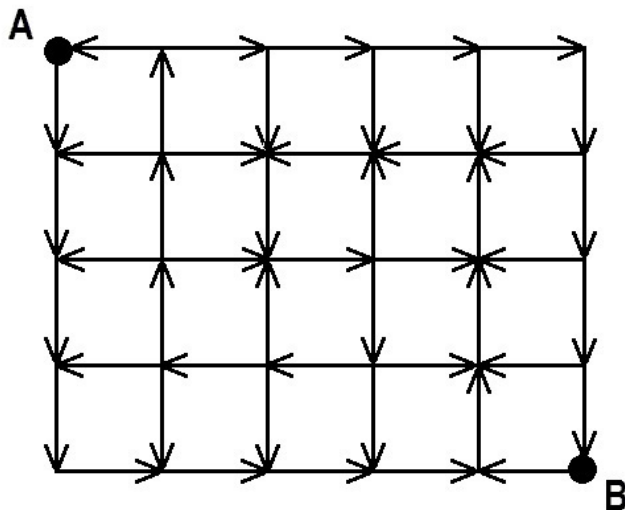


Первый отборочный этап

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 .

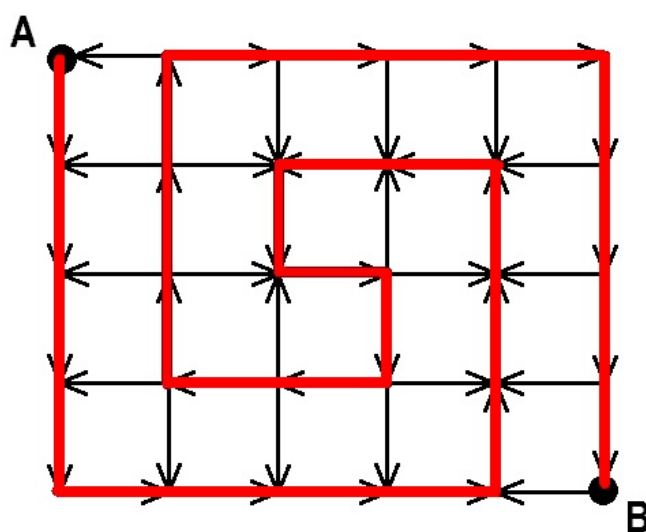
Формат входных данных

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.

Формат выходных данных

В первую строку вывести число отрезков улиц в самом коротком маршруте из точки A в точку B . Во вторую строку нужно выдать описание этого маршрута в виде последовательности символов n , s , w , e без пробелов. Если кратчайших маршрутов несколько, выдать самый первый среди них по алфавитному порядку. Гарантируется, что из точки A можно попасть в точку B .

Пояснения к примеру



Примеры

Пример №1

Стандартный ввод
4 5 weeee snssss wewww snsnns weeew snnsns wwwew ssssns eeeeew
Стандартный вывод
29 sssseeeennnwswseswnnnneeeessss

Пример программы-решения

Ниже представлено решение на языке C++.

```

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.

Формат входных данных

В первой строке содержится число N — количество столбиков. В следующей строке находятся N чисел $h_1, h_2, \dots, h_N$ через пробел, соответствующие высотам соответствующих столбиков. $1 \leq N \leq 10^5$, $1 \leq h_i \leq 2 \cdot 10^9$.

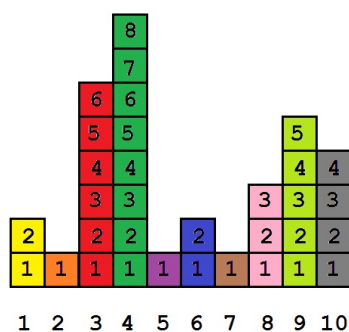
Формат выходных данных

Вывести N чисел через пробел, соответствующих высотам столбиков, после того как WALL-E закончит работу.

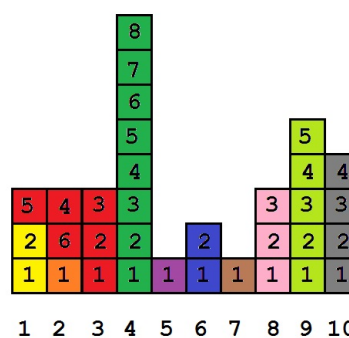
Пояснения к примеру

Следующая серия иллюстраций показывает, куда и какие кубики попали в процессе упорядочивания столбиков:

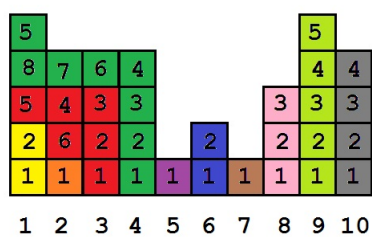
1. Стартовая ситуация



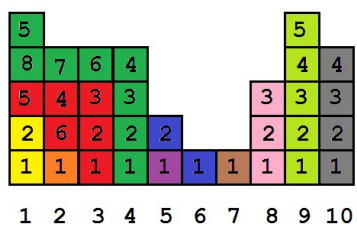
2. Упорядочили столбик 3



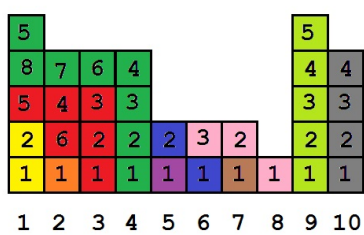
3. Упорядочили столбик 4



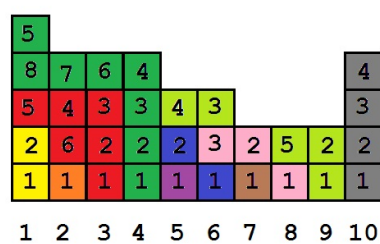
4. Упорядочили столбик 6



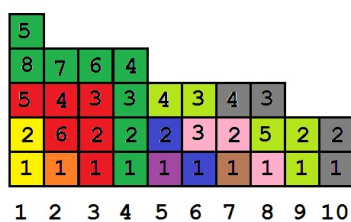
5. Упорядочили столбик 8



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

```

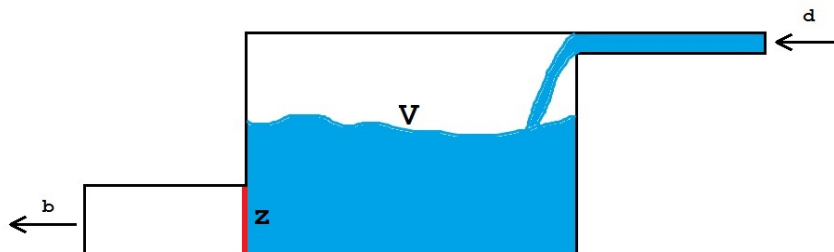


```

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

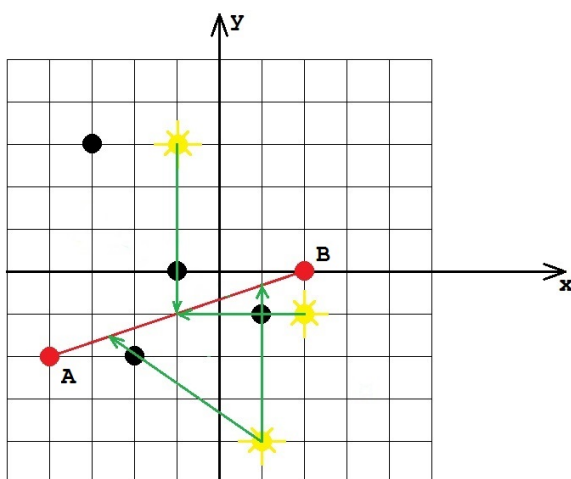
В первой строке содержится четыре целых числа через пробел XA , YA , XB , YB — координаты точек A и B . Во второй строке содержатся числа N и M , разделенные пробелом ($0 \leq N, M \leq 100$) — количество звезд и количество планет соответственно. В каждой из следующих N строк содержатся координаты очередной 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 вышеперечисленных 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 sqt2(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 = sqt2(A, B, s[i]);
42         for0(j, m){
43             int s1 = sqt2(A, B, p[j]);
44             int s2 = sqt2(A, p[j], s[i]);
45             int s3 = sqt2(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, бинарном поиск 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
аасааасааасааа

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 7]

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

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

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

Пример программы-решения

Ниже представлено решение на языке C++.

```

1  #include <bits/stdc++.h>
2
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 long double ld;
16 typedef pair<int, int> pii;
17
18 signed main(){
19     ios::sync_with_stdio(0), cin.tie(0), cout.tie(0);
20
21     int n;
22     cin >> n;
23     vector<int> v(n);
24     for0(i, n) cin >> v[i];
25
26     vector<int> suml(n), sumr(n);
27
28     int sum = 0;
29     for0(i, n){
30         sum += v[i];
31         suml[i] = sum;
32     }
33
34     sum = 0;
35     for(int i = n-1; i >= 0; i--){
36         sum += v[i];
37         sumr[i] = sum;
38     }
39
40     int ans = 0, o1 = 0;
41     if(sum % n == 0){
42         ans++;
43         o1++;
44     }
45
46     for(int i = 0; i < n-1; i++){
47         if(suml[i] % (i+1) == 0 && sumr[i+1] % (n - i - 1) == 0){
48             if(o1 == 1){
49                 if(suml[i] / (i+1) != sum / n) {
50                     ans++;
51                 }
52             }
53             else{
54                 ans++;
55             }
56         }
57     }
58 }

```

```

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.

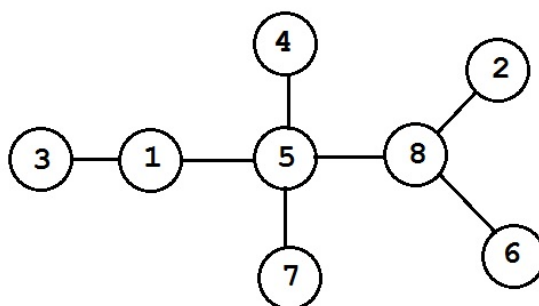
Формат входных данных

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 умножится, то есть перемножится. Особо следует отметить, что эффект умножения 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 **abcaabcaabcaabab**. Если из нее ничего не удалять, то ее наименьший период будет равен 15. Если удалить из ее конца одну букву, то период **abcaabcaabcaaba** не изменится, если же удалить две буквы, период **abcaabcaabcab** станет равен 10. Если удалить три буквы, период **abcaabcaabca** равен 7. И этого достаточно для первого теста. Чтобы период не превосходил 3, придется удалить из конца 10 символов и получить **abca** 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 abcaabcaabcaabab
Стандартный вывод
3

