свой классификатор, принимающий основную модель.

```

class CQClassifier(nn.Module):
    def __init__(self, base_model, units=768):
        super(CQClassifier, self).__init__()
        self.bert = base_model
        self.mlp_head = nn.Sequential(
            nn.Dropout(0.2),
            nn.Linear(units, 1)
        )
    def forward(self, input_ids, attention_mask, token_type_ids):
        # INPUT
        bert_out = self.bert(
            input_ids=input_ids,
            token_type_ids=token_type_ids,
            attention_mask=attention_mask
        )
        x = bert_out['pooler_output']
        x = self.mlp_head(x).view(-1)
        return x

```

```

model = CQClassifier(bert_model, units=768)
model = model.to(device)

```

```

print('Итоговое число параметров:', sum(p.numel() for p in model.parameters()))
print('Обучаемые параметры:', sum(p.numel() for p in model.parameters() if
    ↪ p.requires_grad))

```

```

Итоговое число параметров: 177854209
Обучаемые параметры: 177854209

```

Проверим состояние видеокарты.

```
! nvidia-smi
```

Wed Apr 28 17:28:18 2021

```

+-----+
| NVIDIA-SMI 465.19.01    Driver Version: 460.32.03    CUDA Version: 11.2    |
+-----+-----+-----+
| GPU  Name          Persistence-M| Bus-Id        Disp.A | Volatile Uncorr. ECC |
| Fan  Temp  Perf  Pwr:Usage/Cap|      Memory-Usage | GPU-Util  Compute M. |
|                                       |                  |              MIG M. |
+-----+-----+-----+-----+
|   0   Tesla P100-PCIE...    Off  | 00000000:00:04:0 Off  |            0 |
| N/A   47C    P0     38W / 250W |  1629MiB / 16280MiB |      3%      Default |
|                                       |                  |              N/A   |
+-----+-----+-----+

```

```

+-----+
| Processes:
| GPU  GI    CI       PID   Type   Process name                      GPU Memory
|      ID    ID                                     Usage
+-----+-----+-----+-----+

```

Обучение

```
EPOCHS = 1
```

```
num_train_steps = len(train_dl) * EPOCHS
```

```

def configure_optimizers(model, lr):
    # В качестве оптимизатора AdamW с decay параметров
    param_optimizer = list(model.named_parameters())
    no_decay = ["bias", "gamma", "beta"]
    optimizer_grouped_parameters = [
        {
            "params": [p for n, p in param_optimizer if not any(nd in n for nd
↪ in no_decay)],
            "weight_decay_rate": 0.01
        },
        {
            "params": [p for n, p in param_optimizer if any(nd in n for nd in
↪ no_decay)],
            "weight_decay_rate": 0.0
        },
    ]
    optimizer = optim.AdamW(
        optimizer_grouped_parameters,
        lr=lr,
    )
    return optimizer

scaler = GradScaler()
optimizer = configure_optimizers(model, 2e-5)
loss_fn = nn.BCEWithLogitsLoss().to(device)
# Для улучшения также применяем warm up
scheduler = get_linear_schedule_with_warmup(optimizer, int(num_train_steps*.1),
↪ num_train_steps)

```

Функция для обучения на одну эпоху.

```
def train_epoch(model, data_loader, loss_fn, optimizer, device, scheduler):
    model = model.train()

    losses = []
    pred = []
    true = []

    pbar = tqdm(data_loader)
    for d in pbar:
        optimizer.zero_grad()

        # Перенос информации на видеокарту
        input_ids = d['input_ids'].to(device)
        attention_mask = d['attention_mask'].to(device)
        token_type_ids = d['token_type_ids'].to(device)
        targets = d["target"].to(device)

        with autocast():
            outputs = model(
                input_ids=input_ids,
                attention_mask=attention_mask,
                token_type_ids=token_type_ids
            )

            true += list(targets.cpu().numpy())
            pred += list(torch.round(F.sigmoid(outputs.detach())).cpu().numpy())

            loss = loss_fn(outputs, targets)
            losses.append(loss.detach().item())

        scaler.scale(loss).backward()
        scaler.unscale_(optimizer)
        nn.utils.clip_grad_norm_(model.parameters(), max_norm=1.0)
        scaler.step(optimizer) # выполняем шаг оптимизации
        scaler.update()
        scheduler.step()
        pbar.set_postfix({'loss': loss.detach().item()}) # будем выводить
        ↪ текущую loss чтобы следить за прогрессом

    return np.mean(losses), f1_score(true, pred, average='macro')

# аналогичная train функция, но для оценки
def eval_epoch(model, data_loader, loss_fn, device):
    model = model.eval()

    losses = []
    pred = []
    true = []

    with torch.no_grad():
        for d in tqdm(data_loader):
            input_ids = d['input_ids'].to(device)
```

```

attention_mask = d['attention_mask'].to(device)
token_type_ids = d['token_type_ids'].to(device)
targets = d["target"].to(device)

outputs = model(
    input_ids=input_ids,
    attention_mask=attention_mask,
    token_type_ids=token_type_ids
)

true += list(targets.cpu().numpy())
pred += list(torch.round(F.sigmoid(outputs.detach())).cpu().numpy())

loss = loss_fn(outputs, targets)
losses.append(loss.item())

return np.mean(losses), f1_score(true, pred, average='macro')