Пример программы-решения

Ниже представлено решение на языке C++.

```

1  //префикс-функция, метод Кнута-Морриса-Пракса
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,

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

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

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

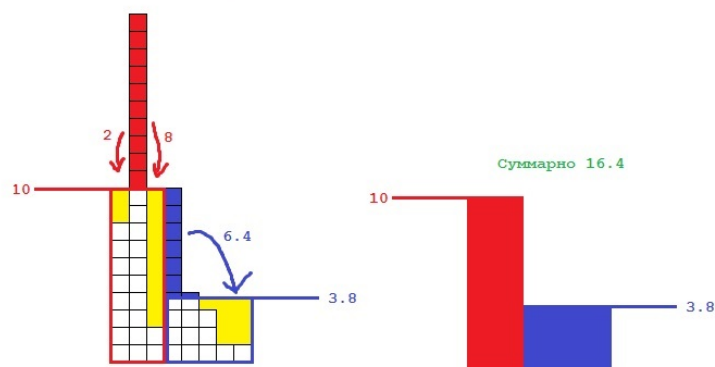
$$(8 + 20 + 2) / 3 = 10$$

на левом участке будет перемещено 10 единиц

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

на правом участке будет перемещено 6.4 единиц

$$\text{Всего будет перемещено } 10 + 6.4 = 16.4 \text{ единиц}$$



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

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

8 20 2 10 4 3 1 1

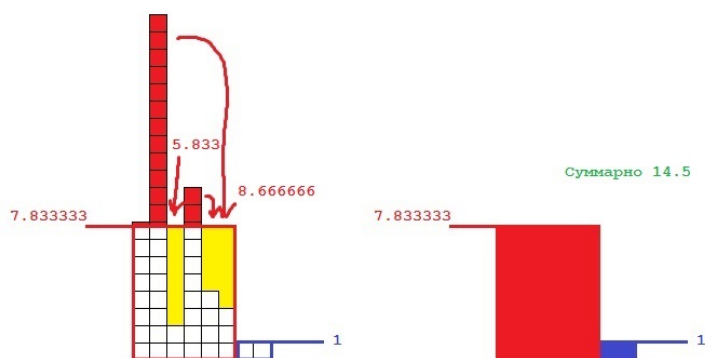
$$(8 + 20 + 2 + 10 + 4 + 3) / 6 = 7.8333333333$$

на левом участке будет перемещено 14.5 единиц

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

на правом участке будет перемещено 0 единиц

$$\text{Всего будет перемещено } 14.5 + 0 = 14.5 \text{ единиц}$$



$$a = 4, b = 4:$$

$a = 4$ $b = 4$

8 20 2 10 4 3 1 1

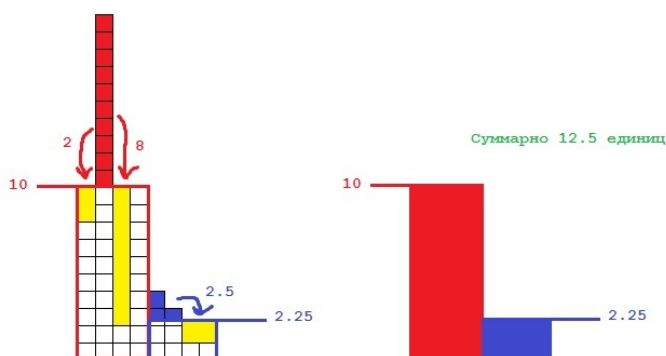
$(8 + 20 + 2 + 10) / 4 = 10$

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

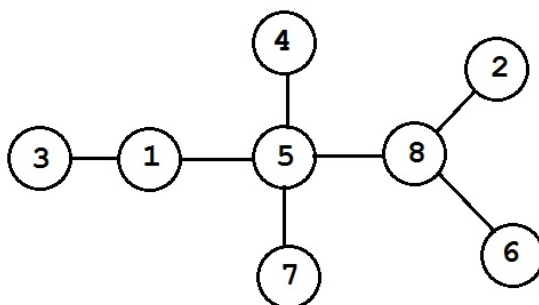
В первой строке содержится два числа через пробел: n — количество островов в архипелаге, $2 \leq n \leq 10^5$ и b — номер моста, для которого нужно узнать, сколько раз по нему проходит указанный маршрут. В следующих $n - 1$ строках содер-

жаты 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 «Т», «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:

- «Т» — 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 «Х».

Напишите программу для решения поставленной задачи.

Формат входных данных

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

Формат выходных данных

Один из символов «Т», «0», «1», «8», «-» или «Х».

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. География

Первая попытка. Задачи 9-11 класса

Привет, участник Олимпиады КД НТИ! Следующие 10 заданий по географии предназначены для учащихся 8-9 классов.

Время на решение ограничено временем открытия модуля (2 суток). После этого ответы на задачи приниматься не будут.

Если в процессе решения задач у тебя возникнет подозрение на ошибку в задании или какой-то существенный вопрос по условию, напиши авторам задачи через сообщения нашей группы vk.com/nticontest. В сообщении укажи ссылку на задачу.

Задача I.2.1.1. (10 баллов)

С помощью геопортала Яндекс Карты (<https://maps.yandex.ru/>), Google Карты (<https://www.google.com/maps>) или другого аналогичного сервиса определите как можно точнее географические координаты самой западной точки Крымского полуострова. Ответ выразите в градусах, минутах и секундах. Секунды округлите до целого числа. Впишите ответ в поля ниже.

_____ 1 _____ 2 _____ 3 (северной/южной широты) _____ 4
 _____ 5 _____ 6 _____ 7 (восточной/западной долготы) _____ 8

Ответ: 1 — 45; 2 — 23; 3 — 44 или 45; 4 — северной; 5 — 32; 6 — 28; 7 — 47; 8 — восточной.

Задача I.2.1.2. (10 баллов)

С помощью геопортала Яндекс Карты (<https://maps.yandex.ru/>), сервиса «Мои места» геопортала Google Карты (<https://www.google.com/maps>) или программы Google Планета Земля Pro (<https://www.google.com/intl/ru/earth/desktop/>) измерьте как можно точнее расстояние по поверхности Земли между самой северной и самой южной точками материка Евразия. Ответ выразите в километрах и округлите до целых единиц.

Ответ: 8501 ± 10 .

Задача I.2.1.3. (10 баллов)

Точка с географическими координатами $63^{\circ}17'00''$ северной широты и $123^{\circ}27'57''$ восточной долготы находится в акватории небольшого озера.

С помощью Конструктора карт (<https://yandex.ru/map-constructor>) геопортала Яндекс Карты (<https://maps.yandex.ru/>), сервиса «Мои места» геопортала Google Карты (<https://www.google.com/maps>), портала Kosmosnimki.RU (<https://kosmosnimki.ru/>), другого аналогичного сервиса или программы Google Планета Земля Pro (<https://www.google.com/intl/ru/earth/desktop/>) определите площадь данного озера.

Полученное значение площади представьте в гектарах, округлите до целых единиц и введите в поле ниже.

Ответ: $52,2 \pm 2$.

Задача I.2.1.4. (10 баллов)

Вокруг точки с географическими координатами $58^{\circ}17'09''$ северной широты и $94^{\circ}01'21''$ восточной долготы расположена вытянутая структура площадью около 100 гектаров.

С помощью геопорталов Яндекс Карты (<https://maps.yandex.ru/>), Kosmosnimki.RU (<https://kosmosnimki.ru/>) или других аналогичных инструментов рассмотрите космические снимки данной территории. Имейте виду, что на разных геопорталах могут быть разные снимки: разной детальности (пространственного разрешения) и разного времени.

В результате какого природного процесса или воздействия человека территория приобрела такой вид? Выберите только один вариант из представленных ниже.

1. Последствия наводнения / паводка.
2. Добыча рассыпного золота.
3. Расчистка леса под сельское хозяйство.
4. Подготовка к строительству гидроэлектростанции (расчистка ложа водохранилища).
5. Строительство судоходного канала.
6. Создание рыбопродуктивных прудов.
7. Таяние вечной мерзлоты.
8. Строительство аэродрома.
9. Строительство оросительных каналов.
10. Добыча нефти / газа.

Ответ: 2.

Задача I.2.1.5. (10 баллов)

На приведенных ниже спутниковых изображениях Вы видите различные пустыни нашей планеты. Все они достаточно крупные и общеизвестные. Введите их названия по-русски (в общепринятой транскрипции) в соответствующие поля ниже. Если в пределы снимка попадают части других пустынь, введите название той, которая занимает самую большую площадь в пределах снимка.

Эти изображения земной поверхности получены со спутников Landsat 7 и Landsat 8. Они не являются единовременным снимком, а получены с помощью синтеза ряда безоблачных снимков за 2019 год, обработанных таким образом, чтобы компенсировать всегда имеющиеся различия в положении солнца, положении спутника, прозрачности атмосферы и пр. Обработка проведена лабораторией GLAD (Global Land Analysis & Discovery — <https://glad.umd.edu/>) Географического факультета Университета Мэриленда (США). Данное изображение сделано с помощью портала Глобальные изменения лесного покрова (<http://earthenginepartners.appspot.com/science-2013-global-forest>).

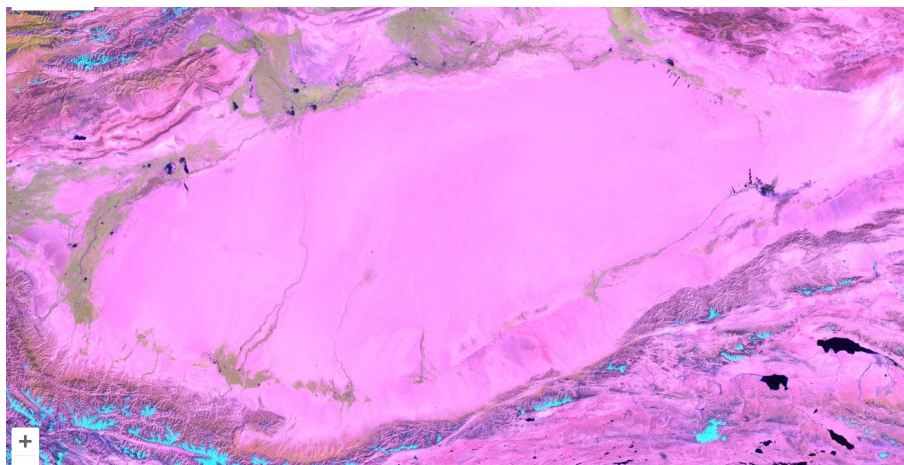
Изображение представляет интенсивность электромагнитного излучения, отраженного от поверхности земли и принятого аппаратурой спутников, в условных цветах: не видимый человеческим глазом коротковолновый инфракрасный свет представлен красным, ближний инфракрасный (также не различимый глазом) — зеленым, видимый красный свет — синим. В таком цветовом синтезе хорошо видны различия в растительности и влажности поверхности.

При этом чистые воды водоемов и водотоков выглядят темными, почти черными, так как равномерно поглощают большую часть солнечных лучей во всех указанных участках спектра. Если в воде имеется взвесь минеральных частиц или высокая плотность планктонных организмов, такие акватории выглядят как оттенки синего и голубого тонов. Такой же цвет могут иметь и мелководные участки, где сквозь толщу воды проходит солнечный свет, отраженный от дна водоема и/или от водной растительности.

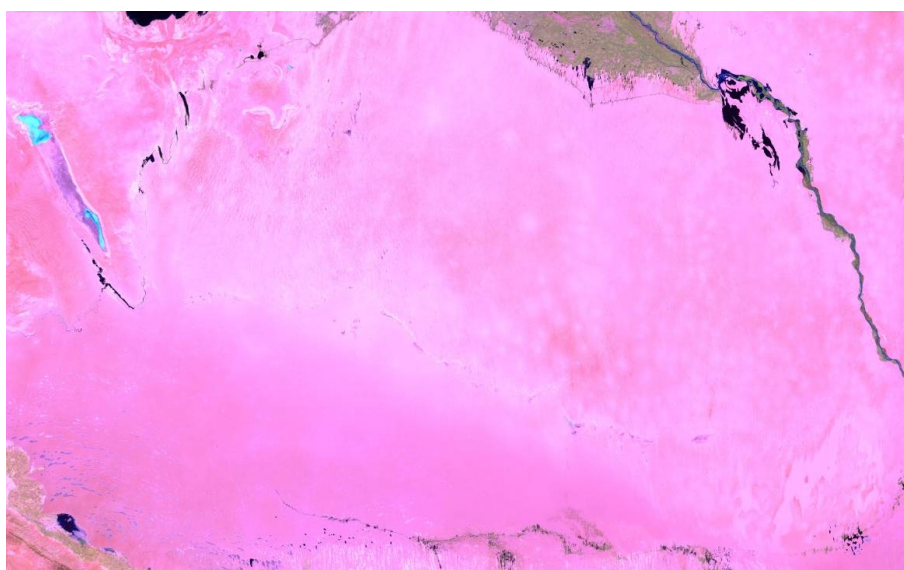
Участки суши, покрытые сомкнутой растительностью, как правило, выглядят при данном цветовом синтезе как различные оттенки зеленого. При этом, более темные и насыщенные тона, как правило, соответствуют древесной растительности, а более светлые — травянистой и кустарниковой.

Открытый грунт, в зависимости от его состава и характера, отображается различными оттенками красного и фиолетового. Частично покрытые растительностью участки (имеющие несомкнутый растительный покров) могут иметь различные промежуточные оттенки (розоватые, бурые и пр.). Голые скалы и песок, как правило, имеют различные оттенки голубого и синего. Близкие оттенки имеют и искусственные поверхности, созданные человеком: здания, сооружения, дороги с твердым покрытием и пр. (Что не удивительно: кирпич и бетон — основные строительные материалы.) Солончаки, а также лед и снег — практически белые или голубоватые (обычно — если покрыты водой). Песок тоже может выглядеть очень светлыми, почти такими же белыми, как снег или соль. Но песок может иметь и розовый цвет.

Пустыня 1:



Пустыня 2:



Пустыня 3:



Пустыня на космическом снимке 1: _____¹.

Пустыня на космическом снимке 2: _____².

Пустыня на космическом снимке 3: _____³.

Ответ: 1 — Такла – Макан; 2 — Каракумы; 3 — Намиб.

Задача I.2.1.6. (10 баллов)

Рассмотрите внимательно приведенные ниже спутниковые изображения различных природных объектов. Определите, к какой из трех категорий относится объекты на каждом изображении: солончаки, ледники или песчаные дюны. Поставьте в таблице ниже галочку в соответствующей колонке для каждого изображения. В каждой строке таблицы должна стоять только одна галочка.

Если на некоторых изображениях присутствуют, по Вашему мнению, два или все три типа объектов, выберите более крупный и компактный, расположенный вблизи центра изображения.

Эти изображения земной поверхности получены со спутников Landsat 5, Landsat 7 и Landsat 8. Они не являются единовременным снимком, а получены с помощью

синтеза ряда безоблачных снимков за год (использовались снимки 2000 и 2019 годов), обработанных таким образом, чтобы компенсировать всегда имеющиеся различия в положении солнца, положении спутника, прозрачности атмосферы и пр. Обработка проведена лабораторией GLAD (Global Land Analysis & Discovery — <https://glad.umd.edu/>) Географического факультета Университета Мэриленда (США). Данное изображение сделано с помощью портала Глобальные изменения лесного покрова (<http://earthenginepartners.appspot.com/science-2013-global-forest>).

Изображение представляет интенсивность электромагнитного излучения, отраженного от поверхности земли и принятого аппаратурой спутников, в условных цветах: не видимый человеческим глазом коротковолновый инфракрасный свет представлен красным, ближний инфракрасный (также не различимый глазом) — зеленым, видимый красный свет — синим. В таком цветовом синтезе хорошо видны различия в растительности и влажности поверхности.

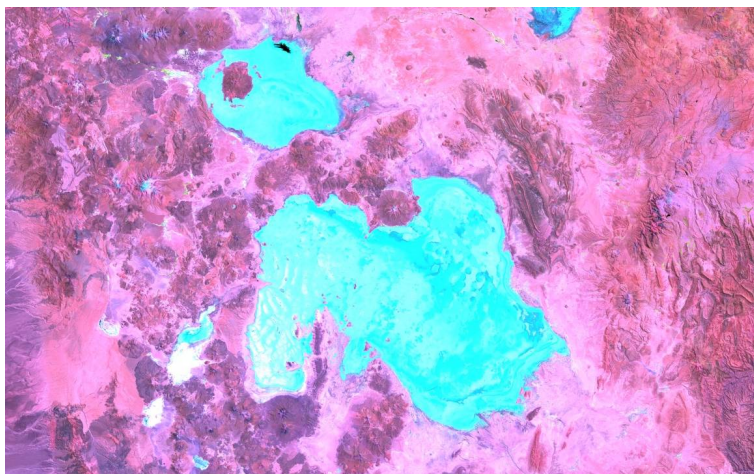
При этом чистые воды водоемов и водотоков выглядят темными, почти черными, так как равномерно поглощают большую часть солнечных лучей во всех указанных участках спектра. Если в воде имеется взвесь минеральных частиц или высокая плотность планктонных организмов, такие акватории выглядят как оттенки синего и голубого тонов. Такой же цвет могут иметь и мелководные участки, где сквозь толщу воды проходит солнечный свет, отраженный от дна водоема и/или от водной растительности.

Участки суши, покрытые сомкнутой растительностью, как правило, выглядят при данном цветовом синтезе как различные оттенки зеленого. При этом, более темные и насыщенные тона, как правило, соответствуют древесной растительности, а более светлые — травянистой и кустарниковой.

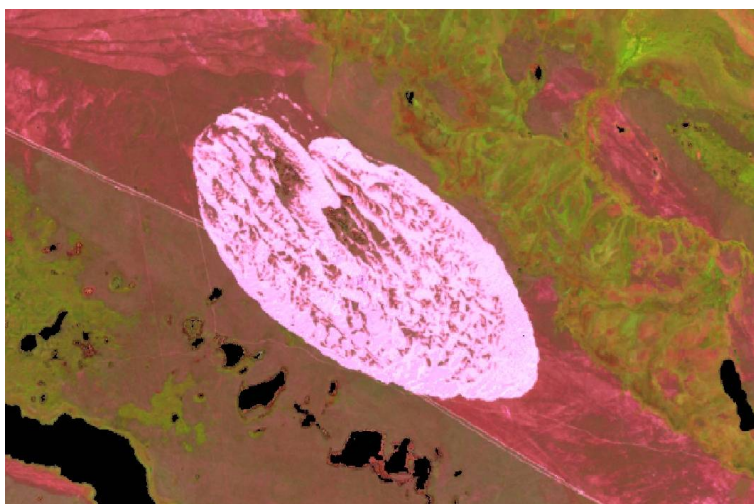
Открытый грунт, в зависимости от его состава и характера, отображается различными оттенками красного и фиолетового. Частично покрытые растительностью участки (имеющие несомкнутый растительный покров) могут иметь различные промежуточные оттенки (розоватые, бурые и пр.). Красноватые тона может иметь и болотная растительность. Голые скалы и песок, как правило, имеют различные оттенки голубого и синего. Близкие оттенки имеют и искусственные поверхности, созданные человеком: здания, сооружения, дороги с твердым покрытием и пр. (Что не удивительно: кирпич и бетон — основные строительные материалы.) Солончаки, а также лед и снег — практически белые или голубоватые (обычно — если покрыты водой). Песок и бетон тоже могут выглядеть очень светлыми, почти такими же белыми, как снег или соль. Но песок может иметь и розовый цвет.

Поэтому для отнесения объектов к той или иной категории недостаточно просто посмотреть их цвет на изображениях — необходимо проанализировать весь окружающий ландшафт, а также форму и текстуру объектов.

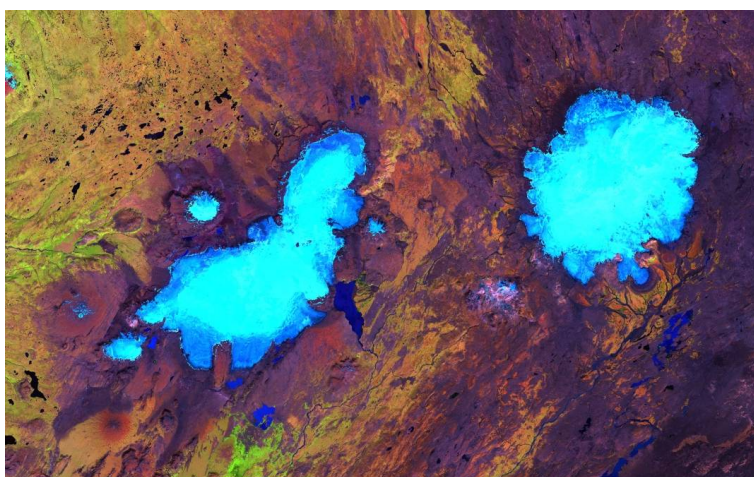
Изображение 1:



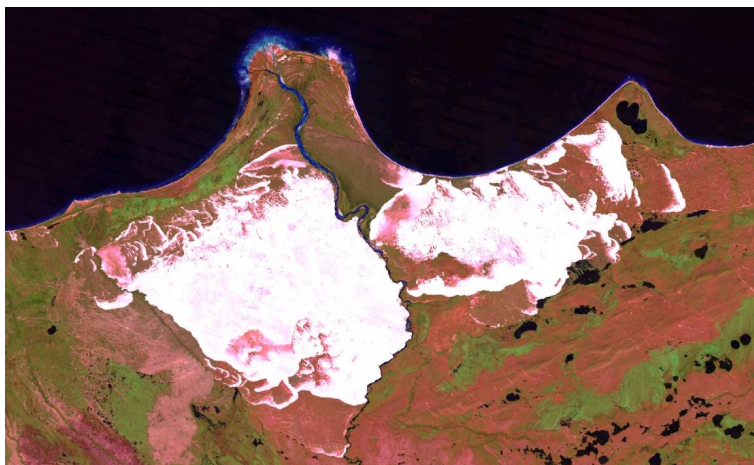
Изображение 2:



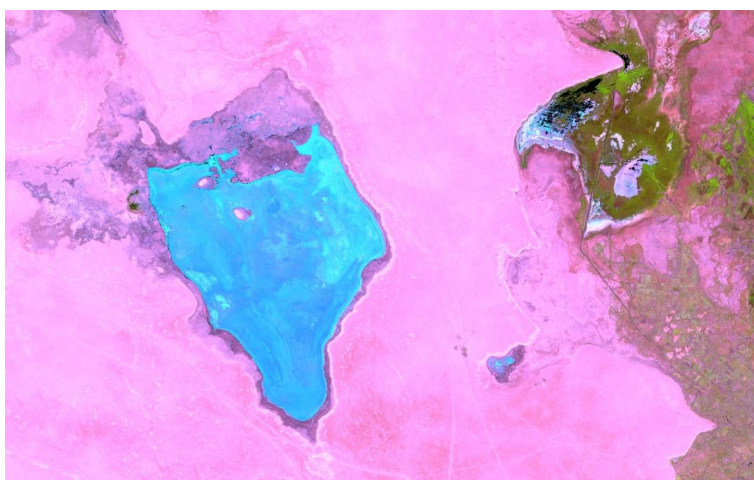
Изображение 3:



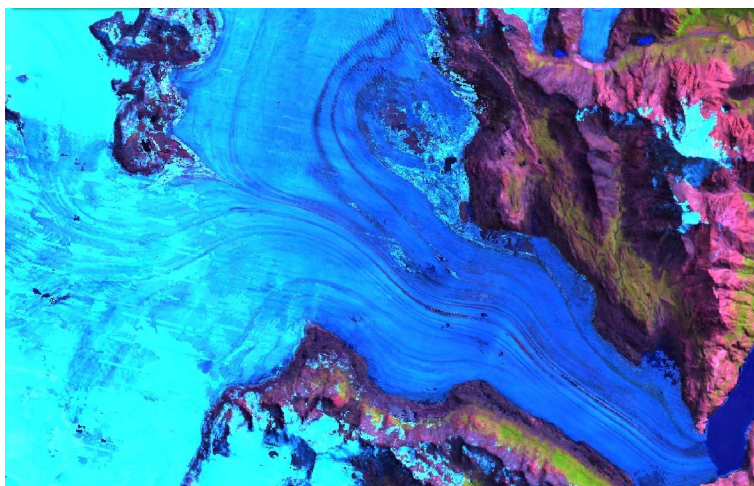
Изображение 4:



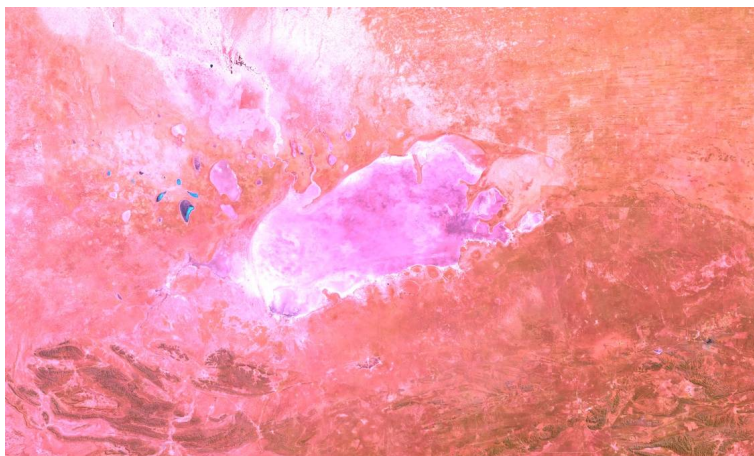
Изображение 5:



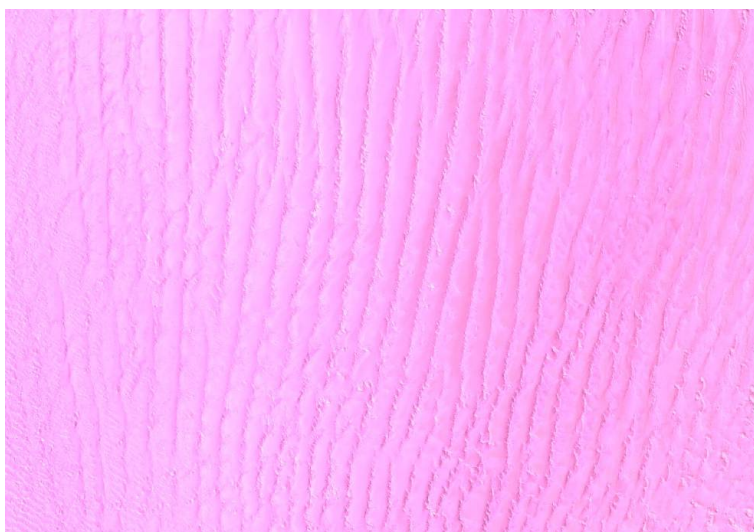
Изображение 6:



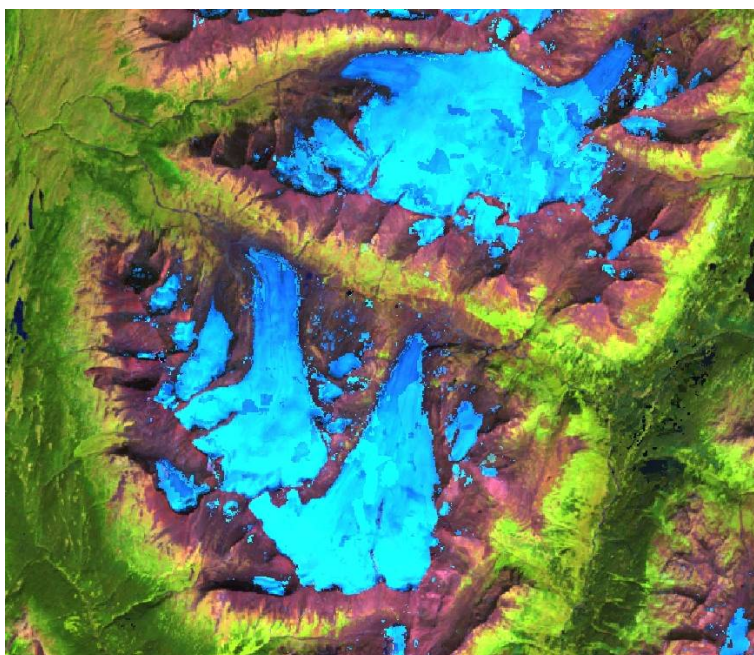
Изображение 7:



Изображение 8:



Изображение 9:



Отметьте верные ячейки

Спутниковые изображения	Солончаки	Ледники	Песчаные дюны
Изображение 1	1	2	3
Изображение 2	4	5	6
Изображение 3	7	8	9
Изображение 4	10	11	12
Изображение 5	13	14	15
Изображение 6	16	17	18
Изображение 7	19	20	21
Изображение 8	22	23	24
Изображение 9	25	26	27

Ответ: 1; 6; 8; 12; 13; 17; 19; 24; 26.

Задача I.2.1.7. (10 баллов)

Количество и расположение поверхностных вод (как материковых, так и прибрежных) — важнейшая характеристика любой территории. От них во многом зависит локальный климат, биологическое разнообразие и благополучие людей. В свою очередь поверхностные воды сами находятся под влиянием климата и хозяйственной деятельности человека. Некоторые территории, ранее покрытые водой, теряют свою водную поверхность в результате пересыхания или изменений береговой линии. Другие, напротив, из участков суши превращаются в акватории. Многие территории заливаются водой только в определенный сезон. Сегодня Вы можете познакомиться с результатами спутникового мониторинга поверхностных вод с помощью общедоступных онлайн-инструментов, например, Global Surface Water Explorer (<https://global-surface-water.appspot.com/>) и Freshwater Ecosystems Explorer (<https://www.sd661.app/>). (Последний, вопреки названию, содержит также информацию о соленых озерах и прибрежных морских акваториях.)

Опираясь на данные этих порталов, расположите страны из нижеприведенного списка по уменьшению абсолютной площади территорий, которые, согласно данным наблюдения из космоса, были постоянно покрыты водой в 2000-2004 году (permanent water), но на 2019 год (точнее — на период 2015-2019) уже являются сушей и не покрываются водой даже сезонно (lost permanent).

1. Бразилия.
2. Китай.
3. Туркменистан.
4. Боливия.
5. Россия.
6. Узбекистан.
7. США.
8. Канада.
9. Иран.
10. Аргентина.
11. Казахстан.

Ответ: 11; 5 ; 6; 8; 10; 7; 2; 1; 9; 4.

Задача I.2.1.8. (10 баллов)

Довольно большой участок земной поверхности вокруг точки с географическими координатами 41°53'00»северной широты и 53°07'00» восточной долготы круглогодично является акваторией (по состоянию на 2019 год). Однако, он не всегда был покрыт водой: в конце XX-го века были периоды, когда этот участок был сушей.

Воспользуйтесь порталом Global Surface Water Explorer (<https://global-surface-water.appspot.com/>) и/или порталами, которые позволяют рассматривать онлайн космические снимки за разные периоды времени (например, LandLook Viewer (<https://landlook.usgs.gov/viewer.html>) или GloVis (<https://glovis.usgs.gov/>) Геологической службы США, USGS). Определите год (четырёхзначное число), когда участок вблизи указанной точки впервые вновь был покрыт водой в течение всего года. Лучше проверить Ваши выводы, по крайней мере, по двум разным источникам.

Ответ: 1994.

Задача I.2.1.9. (10 баллов)

Определите страну по описанию. В поле ниже впишите ее полное название по-русски.

Согласно данным Продовольственной и сельскохозяйственной организации Объединенных наций (UN Food and Agriculture Organization) на 2016 год, приведенным на сайте Всемирного банка (<https://data.worldbank.org/>), сельскохозяйственные земли занимают менее 4% территории этой страны, хотя в сельской местности, по данным из того же сайта, проживает более половины ее весьма многочисленного населения. (По этому последнему показателю страна входит в тройку крупнейших на своем континенте.)

Из 14 биомов (<https://ru.wikipedia.org/wiki/%D0%91%D0%B8%D0%BE%D0%BC>), выделяемых Всемирным фондом дикой природы (WWF) в рамках глобальной классификации наземных экорегионов (<https://www.worldwildlife.org/biome-categories/terrestrial-ecoregions>) (см.также таблицу и одну из карт), в пределах территории данной страны представлены три. При этом один из них занимает более трех ее четвертей.

Согласно данным портала Всемирной лесной вахты (<https://www.globalforestwatch.org/>) (Global Forest Watch), древесная растительность (с сомкнутостью древесного полога 10% и выше) занимала, по состоянию на 2010 год, чуть более 1% территории этой страны. Лишь незначительную часть этой растительности Продовольственная и сельскохозяйственная организация Объединенных наций (UN Food and Agriculture Organization) считает лесом (<http://www.fao.org/3/I8661RU/i8661ru.pdf>).

Крупнейшее водно-болотное угодье международного значения, включенное в список Рамсарской конвенции (<https://www.ramsar.org/>), на территории этой страны также является ее единственным природным объектом, включенным в список Всемирного наследия ЮНЕСКО (<https://whc.unesco.org/ru/list/>). В этот список

территория попала благодаря найденным здесь ископаемым остаткам древних китов.

Страна находится в острой фазе дипломатического конфликта с одним из соседних государств по вопросам использования важнейших трансграничных водных ресурсов. По состоянию на конец лета 2020 года позиции сторон сблизить не удалось. Со стороны официальных лиц страны неоднократно звучали угрозы решить конфликт с помощью военной силы.

Ответ: Египет.