```

Запустим обучение.

```

%%time

print('Number of train steps:', num_train_steps)
for epoch in range(EPOCHS):
    print(f'Epoch {epoch + 1}/{EPOCHS}:')
    print('-' * 10)

    train_loss, train_f1 = train_epoch(
        model,
        train_dl,
        loss_fn,
        optimizer,
        device,
        scheduler
    )
    print(f'    loss: {train_loss:.4f}, f1: {train_f1:.4f} |', end=' ')

    val_loss, val_f1 = eval_epoch(
        model,
        val_dl,
        loss_fn,
        device
    )
    print(f'val_loss: {val_loss:.4f}, val_f1: {val_f1:.4f}')
    torch.save(model.state_dict(), f'bert_model_{epoch+1}.bin')

```

Number of train steps: 4681

Epoch 1/1:

HBox(children=(FloatProgress(value=0.0, max=4681.0), HTML(value='')))

loss: 0.2758, f1: 0.7118 |

HBox(children=(FloatProgress(value=0.0, max=292.0), HTML(value='')))


```
val_loss: 0.1845, val_f1: 0.7976
CPU times: user 1h 14s, sys: 45min 38s, total: 1h 45min 52s
Wall time: 1h 46min 19s
```

Предсказание на тестовой выборке

```
test['label'] = 0
```

Функция предсказания на данных, используя модель.

```
def predict(dl, model):
    model = model.eval()
    preds = []
    with torch.no_grad(): # без градиентов - только для предсказания
        for d in tqdm(dl):
            input_ids = d['input_ids'].to(device)
            attention_mask = d['attention_mask'].to(device)
            token_type_ids = d['token_type_ids'].to(device)

            outputs = model(
                input_ids=input_ids,
                attention_mask=attention_mask, token_type_ids=token_type_ids
            )
            outputs = (F.sigmoid(outputs).cpu()).tolist()
            preds += outputs
    return preds
```

```
test_dl = create_data_loader(test, tokenizer, SEQ_LEN, VAL_BATCH_SIZE,
    ↪ shuffle=False)
test['pred'] = predict(test_dl, model)
```

```
HBox(children=(FloatProgress(value=0.0, max=1059.0), HTML(value='')))
```

Подготовка решения в необходимом формате.

```
submission = []

for text_idx in tqdm(test['text_id'].unique()):
    pdict = {'idx':int(text_idx)}
    tmp = test[test['text_id'] == text_idx]
    tmp = tmp.iloc[np.argmax(tmp['pred'])]
    pdict['start'] = int(tmp['start'])
    pdict['end'] = int(tmp['end'])
    pdict['text'] = str(tmp['qtext'])

    submission.append(pdict)
submission = [json.dumps(e) for e in submission]
submission = '\n'.join(submission)
```

```
HBox(children=(FloatProgress(value=0.0, max=7257.0), HTML(value='')))
```

Сохранение решения.

```
with open('submission.jsonl', 'w', encoding='utf-8') as f:
    f.write(submission)
```

Посмотрим на случайные строки.

```
pd.DataFrame(test).sample(10)[['text', 'query', 'pred']]
```

	text	query	pred
7413	Близкий к турецкому правительству фонд SETA оп...	При этом авторы исследования критикуют, в част...	0.002272
32866	Дочь россиянки, которую планировали госпитализ...	Россиянка поехала в больницу в поселке @placeh...	0.002103
82735	Бывшего командира украинского националистическ...	@placeholder предоставили русскоязычного адвок...	0.017532
28317	Комиссия ООН по разоружению снова перенесла на...	17 февраля глава МИД России Сергей Лавров заяв...	0.667688
31452	В Совет Федерации поступили результаты голосов...	Накануне Совет Федерации одобрил президентский...	0.005855
92523	Правительство Украины расширило действие режим...	По последним данным, на @placeholder зарегистр...	0.885366
83964	Самая молодая миллиардерша в мире и предприним...	Подписчики обратили внимание на палец @placeho...	0.047775
69178	Министр энергетики России Александр Новак расс...	Спрос не падает, у нас объемы экспорта угля в ...	0.001590
57605	Официальный представитель министерства иностра...	Помимо сторон конфликта, в берлинской конферен...	0.060163
21684	Сотрудники военной контрразведки ФСБ задержали...	По его словам, к нему в СИЗО пришел следовател...	0.006590

Критерии определения победителей и призеров

Первый отборочный этап

В первом отборочном этапе участники решали задачи по двум предметам: математика и информатика, в каждом предмете максимально можно было набрать 100 баллов. Для того, чтобы пройти во второй этап, участники должны были набрать в сумме по обоим предметам не менее 50 баллов, независимо от уровня.