Задача I.2.1.10. (баллов)

Наблюдение Земли из космоса (дистанционное зондирование) широко используется сегодня как инструмент изучения нашей планеты. Современные спутники дистанционного зондирования фиксируют интенсивность электромагнитного излучения, отраженного от поверхности Земли, не только в видимом свете, но и в тех частях спектра электромагнитных волн, которые не различимы человеческим глазом: в ультрафиолетовом, ближнем инфракрасном, средневолновом инфракрасном и дальнем инфракрасном. (Радарные спутники также фиксируют радиоволновое излучение.) Электромагнитные волны разной длины по-разному отражаются от поверхности Земли и поэтому несут разную информацию о ней. Аппаратура современных спутников обычно измеряет и записывает яркость излучения в разных диапазонах длин волн (в разных спектральных каналах) независимо. Число таких каналов может быть различно. Как правило, их не меньше 3-4-х. А иногда — 10 и больше.

Используя три любых спектральных канала, можно изучать снимок как цветное изображение на компьютере. При этом зафиксированную аппаратурой спутника яркость электромагнитного излучения в одном канале представляют как яркость красного цвета, в другом — как яркость зеленого цвета, в третьем — как синего. Такие изображения Земли из космоса называют мультиспектральными (многоканальными). Если складывать мультиспектральное изображение из каналов в диапазоне видимого света и отображать яркость в сине-голубом диапазоне синим цветом, в желто-зеленом — зеленым цветом и в красном диапазоне — красным, то такое изображение («в естественных цветах») будет похоже на обычную цифровую фотографию — как если бы сами фотографировали Землю со спутника цифровым фотоаппаратом. Примерно в таком виде космические снимки размещаются на геопорталах общего пользования — таких как «Яндекс карты» и «Google карты».

Однако, можно отображать на компьютере синим, зеленым и красным цветами не обязательно яркости излучения в видимой части электромагнитного спектра — можно использовать яркости в тех диапазонах, которые человеческий глаз не различает — в ультрафиолетовом и инфракрасном. Часто это позволяет нам увидеть такую информацию о нашей планете, которую даже космонавты не могут увидеть с орбиты собственными глазами. Конечно, такие изображения не будут выглядеть как привычные нам фотографии — это будут изображения в условных цветах. Но часто, они несут больше информации о поверхности нашей планеты, чем изображения в видимой части спектра.

Рассмотрите приведенные ниже мультиспектральные изображения участка территории центра Европейской части России, сделанные спутником Landsat 8 22-го ноября 2019 года (глубокой осенью). Аппаратура этого спутника фиксирует яркость

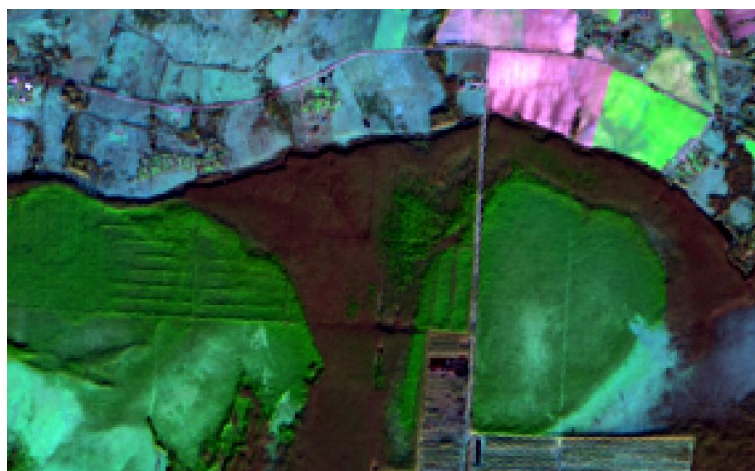
электромагнитного излучения в 10-ти разных спектральных каналах. Описание этих каналов можно найти, например, [здесь](#) или [здесь](#).

На первом изображении желтыми цифрами обозначены различные объекты (однородные участки местности), список которых приведен ниже. Используя знания о географии нашей страны и информацию о спектральных каналах данного спутника, поставьте в соответствие каждому номеру участка его характеристику. Про использование различных комбинаций спектральных каналов спутников Landsat для изучения Земли можно прочесть, например, [здесь](#) и [здесь](#).

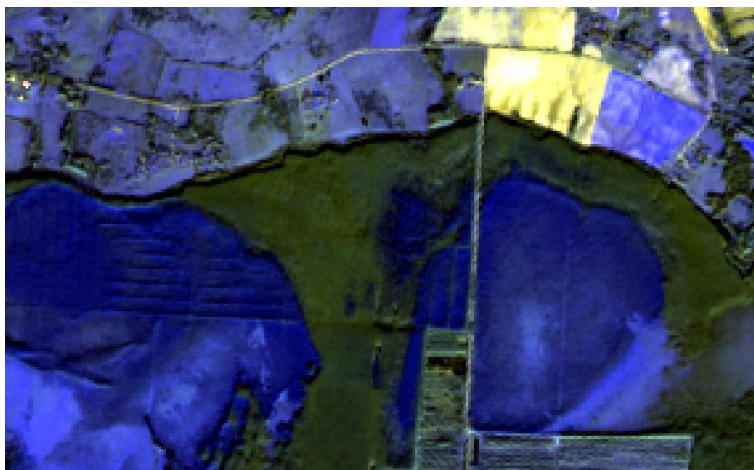
Изображение 1: красным цветом представлена интенсивность излучения в канале 4, зеленым — в канале 3, синим — в канале 2.



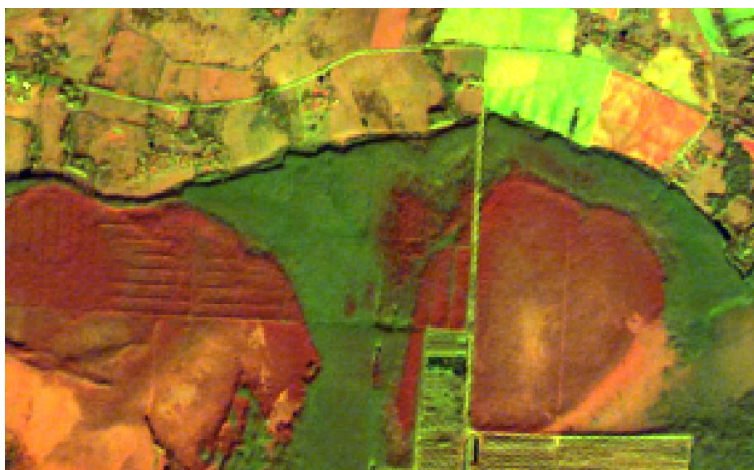
Изображение 2: красным цветом представлена интенсивность излучения в канале 6, зеленым — в канале 5, синим — в канале 4.



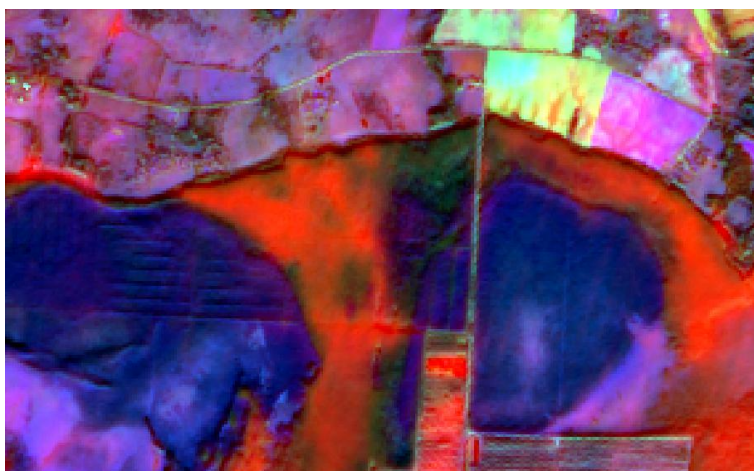
Изображение 3: красным цветом представлена интенсивность излучения в канале 7, зеленым — в канале 6, синим — в канале 5.



Изображение 4: красным цветом представлена интенсивность излучения в канале 5, зеленым — в канале 6, синим — в канале 9.



Изображение 5: красным цветом представлена интенсивность излучения в канале 10, зеленым — в канале 6, синим — в канале 5.



Сопоставьте значения из двух списков:

Участок 1	а. Переходное болото с сосной и кустарничками
Участок 2	б. Зброшенныя поля на осушенном болоте, частично зарастают лиственными деревьями
Участок 3	с. Участок более старогO хвойного леса
Участок 4	д. Лиственный лес, возможно заболоченный
Участок 5	е. Распаханное и/или сжатое поле
Участок 6	ф. Открытое верховое сфагновое болото
Участок 7	г. Населенный пункт (деревня)
Участок 8	h. Поле, засеянное озимыми
Участок 9	и. Зброшенное поле, зарастающее лиственными деревьями
Участок 10	j. Относительно молодой хвойный лес по осушенному болоту

Ответ: 1 — j; 2 — и; 3 — г; 4 — д; 5 — с; 6 — е; 7 — h; 8 — а; 9 — ф; 10 — б.

Вторая попытка. Задачи 9-11 класса

Привет, участник Олимпиады КД НТИ! Следующие 10 заданий по географии предназначены для учащихся 8-9 классов.

Время на решение ограничено временем открытия модуля (2 суток). После этого ответы на задачи приниматься не будут.

Если в процессе решения задач у тебя возникнет подозрение на ошибку в задании или какой-то существенный вопрос по условию, напиши авторам задачи через сообщения нашей группы vk.com/nticontest. В сообщении укажи ссылку на задачу.

Задача I.2.2.1. (10 баллов)

С помощью геопортала Яндекс Карты (<https://maps.yandex.ru/>), Google Карты (<https://www.google.com/maps>) или другого аналогичного сервиса определите как можно точнее географические координаты самой восточной точки острова Мадагаскар. Ответ выразите в градусах, минутах и секундах. Секунды округлите до целого числа. Впишите ответ в поля ниже.

Вследствие приливов и отливов в Индийском океане крайняя восточная точка суши несколько смещается в течение суток. Выберите точку при самом низком уровне воды (при отливе), видимом на доступных на геопорталах космических снимках высокого разрешения.

_____ 1 _____ 2 _____ 3 (северной/южной широты) _____ 4
 _____ 5 _____ 6 _____ 7 (восточной/западной долготы) _____ 8

Ответ: 1 — 15; 2 — 15; 3 — 55; 4 — южной широты; 5 — 50; 6 — 29; 7 — 10 ± 1 ; 8 — восточной долготы.

Задача I.2.2.2. (10 баллов)

С помощью геопортала Яндекс Карты (<https://maps.yandex.ru/>), сервиса «Мои места» геопортала Google Карты (<https://www.google.com/maps>) или программы Google Планета Земля Pro (<https://www.google.com/intl/ru/earth/desktop/>) измерьте как можно точнее расстояние по поверхности Земли между двумя точками со следующими географическими координатами.

Точка 1: 59°09'43» северной широты и 36°35'18» восточной долготы.

Точка 2: 61°21'30» северной широты и 58°37'03» восточной долготы.

Ответ выразите в километрах и округлите до целых единиц.

Ответ: 1236,5 ± 2.

Задача I.2.2.3. (10 баллов)

Точка с географическими координатами 70°04'00» северной широты и 70°20'00» восточной долготы находится в акватории достаточно крупного озера, одного из крупнейших озер полуострова Ямал.

С помощью Конструктора карт (<https://yandex.ru/map-constructor>) геопортала Яндекс Карты (<https://maps.yandex.ru/>), сервиса «Мои места» геопортала Google Карты (<https://www.google.com/maps>), портала Kosmosnimki.RU (<https://kosmosnimki.ru/>), другого аналогичного сервиса или программы Google Планета Земля Pro (<https://www.google.com/intl/ru/earth/desktop/>) определите площадь данного озера.

Полученное значение площади представьте в квадратных километрах, округлите до целых единиц и введите в поле ниже.

Ответ: 200 ± 5.

Задача I.2.2.4. (10 баллов)

Точка с географическими координатами 37°33'57» северной широты и 105°02'28» восточной долготы расположена в пределах участка земной поверхности характерного внешнего вида площадью около 20 гектаров. Общая площадь аналогичных участков в данном районе превышает десять квадратных километров.

С помощью геопорталов Яндекс Карты (<https://maps.yandex.ru/>), Kosmosnimki.RU (<https://kosmosnimki.ru/>), геопортала Google Карты (<https://www.google.com/maps>), программы Google Планета Земля Pro (<https://www.google.com/intl/ru/earth/desktop/>) или других аналогичных инструментов рассмотрите космические снимки на данную территорию.

Что из себя представляет данная территория? В результате какого природного процесса или воздействия человека она приобрела такой вид? Выберите только один вариант из представленных ниже.

Выберите один вариант из списка

1. рисовые чеки;

2. плантации эвкалиптов и/или других быстрорастущих деревьев;
3. чайные плантации;
4. территория, занятая солнечными батареями (солнечная электростанция);
5. плантации масличной пальмы;
6. марикультура: выращивание рыбы / морских беспозвоночных;
7. добыча полезных ископаемых;
8. очистные сооружения крупного населенного пункта;
9. фруктовые сады;
10. пастбища и загоны для скота.

Ответ: 4.

Задача I.2.2.5. (10 баллов)

На приведенных ниже спутниковых изображениях Вы видите различные острова и архипелаги нашей страны. Все они не самые крупные, но достаточно известные. Введите их названия по-русски (в общепринятой транскрипции) в соответствующие поля ниже. Если в пределы снимка попадают части других островов/архипелагов, введите название того/тех, который/ые помещаются на снимке целиком и/или занимает самую большую площадь в пределах снимка.

Эти изображения земной поверхности получены со спутников Landsat 7 и Landsat 8. Они не являются единовременным снимком, а получены с помощью синтеза ряда безоблачных снимков за 2019 год, обработанных таким образом, чтобы компенсировать всегда имеющиеся различия в положении солнца, положении спутника, прозрачности атмосферы и пр. Обработка проведена лабораторией GLAD (Global Land Analysis & Discovery — <https://glad.umd.edu/>) Географического факультета Университета Мэриленда (США). Данное изображение сделано с помощью портала Глобальные изменения лесного покрова (<http://earthenginepartners.appspot.com/science-2013-global-forest>).

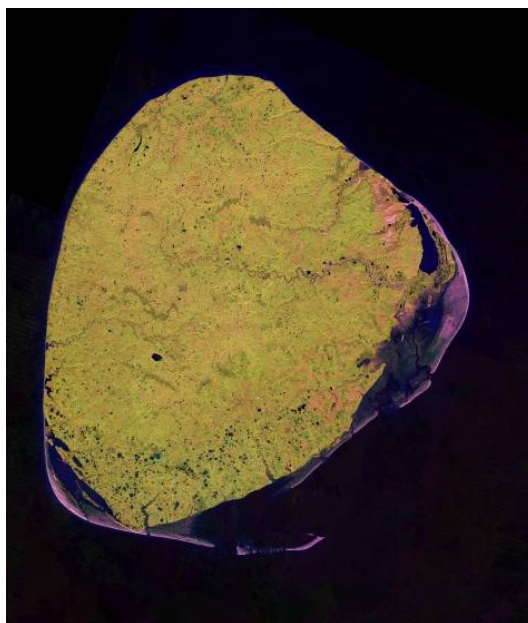
Изображение представляет интенсивность электромагнитного излучения, отраженного от поверхности земли и принятого аппаратурой спутников, в условных цветах: не видимый человеческим глазом коротковолновый инфракрасный свет представлен красным, ближний инфракрасный (также не различимый глазом) — зеленым, видимый красный свет — синим. В таком цветовом синтезе хорошо видны различия в растительности и влажности поверхности.

При этом чистые воды водоемов и водотоков выглядят темными, почти черными, так как равномерно поглощают большую часть солнечных лучей во всех указанных участках спектра. Если в воде имеется взвесь минеральных частиц или высокая плотность планктонных организмов, такие акватории выглядят как оттенки синего и голубого тонов. Такой же цвет могут иметь и мелководные участки, где сквозь толщу воды проходит солнечный свет, отраженный от дна водоема и/или от водной растительности.

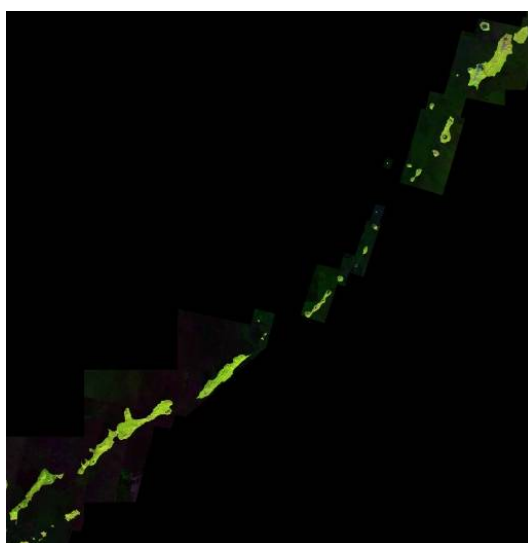
Участки суши, покрытые сомкнутой растительностью, как правило, выглядят при данном цветовом синтезе как различные оттенки зеленого. При этом, более темные и насыщенные тона, как правило, соответствуют древесной растительности, а более светлые — травянистой и кустарниковой.

Открытый грунт, в зависимости от его состава и характера, отображается различными оттенками красного и фиолетового. Частично покрытые растительностью участки (имеющие несомкнутый растительный покров) могут иметь различные промежуточные оттенки (розоватые, бурые и пр.). Голые скалы и песок, как правило, имеют различные оттенки голубого и синего. Близкие оттенки имеют и искусственные поверхности, созданные человеком: здания, сооружения, дороги с твердым покрытием и пр. (Что не удивительно: кирпич и бетон — основные строительные материалы.) Солончаки, а также лед и снег — практически белые или голубоватые (обычно — если покрыты водой). Песок тоже может выглядеть очень светлыми, почти такими же белыми, как снег или соль. Но песок может иметь и розовый цвет.

Остров/архипелаг 1:



Остров/архипелаг 2:



Остров/архипелаг 3:



Заполните пропуски

Остров на космическом снимке 1: _____¹.

Архипелаг на космическом снимке 2: _____².

Архипелаг на космическом снимке 3: _____³.

Ответ: 1 — Колгуев; 2 — Курильские острова; 3 — Олений остров.

Задача I.2.2.6. (10 баллов)

Рассмотрите внимательно приведенные ниже спутниковые изображения различных природных объектов. Определите, к какой из трех категорий относится объекты на каждом изображении: природные болота, горы или сельхозугодья. Поставьте в таблице ниже галочку в соответствующей колонке для каждого изображения. В каждой строке таблицы должна стоять только одна галочка.

Если на некоторых изображениях присутствуют, по Вашему мнению, два или все три типа объектов, выберите более крупный и компактный, расположенный вблизи центра изображения.

Эти изображения земной поверхности получены со спутников Landsat 5, Landsat 7 и Landsat 8. Они не являются единовременным снимком, а получены с помощью синтеза ряда безоблачных снимков за год (использовались снимки 2000 и 2019 годов), обработанных таким образом, чтобы компенсировать всегда имеющиеся различия в положении солнца, положении спутника, прозрачности атмосферы и пр. Обработка проведена лабораторией GLAD (Global Land Analysis & Discovery — <https://glad.umd.edu/>) Географического факультета Университета Мэриленда (США). Данное изображение сделано с помощью портала Глобальные изменения лесного покрова (<http://earthenginepartners.appspot.com/science-2013-global-forest>).

Изображение представляет интенсивность электромагнитного излучения, отраженного от поверхности земли и принятого аппаратурой спутников, в условных цветах: не видимый человеческим глазом коротковолновый инфракрасный свет представлен красным, ближний инфракрасный (также не различимый глазом) — зеленым, видимый красный свет — синим. В таком цветовом синтезе хорошо видны различия в растительности и влажности поверхности.

При этом чистые воды водоемов и водотоков выглядят темными, почти черными, так как равномерно поглощают большую часть солнечных лучей во всех указан-

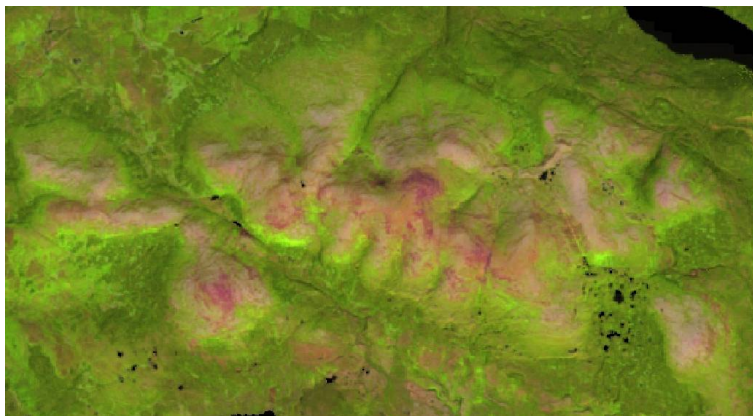
ных участках спектра. Если в воде имеется взвесь минеральных частиц или высокая плотность планктонных организмов, такие акватории выглядят как оттенки синего и голубого тонов. Такой же цвет могут иметь и мелководные участки, где сквозь толщу воды проходит солнечный свет, отраженный от дна водоема и/или от водной растительности.

Участки суши, покрытые сомкнутой растительностью, как правило, выглядят при данном цветовом синтезе как различные оттенки зеленого. При этом, более темные и насыщенные тона, как правило, соответствуют древесной растительности, а более светлые — травянистой и кустарниковой. Открытые болота, покрытые мхом, могут иметь светло-зеленый, белесый, зеленовато-бурый оттенки.

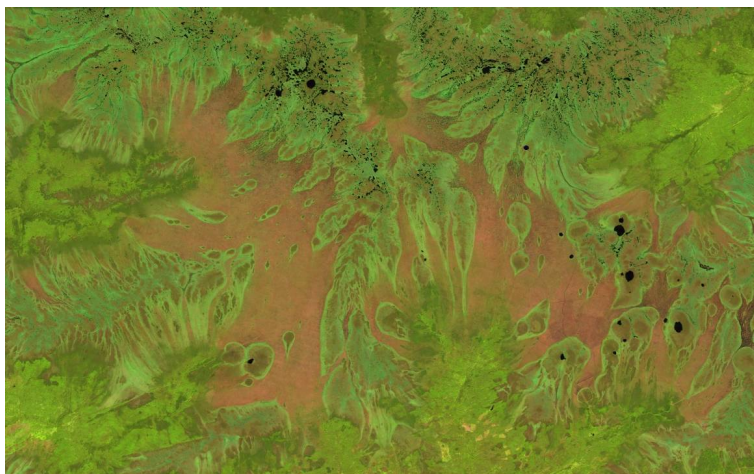
Открытый грунт, в зависимости от его состава и характера, отображается различными оттенками красного и фиолетового. Частично покрытые растительностью участки (имеющие несомкнутый растительный покров) могут иметь различные промежуточные оттенки (розоватые, бурые и пр.). Красноватые тона может иметь и болотная растительность. Голые скалы и песок, как правило, имеют различные оттенки голубого и синего. Близкие оттенки имеют и искусственные поверхности, созданные человеком: здания, сооружения, дороги с твердым покрытием и пр. (Что не удивительно: кирпич и бетон — основные строительные материалы.) Солончаки, а также лед и снег — практически белые или голубоватые (обычно — если покрыты водой). Песок и бетон тоже могут выглядеть очень светлыми, почти такими же белыми, как снег или соль. Но песок может иметь и розовый цвет.

Поэтому для отнесения объектов к той или иной категории недостаточно просто посмотреть их цвет на изображениях — необходимо проанализировать весь окружающий ландшафт, а также форму и текстуру объектов.

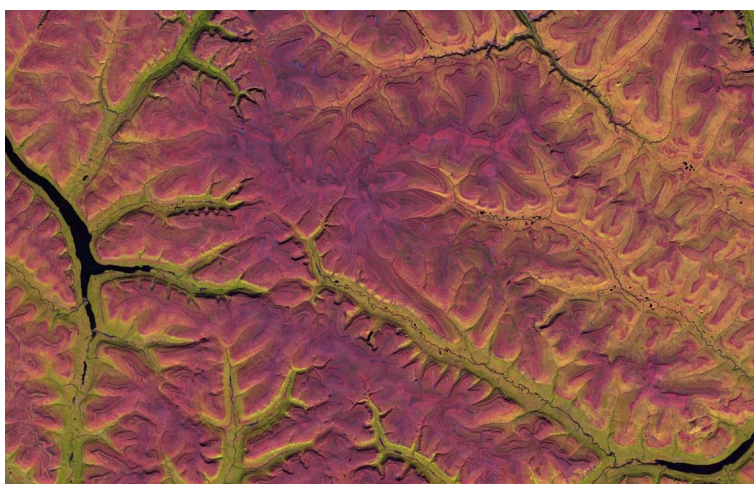
Изображение 1:



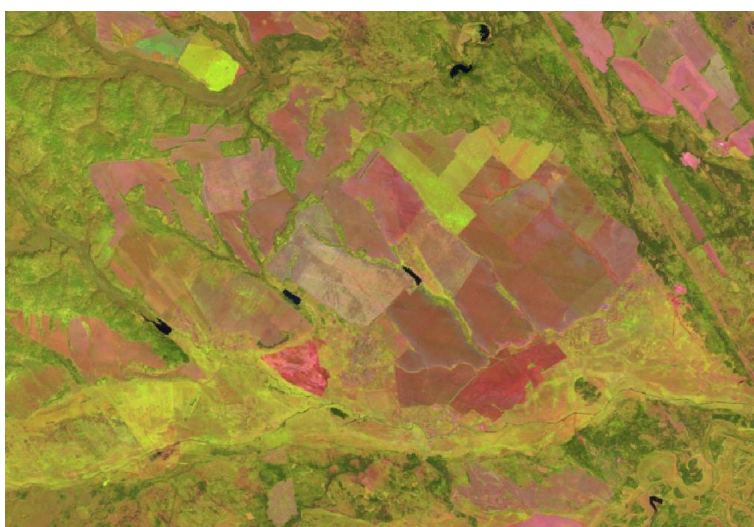
Изображение 2:



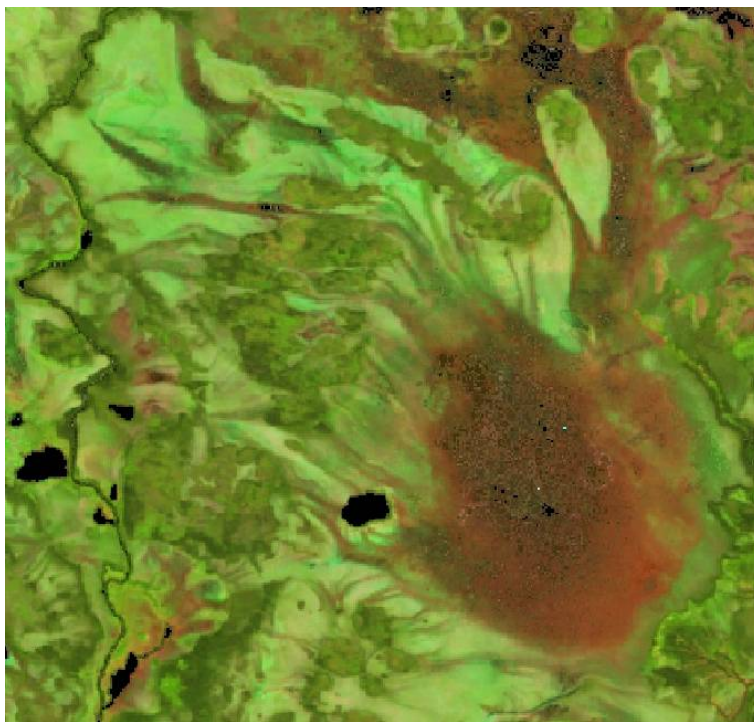
Изображение 3:



Изображение 4:



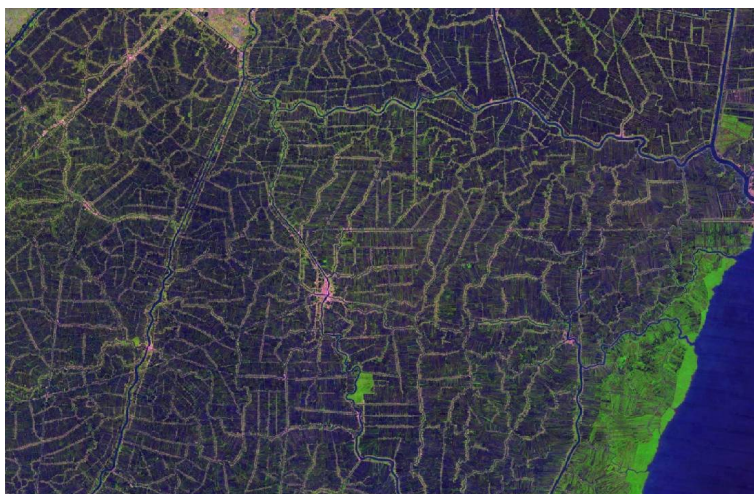
Изображение 5:



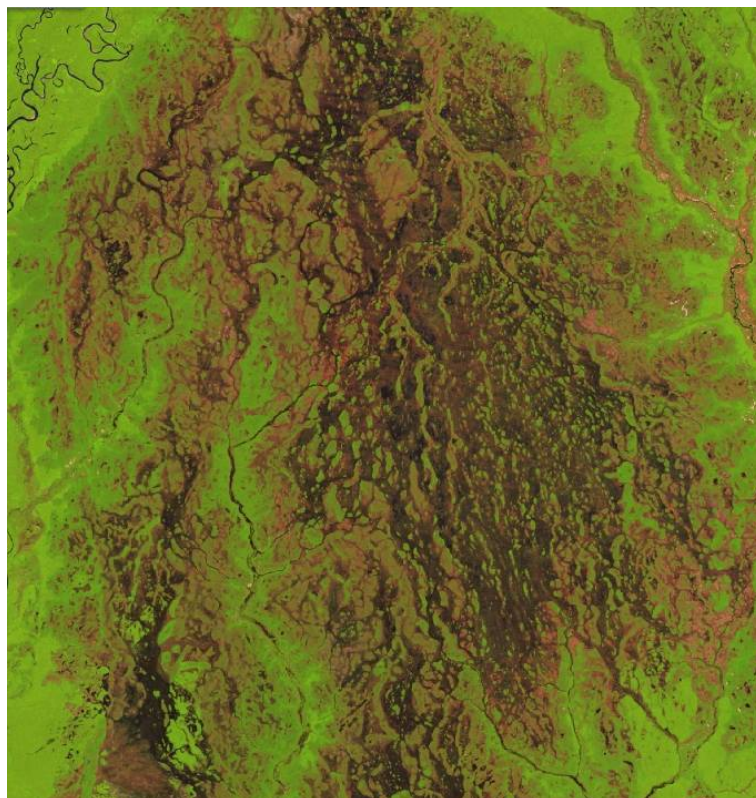
Изображение 6:



Изображение 7:



Изображение 8:



Изображение 9:



Отметьте верные ячейки

Спутниковые изображения	Природные болота	Горы	Сельхозугодья
Изображение 1	1	2	3
Изображение 2	4	5	6
Изображение 3	7	8	9
Изображение 4	10	11	12
Изображение 5	13	14	15
Изображение 6	16	17	18
Изображение 7	19	20	21
Изображение 8	22	23	24
Изображение 9	25	26	27

Ответ: 2; 4; 8; 12; 13; 18; 21; 22; 26.

Задача I.2.2.7. (10 баллов)

Потеря части лесного покрова происходит почти во всех странах мира, где есть хоть какие-то леса. Она может быть временной: через некоторое время древесный полог восстанавливается. Но может быть и практически необратимой — тогда говорят об обезлесивании (deforestation). Результаты спутникового мониторинга позволяют наблюдать потерю древесного полога (tree cover loss) в близком к реальному масштабу времени, а также получать ежегодную статистику по всему миру. С некоторыми данными можно ознакомиться, например, на портале Всемирной лесной вахты (<https://www.globalforestwatch.org/map>) (Global Forest Watch) или с помощью портала Глобальные изменения лесного покрова (<http://earthenginepartners.appspot.com/science-2013-global-forest>).

Лесной покров может исчезать и по естественным причинам. Например, в результате заболачивания территории или лесных пожаров, случающихся от молний. Но в подавляющем большинстве случаев наблюдаемые сегодня потери лесного покрова происходят в результате деятельности человека. (В том числе, по его вине происходит и подавляющее большинство лесных пожаров.)