Второй отборочный этап

На втором отборочном этапе участники создавали алгоритм с использованием технологий искусственного интеллекта, который генерирует файл с ответами. Этот файл загружался на автоматизированную платформу, где сравнивался с верными ответами. По результатам проверки определялась точность решения по заданной метрике и формировался рейтинг участников (чем выше точность, тем выше позиция в рейтинге).

Проверочный файл был разделен на 2 примерно равные части (публичную и приватную). В ходе всего второго этапа участникам был доступен рейтинг на основании публичной части данных. После завершения этапа формировался итоговый рейтинг на основании проверки всех ответов (публичной и приватной части файла с ответами). На протяжении соревнования участники могли выбрать 2 решения, которые они считали наилучшими. В итоговый рейтинг шло то решение, которое показало наилучший результат. Если выбор не был сделан, в итоговый рейтинг шло решение, которое показало наилучший результат на публичной части данных.

После формирования итогового рейтинга у участников, чьи результаты полностью совпадали, был запрошен код решения для проверки на плагиат.

В заключительный этап были приглашены первые 100 участников итогового рейтинга, а также призеры и победители прошлого года по данному профилю. Если кто-то из финалистов отказывался от участия в соревновании, приглашался 101-й участник и далее по рейтингу. Общее число финалистов с учетом победителей и призеров прошлого года — 102 участника.

Проходной балл в заключительный этап — 0,6 из 1.

Заключительный этап

Заключительный этап состоял из двух туров: индивидуальный предметный тур и командный практический тур.

Индивидуальный предметный тур

Математика: максимально возможный балл за все задачи — 100 баллов.

Информатика: максимально возможный балл за все задачи — 100 баллов.

Командный практический тур

Командный практический тур — это решение задачи по машинному обучению командой из двух участников. На протяжении всего этапа на платформе соревнования для участников был доступен открытый рейтинг по одной половине датасета — публичная часть (проверочный файл был поделен случайным образом на 2 примерно равные части).

Рейтинг содержал следующие поля:

- место команды в рейтинге;
- название команды;
- лучший результат на публичной части данных.

Участники были ограничены 100 попытками в день. Попытка — возможность загрузить ответ и узнать точность решения.

В конце соревнования каждая команда могла в личном кабинете самостоятельно выбрать 2 решения, которые участники команды считали наилучшими. В таком случае в итоговую оценку шло одно из выбранных решений, показавшее лучший результат на полном наборе данных. Если команда не отмечала лучшие решения, в зачет шло решение, которое показало наилучший результат на публичной части данных.

$$\frac{\text{финальная точность} \times (\text{общее количество команд} + 1 - \text{место команды в рейтинге})}{\text{максимальная финальная точность среди всех команд} \times \text{общее количество команд}} \times 100$$

Человеческий и личный фактор в оценке задачи целиком исключен, вся оценка проводилась в автоматическом режиме.

После формирования итогового рейтинга команды, в которых участники претендовали на статус призера или победителя олимпиады, подтверждали свой балл на защите, которая состояла из проверки кода экспертным жюри и презентация решения участниками.

Командой-победителем заключительного этапа считалась команда, набравшая наибольшее количество баллов за решение задачи, т. е. первая в рейтинге команд, построенном по данной формуле.

Итоговые баллы участников в личном зачете заключительного этапа рассчитывались следующим образом:

$$\text{Итоговый балл} = \text{балл предметного тура по математике} \times 0,1 + \\ + \text{балл предметного тура по информатике} \times 0,1 + \text{балл за решение командной задачи} \times 0,8$$

Критерий определения победителей и призеров

По завершении соревнования был сформирован итоговый рейтинг учащихся независимо от их уровня в соответствии с итоговым личным баллом, полученным на заключительном этапе олимпиады. Согласно правилам РСОШ 24 первых участника в

рейтинге (25%) получили статусы призеров и победителей, участники, занявшие первые 7 мест рейтинга (8% от общего числа участников), получили статус победителя олимпиады, остальные 17 участников, занявшие с 8 по 24 места, — статус призера.

Критерий определения победителей и призеров (независимо от уровня)

Категория	Количество баллов
Победители	от 72,32 до 94,4
Призеры	от 56 до 71,75