Опираясь на данные портала Всемирной лесной вахты (Global Forest Watch), расположите страны из нижеприведенного списка по уменьшению доли (процента) потерь лесного покрова (tree cover loss) за 2019 год от общей площади всех сомкнутых лесов в пределах их границ, согласно данным наблюдения из космоса.

Для целей данной задачи учитывайте в качестве сомкнутых лесов территории с древесной растительностью (как естественного, так и искусственного происхождения) с сомкнутостью древесного полога (tree cover) не менее 20%. Используйте показатель всех суммарных потерь древесного полога за указанный период в процентах от его общей площади по состоянию на 2010 год, включая его временные потери, компенсируемые восстановлением древесной растительности.

Расположите элементы списка в правильном порядке

1. Беларусь;
2. Индонезия;
3. Мадагаскар;
4. Финляндия;

5. Австралия;
6. Лаос;
7. Малайзия;
8. Демократическая республика Конго;
9. Камбоджа;
10. Эстония.

Ответ: 5; 6; 9; 10; 7; 3; 4; 1; 2; 8.

Задача I.2.2.8. (10 баллов)

Вокруг точки с географическими координатами 63°44'53» северной широты и 39°49'12» восточной долготы в этом веке был почти полностью потерян лесной покров на площади более 8 тысяч гектаров.

Воспользуйтесь доступными в интернете источниками открытых космических снимков и других пространственных данных. Например, порталами Всемирной лесной вахты (<https://www.globalforestwatch.org/map>) (Global Forest Watch), Fires.RU (<https://fires.ru/>) российской компании Сканэкс, Worldview (<https://worldview.earthdata.nasa.gov/>) американского космического агентства НАСА и LandLook Viewer (<https://landlook.usgs.gov/viewer.html>) или GloVis (<https://glovis.usgs.gov/>) Геологической службы США (USGS). С их помощью определите год (четырёхзначное число), когда процесс или явление, в результате которого данная территория утратила свой лесной покров, имело место.

Помните, что использование только одного источника информации может привести Вас к неверным выводам (особенно, если данный источник опирается на результаты массовой автоматизированной обработки космических снимков по стандартным алгоритмам, которые могут не учитывать региональных особенностей). Постарайтесь проверить Ваши выводы, по крайней мере, по двум разным источникам.

Ответ: 2011.

Задача I.2.2.9. (10 баллов)

Определите страну по описанию. В поле ниже впишите ее полное название по-русски.

Согласно информации Всемирной базы данных по охраняемым природным территориям (<https://www.unep-wcmc.org/resources-and-data/wdpa>) (World Database on Protected Areas — WDPA) от Всемирного центра природоохранного мониторинга (<https://www.unep-wcmc.org/>) (World Conservation Monitoring Centre) Программы ООН по окружающей среде, на которую также ссылается Всемирный банк (<https://data.worldbank.org/>), особо охраняемые природные территории (ООПТ) занимают более 30% территории этой страны (по данным на 2018 год, морские охраняемые природные территории в территориальных водах страны не включены в эти расчеты). По данному показателю страна входит в пятерку лидеров на континенте. Правда, значительную часть этой площади составляют так называемые «территории охраняемого ландшафта», которые сложно признать природными резерватами со строгой охраной, — хозяйственная деятельность человека ограничена на них лишь

в некоторой степени. Общее количество ООПТ разного статуса и категорий в стране исчисляется десятками тысяч (включая совсем небольшие).

Десятки объектов на территории страны включены в список Всемирного наследия ЮНЕСКО (<https://whc.unesco.org/ru/list/>). Большинство из них является объектами культурного наследия. Лишь три природных объекта удостоились включения в список. И только один из них принадлежит исключительно данной стране — два других имеют трансграничный статус.

Согласно глобальной классификации (<https://www.worldwildlife.org/biome-categories/terrestrial-ecoregions>) наземных экорегионов Всемирного фонда дикой природы (WWF) (см. также таблицу и одну из карт), в пределах территории данной страны представлены пять экорегионов. Все они относятся к двум из 14 биомов, выделяемых согласно данной классификации. (На уровне биомов (<https://ru.wikipedia.org/wiki/%D0%91%D0%B8%D0%BE%D0%BC>) она достаточно близка к классификации природных зон, принятых в отечественной науке.) Оба этих биома и все пять экорегионов — лесные.

Согласно данным (<http://www.iea.org/stats/index.asp>) Международного энергетического агентства (International Energy Agency, IEA), на которые также ссылается Всемирный банк (<https://data.worldbank.org/>), с 2015 года данная страна получает более четверти производимого электричества из возобновляемых источников (и это без учета гидроэнергетики). И по доле, и по абсолютным значениям электроэнергии, вырабатываемой из возобновляемых источников (кроме гидроэлектростанций), страна входит в десятку мировых лидеров. Это больше электроэнергии, чем страна получает от сжигания нефти и газа, а также больше, чем она получает на атомных электростанциях.

Однако, выработка электроэнергии при сжигании угля в 2015 году все равно существенно превышало количество электроэнергии из возобновляемых источников. Всего при сжигании ископаемого топлива в 2015 году страна по-прежнему получала более половины своего электричества. В 2019 году, по данным (<http://www.iea.org/stats/index.asp>) Международного энергетического агентства, этот баланс уже был иной: выработка энергии из возобновляемых источников превысила выработку при сжигании угля, а общая доля электричества, получаемого при сжигании ископаемого топлива, опустилась ниже 50%.

При этом в общем энергетическом балансе страны импорт составляет более 60% от всей используемой энергии (рассчитанной в нефтяном эквиваленте).

Ответ: Германия.

Задача I.2.2.10. (10 баллов)

Наблюдение Земли из космоса (дистанционное зондирование) широко используется сегодня как инструмент изучения нашей планеты. Современные спутники дистанционного зондирования фиксируют интенсивность электромагнитного излучения, отраженного от поверхности Земли, не только в видимом свете, но и в тех частях спектра электромагнитных волн, которые не различимы человеческим глазом: в ультрафиолетовом, ближнем инфракрасном, средневолновом инфракрасном и дальнем инфракрасном. (Радарные спутники также фиксируют радиоволновое излучение.) Электромагнитные волны разной длины по-разному отражаются от поверх-

ности Земли и поэтому несут разную информацию о ней. Аппаратура современных спутников обычно измеряет и записывает яркость излучения в разных диапазонах длин волн (в разных спектральных каналах) независимо. Число таких каналов может быть различно. Как правило, их не меньше 3-4-х. А иногда — 10 и больше.

Используя три любых спектральных канала, можно рассматривать снимок как цветное изображение на компьютере. При этом зафиксированную аппаратурой спутника яркость электромагнитного излучения в одном канале представляют как яркость красного цвета, в другом — как яркость зеленого цвета, в третьем — как синего. Такие изображения Земли из космоса называют мультиспектральными (многоканальными). Если складывать мультиспектральное изображение из каналов в диапазоне видимого света и отображать яркость в сине-голубом диапазоне синим цветом, в желто-зеленом — зеленым цветом и в оранжево-красном диапазоне — красным, то такое изображение («в естественных цветах») будет похоже на обычную цифровую фотографию — как если бы сами фотографировали Землю со спутника цифровым фотоаппаратом (хотя, конечно, похоже не на 100%). Примерно в таком виде космические снимки размещаются на геопорталах общего пользования — таких как «Яндекс карты» и «Google карты».

Однако, можно отображать на компьютере синим, зеленым и красным цветами не обязательно яркости излучения в видимой части электромагнитного спектра — можно использовать яркости в тех диапазонах, которые человеческий глаз не различает — в ультрафиолетовом и инфракрасном. Часто это позволяет нам увидеть такую информацию о нашей планете, которую даже космонавты не могут увидеть с орбиты собственными глазами. Конечно, такие изображения не будут выглядеть как привычные нам фотографии — это будут изображения в условных цветах. Но часто, они несут больше информации о поверхности нашей планеты, чем изображения в видимой части спектра.

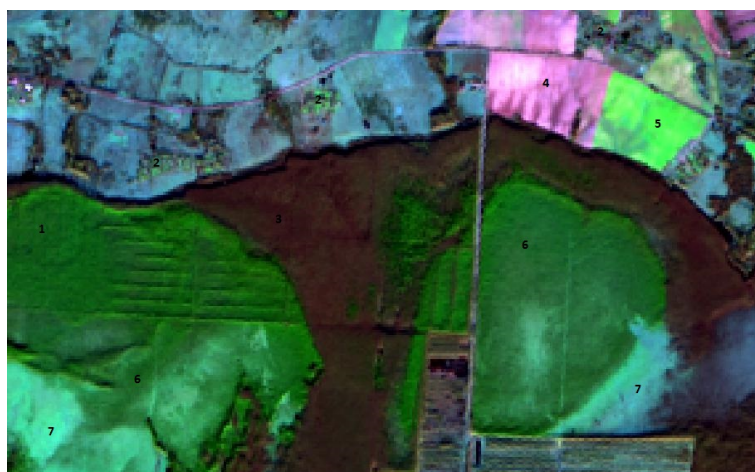
Рассмотрите приведенные ниже мультиспектральные изображения участка территории центра Европейской части России, сделанные спутником Landsat 8 22-го ноября 2019 года (глубокой осенью). Аппаратура этого спутника фиксирует яркость электромагнитного излучения в 10-ти разных спектральных каналах. Описание этих каналов можно найти, например, [здесь](#) или [здесь](#).

На изображениях цифрами обозначены различные объекты (однородные участки местности), список которых приведен ниже. Используя знания о географии нашей страны и информацию о спектральных каналах данного спутника, поставьте в соответствие каждому номеру участка его характеристику. Про использование различных комбинаций спектральных каналов спутников Landsat для изучения Земли можно прочесть, например, [здесь](#) и [здесь](#).

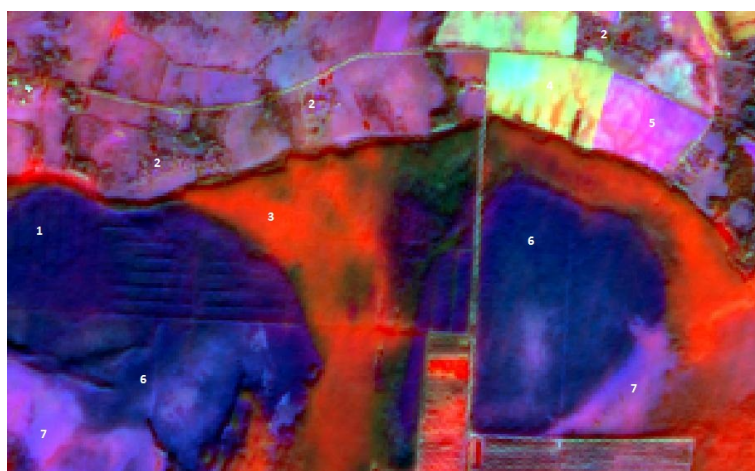
Изображение 1: красным цветом представлена интенсивность излучения в канале 4, зеленым — в канале 3, синим — в канале 2.



Изображение 2: красным цветом представлена интенсивность излучения в канале 6, зеленым — в канале 5, синим — в канале 4.



Изображение 3: красным цветом представлена интенсивность излучения в канале 10, зеленым — в канале 6, синим — в канале 5.



Сопоставьте значения из двух списков:

Участок 1	а. Хвойный лес по осушенному болоту
Участок 2	б. Лиственный лес
Участок 3	с. Распаханное и/или сжатое поле
Участок 4	д. Переходное болото с сосной и кустарничками
Участок 5	е. Населенный пункт (деревня)
Участок 6	ф. Открытое верховое сфагновое болото
Участок 7	г. Поле, засеянное озимыми

Ответ: 1 — а; 2 — е; 3 — б; 4 — с; 5 — г; 6 — д; 7 — ф.

Третья попытка. Задачи 9-11 класса

Привет, участник Олимпиады КД НТИ! Следующие 10 заданий по географии предназначены для учащихся 8-9 классов.

Время на решение ограничено временем открытия модуля (2 суток). После этого ответы на задачи приниматься не будут.

Если в процессе решения задач у тебя возникнет подозрение на ошибку в задании или какой-то существенный вопрос по условию, напиши авторам задачи через сообщения нашей группы vk.com/nticontest. В сообщении укажи ссылку на задачу.

Задача I.2.3.1. (10 баллов)

Загадочные геоглифы (рисунки и узоры на поверхности Земли) плато Наска в Перу всемирно известны и внесены в список Всемирного наследия ЮНЕСКО (<http://whc.unesco.org/en/list/700>). Размеры их столь велики, что они видны и на снимках Земли из космоса. Найдите и рассмотрите их с помощью геопортала Google Карты (<https://www.google.com/maps>), Bing Maps (<https://www.bing.com/maps>) или другого аналогичного сервиса.

Один из самых знаменитых рисунков известен как «кондор». С помощью геопорталов определите как можно точнее географические координаты самого кончика «клюва» этой нарисованной птицы. Ответ выразите в градусах, минутах и секундах. Секунды округлите до целого числа. Впишите ответ в поля ниже.

_____ 1 _____ 2 _____ 3 (северной/южной широты) _____ 4
 _____ 5 _____ 6 _____ 7 (восточной/западной долготы) _____ 8

Ответ: 1 — 14; 2 — 41; 3 — 52; 4 — южной широты; 5 — 75; 6 — 7; 7 — 34; 8 — западной долготы.

Задача I.2.3.2. (10 баллов)

Основателю Российской Империи Петру Первому поставлено в нашей стране множество памятников. С помощью геопортала Яндекс Карты (<https://maps.yandex.ru/>), сервиса «Мои места» геопортала Google Карты (<https://www.google.com/maps>) или программы Google Планета Земля Pro (<https://www.google.com/intl/ru/earth/>)

[desktop/](#)) измерьте как можно точнее расстояние по поверхности Земли между двумя, наверное, самыми известными из них: между Медным всадником в Санкт-Петербурге и памятником «В ознаменование 300-летия российского флота» на Москве-реке работы Зураба Церетели. Ответ выразите в километрах и округлите до целых единиц.

Ответ: $636,5 \pm 0.5$.

Задача I.2.3.3. (10 баллов)

С помощью Конструктора карт (<https://yandex.ru/map-constructor>) геопортала Яндекс Карты (<https://maps.yandex.ru/>), сервиса «Мои места» геопортала Google Карты (<https://www.google.com/maps>), портала Kosmosnimki.RU (<https://kosmosnimki.ru/>), другого аналогичного сервиса или программы Google Планета Земля Pro (<https://www.google.com/intl/ru/earth/desktop/>) определите площадь беломорского острова Моржовец.

Полученное значение площади представьте в квадратных километрах, округлите до целых единиц и введите в поле ниже.

Ответ: 79 ± 1.5 .

Задача I.2.3.4. (10 баллов)

Точка с географическими координатами $04^{\circ}13'00''$ южной широты и $105^{\circ}47'37''$ восточной долготы расположена в центральной части большой территории специфического внешнего вида площадью около 100 квадратных километров.

С помощью геопорталов Яндекс Карты (<https://maps.yandex.ru/>), Kosmosnimki.RU (<https://kosmosnimki.ru/>)U, геопортала Google Карты (<https://www.google.com/maps>), программы Google Планета Земля Pro (<https://www.google.com/intl/ru/earth/desktop/>) или других аналогичных инструментов рассмотрите космические снимки на данную территорию.

Что из себя представляет данная территория? В результате какого природного процесса или воздействия человека она приобрела такой вид? Выберите только один вариант из представленных ниже.

Выберите один вариант из списка

1. добыча рассыпного золота;
2. территория, занятая солнечными батареями (солнечная электростанция);
3. пастбища и загоны для скота;
4. зона отдыха / туризма с бассейнами;
5. очистные сооружения крупного населенного пункта;
6. плантации эвкалиптов и/или других быстрорастущих деревьев;
7. плантации масличной пальмы;
8. рисовые чеки;
9. марикультура: выращивание рыбы / морских беспозвоночных;
10. фруктовые сады.

Ответ: 9.

Задача I.2.3.5. (10 баллов)

На приведенных ниже спутниковых изображениях Вы видите различные нагорья Евразии. Все они достаточно крупные и общеизвестные. Введите их названия по-русски (в общепринятой транскрипции) в соответствующие поля ниже. Если в пределы снимка попадают части других нагорий, введите название того, которое занимает самую большую площадь в пределах снимка.

Эти изображения земной поверхности получены со спутников Landsat 7 и Landsat 8. Они не являются единовременным снимком, а получены с помощью синтеза ряда безоблачных снимков за 2019 год, обработанных таким образом, чтобы компенсировать всегда имеющиеся различия в положении солнца, положении спутника, прозрачности атмосферы и пр. Обработка проведена лабораторией GLAD (Global Land Analysis & Discovery — <https://glad.umd.edu/>) Географического факультета Университета Мэриленда (США). Данное изображение сделано с помощью портала Глобальные изменения лесного покрова (<http://earthenginepartners.appspot.com/science-2013-global-forest>).

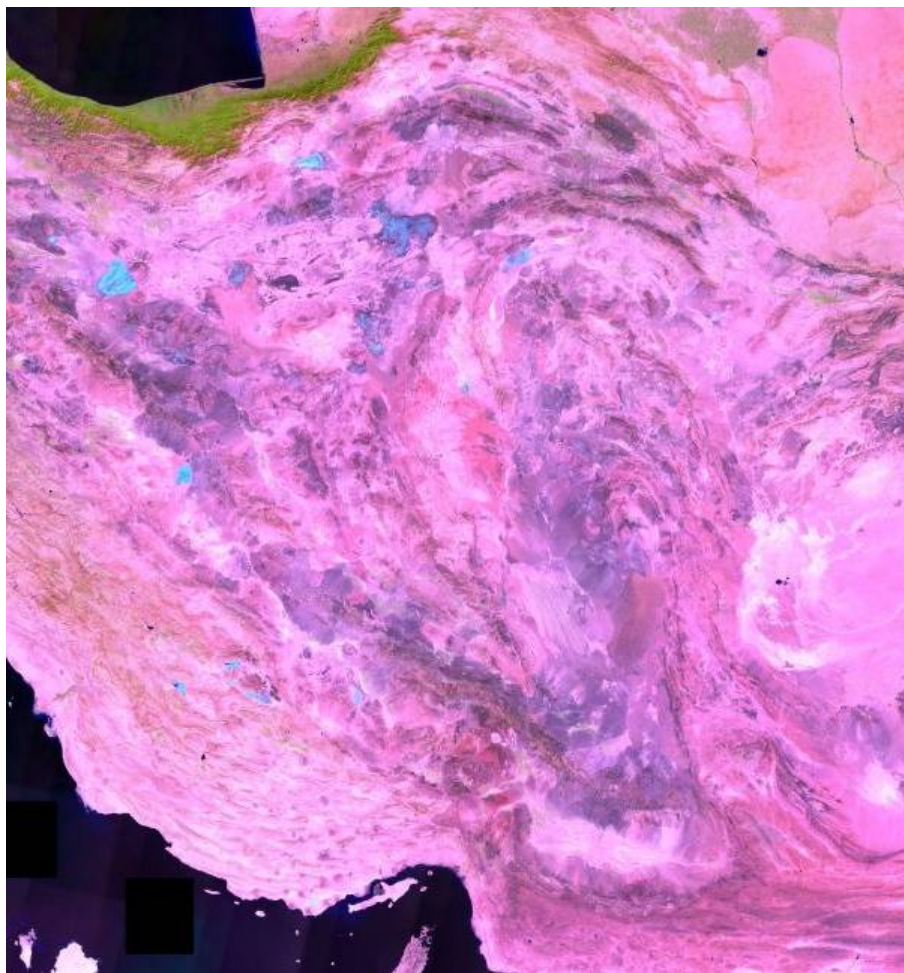
Изображение представляет интенсивность электромагнитного излучения, отраженного от поверхности земли и принятого аппаратурой спутников, в условных цветах: не видимый человеческим глазом коротковолновый инфракрасный свет представлен красным, ближний инфракрасный (также не различимый глазом) — зеленым, видимый красный свет — синим. В таком цветовом синтезе хорошо видны различия в растительности и влажности поверхности.

При этом чистые воды водоемов и водотоков выглядят темными, почти черными, так как равномерно поглощают большую часть солнечных лучей во всех указанных участках спектра. Если в воде имеется взвесь минеральных частиц или высокая плотность планктонных организмов, такие акватории выглядят как оттенки синего и голубого тонов. Такой же цвет могут иметь и мелководные участки, где сквозь толщу воды проходит солнечный свет, отраженный от дна водоема и/или от водной растительности.

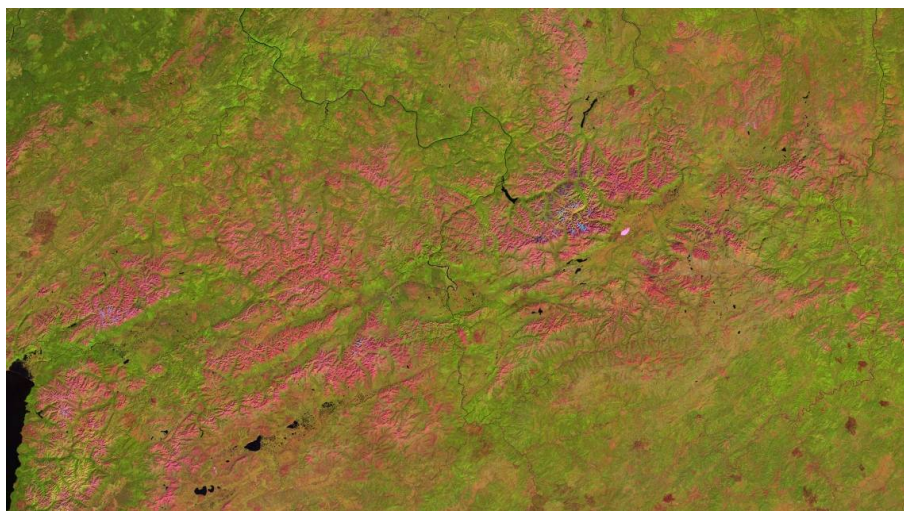
Участки суши, покрытые сомкнутой растительностью, как правило, выглядят при данном цветовом синтезе как различные оттенки зеленого. При этом, более темные и насыщенные тона, как правило, соответствуют древесной растительности, а более светлые — травянистой и кустарниковой.

Открытый грунт, в зависимости от его состава и характера, отображается различными оттенками красного и фиолетового. Частично покрытые растительностью участки (имеющие несомкнутый растительный покров) могут иметь различные промежуточные оттенки (розоватые, бурые и пр.). Голые скалы и песок, как правило, имеют различные оттенки голубого и синего. Близкие оттенки имеют и искусственные поверхности, созданные человеком: здания, сооружения, дороги с твердым покрытием и пр. (Что не удивительно: кирпич и бетон — основные строительные материалы.) Солончаки, а также лед и снег — практически белые или голубоватые (обычно — если покрыты водой). Песок тоже может выглядеть очень светлыми, почти такими же белыми, как снег или соль. Но песок может иметь и розовый цвет.

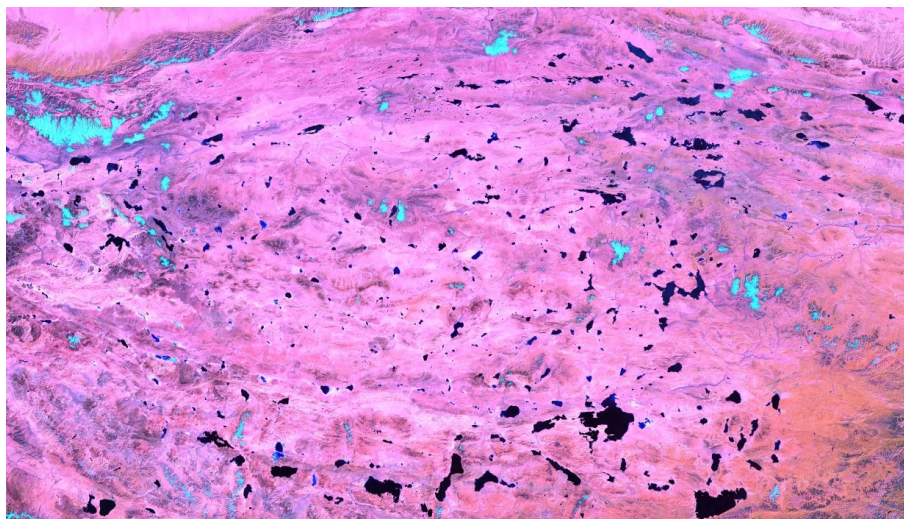
Нагорье 1:



Нагорье 2:



Нагорье 3:



Нагорье на космическом снимке 1: _____¹.

Нагорье на космическом снимке 2: _____².

Нагорье на космическом снимке 3: _____³.

Ответ: 1 — Иранское нагорье; 2 — Становое нагорье; 3 — Тибетское нагорье.

Задача I.2.3.6. (10 баллов)

Рассмотрите внимательно приведенные ниже спутниковые изображения различных природных объектов. Определите, к какой из трех категорий относятся следы хозяйственной деятельности человека / антропогенные ландшафты на каждом изображении: (1) рубки леса с целью заготовки древесины, (2) сельхозугодья или (3) городская застройка. Поставьте в таблице ниже галочку в соответствующей колонке для каждого изображения. В каждой строке таблицы должна стоять только одна галочка.

Если на некоторых изображениях присутствуют, по Вашему мнению, два или все три типа антропогенных ландшафтов / объектов, выберите занимающий большую площадь в пределах карты и расположенный вблизи центра изображения.

Эти изображения земной поверхности получены со спутников Landsat 5, Landsat 7 и Landsat 8. Они не являются единовременным снимком, а получены с помощью синтеза ряда безоблачных снимков за год (использовались снимки 2000 и 2019 годов), обработанных таким образом, чтобы компенсировать всегда имеющиеся различия в положении солнца, положении спутника, прозрачности атмосферы и пр. Обработка проведена лабораторией GLAD (Global Land Analysis & Discovery — <https://glad.umd.edu/>) Географического факультета Университета Мэриленда (США). Данное изображение сделано с помощью портала Глобальные изменения лесного покрова (<http://earthenginepartners.appspot.com/science-2013-global-forest>).

Изображение представляет интенсивность электромагнитного излучения, отраженного от поверхности земли и принятого аппаратурой спутников, в условных цветах: не видимый человеческим глазом коротковолновый инфракрасный свет представлен красным, ближний инфракрасный (также не различимый глазом) — зеленым, видимый красный свет — синим. В таком цветовом синтезе хорошо видны различия в растительности и влажности поверхности.

При этом чистые воды водоемов и водотоков выглядят темными, почти черными, так как равномерно поглощают большую часть солнечных лучей во всех указанных участках спектра. Если в воде имеется взвесь минеральных частиц или высокая плотность планктонных организмов, такие акватории выглядят как оттенки синего и голубого тонов. Такой же цвет могут иметь и мелководные участки, где сквозь толщу воды проходит солнечный свет, отраженный от дна водоема и/или от водной растительности.

Участки суши, покрытые сомкнутой растительностью, как правило, выглядят при данном цветовом синтезе как различные оттенки зеленого. При этом, более темные и насыщенные тона, как правило, соответствуют древесной растительности, а более светлые — травянистой и кустарниковой.

Открытый грунт, в зависимости от его состава и характера, отображается различными оттенками красного и фиолетового. Частично покрытые растительностью участки (имеющие несомкнутый растительный покров) могут иметь различные промежуточные оттенки (розоватые, бурые и пр.). Красноватые тона может иметь и болотная растительность. Голые скалы и песок, как правило, имеют различные оттенки голубого и синего. Близкие оттенки имеют и искусственные поверхности, созданные человеком: здания, сооружения, дороги с твердым покрытием и пр. (Что не удивительно: кирпич и бетон — основные строительные материалы.) Солончаки, а также лед и снег — практически белые или голубоватые (обычно — если покрыты водой). Песок и бетон тоже могут выглядеть очень светлыми, почти такими же белыми, как снег или соль. Но песок может иметь и розовый цвет.

Поэтому для отнесения ландшафтов / объектов к той или иной категории недостаточно просто посмотреть их цвет на изображениях — необходимо проанализировать весь окружающий ландшафт, а также форму и текстуру антропогенных объектов.

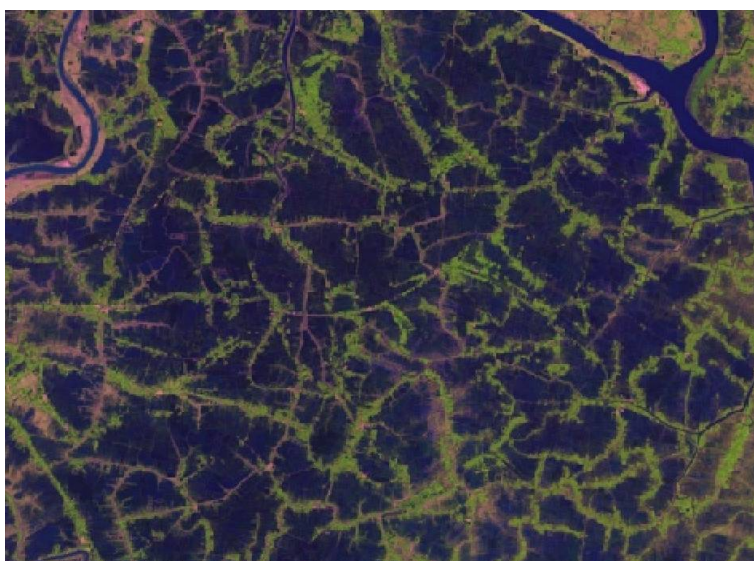
Изображение 1:



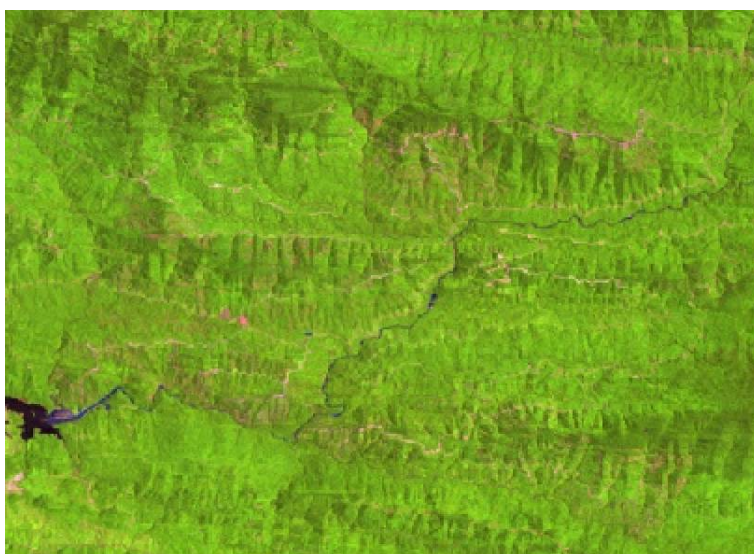
Изображение 2:



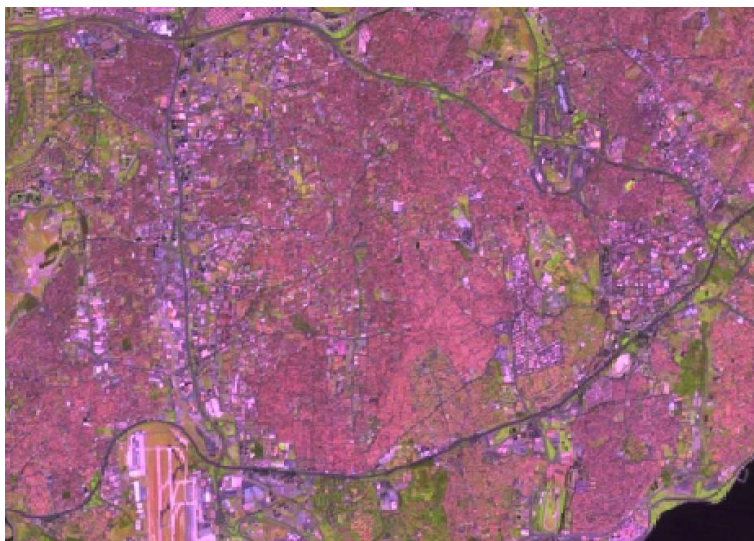
Изображение 3:



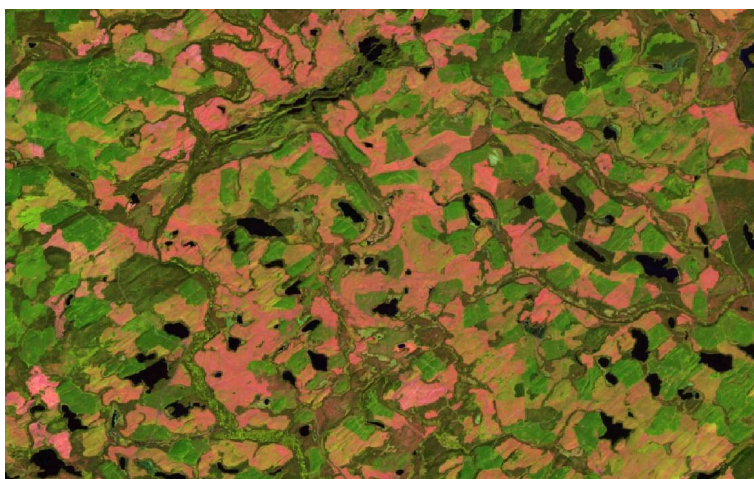
Изображение 4:



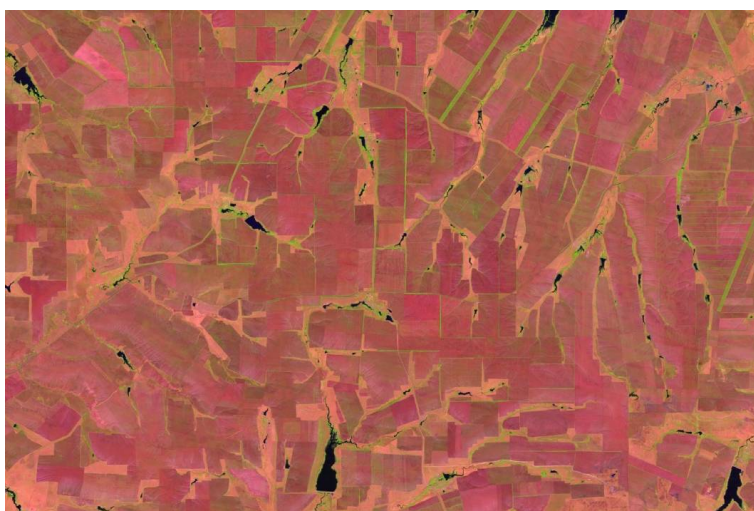
Изображение 5:



Изображение 6:



Изображение 7:



Изображение 8:



Изображение 9:



Отметьте верные ячейки

Спутниковые изображения	Лесозаготовки	Сельхозугодья	Городская застройка
Изображение 1	_____ 1	_____ 2	_____ 3
Изображение 2	_____ 4	_____ 5	_____ 6
Изображение 3	_____ 7	_____ 8	_____ 9
Изображение 4	_____ 10	_____ 11	_____ 12
Изображение 5	_____ 13	_____ 14	_____ 15
Изображение 6	_____ 16	_____ 17	_____ 18
Изображение 7	_____ 19	_____ 20	_____ 21
Изображение 8	_____ 22	_____ 23	_____ 24
Изображение 9	_____ 25	_____ 26	_____ 27

Ответ: 2; 6; 8; 10; 15; 16; 20; 23; 25.

Задача I.2.3.7. (10 баллов)

Количество и расположение поверхностных вод (как материковых, так и прибрежных) — важнейшая характеристика любой территории. От них во многом зави-

сит локальный климат, биологическое разнообразие и благополучие людей. В свою очередь поверхностные воды сами находятся под влиянием климата и хозяйственной деятельности человека. Некоторые территории, ранее покрытые водой, теряют свою водную поверхность в результате пересыхания или изменений береговой линии. Другие, напротив, из участков суши превращаются в акватории. Многие территории заливаются водой только в определенный сезон. Сегодня Вы можете познакомиться с результатами спутникового мониторинга поверхностных вод с помощью общедоступных онлайн-инструментов, например, Global Surface Water Explorer (<https://global-surface-water.appspot.com/>) и Freshwater Ecosystems Explorer (<https://www.sdg661.app/>). (Последний, вопреки названию, содержит также информацию о соленых озерах и прибрежных морских акваториях.)

Опираясь на данные этих порталов, рассчитайте для ряда стран площади, которые были не покрыты водой при минимальном уровне наполнения водохранилищ страны в 2019 году, но которые были покрыты водой при их максимальном уровне, наблюдавшемся в тот же год. Какую долю (процент) эти площади составляют от максимальной площади водохранилищ каждой страны в 2019 году? Расположите страны из нижеприведенного списка по уменьшению этой относительной величины.

Расположите элементы списка в правильном порядке

1. Финляндия.
2. США.
3. Индия.
4. Бангладеш.
5. Бразилия.
6. Узбекистан.
7. Эфиопия.
8. Нигерия.
9. Германия.
10. Китай.

Ответ: 3; 4; 6; 7; 8; 10; 5; 9; 2; 1.

Задача I.2.3.8. (10 баллов)

Участок земной поверхности вокруг точки с географическими координатами 58°33'52»северной широты и 100°35'30» восточной долготы круглогодично является акваторией (по состоянию на 2019 год). Однако, ранее этот участок был сушей.

Воспользуйтесь порталом Global Surface Water Explorer (<https://global-surface-water.appspot.com/>) и/или порталами, которые позволяют рассматривать онлайн космические снимки за разные периоды времени (например, LandLook Viewer (<https://landlook.usgs.gov/viewer.html>) или GloVis (<https://glovis.usgs.gov/>) Геологической службы США, USGS, программой Google Планета Земля Pro (<https://www.google.com/intl/ru/earth/desktop/>)). Определите год (четырёхзначное число), когда участок вблизи указанной точки впервые вновь был покрыт водой в течение всего года. Лучше проверить Ваши выводы, по крайней мере, по двум разным источникам.

Ответ: 2014.

Задача I.2.3.9. (10 баллов)

Определите страну по описанию. В поле ниже впишите ее полное название по-русски.

Согласно данным Продовольственной и сельскохозяйственной организации Объединенных наций (UN Food and Agriculture Organization) на 2016 год, приведенным на сайте Всемирного банка (<https://data.worldbank.org/>), сельскохозяйственные земли занимают менее 4% территории этой страны, хотя в сельской местности, по данным из того же сайта, проживает более половины ее населения. Суммарный коэффициент рождаемости по данным на 2018 год превышает 3,5%. Довольно высокий уровень преступности.

Согласно данным портала Всемирной лесной вахты <https://www.globalforestwatch.org/> (Global Forest Watch), древесная растительность (с сомкнутостью древесного полога 30% и выше) занимала, по состоянию и на 2000, и на 2010 год, более 90% территории этой страны. По этому показателю страна стабильно входит в десятку мировых лидеров. За весь период с 2001 по 2019 годы потери лесного покрова в стране составили первые проценты от общей площади лесов. Ведущей причиной потери лесного покрова — примитивное подсечно-огневое земледелие.

Из 14 биомов (<https://ru.wikipedia.org/wiki/%D0%91%D0%B8%D0%BE%D0%BC>), выделяемых Всемирным фондом дикой природы (WWF) в рамках глобальной классификации наземных экорегионов (<https://www.worldwildlife.org/biome-categories/terrestrial-ecoregions>) (см. также таблицу ([https://en.wikipedia.org/wiki/List_of_terrestrial_ecoregions_\(WWF\)](https://en.wikipedia.org/wiki/List_of_terrestrial_ecoregions_(WWF)))) и одну из карт (<http://ecoregions2017.appspot.com/>)), в пределах территории данной страны представлены четыре. При этом один из них занимает более трех ее четвертей. Во флоре и фауне страны представлено много эндемичных видов. В стране обитают, в том числе, эндемичные представители примитивных отрядов млекопитающих.

Ответ: Папуа — Новая Гвинея.

Задача I.2.3.10. (10 баллов)

Наблюдение Земли из космоса (дистанционное зондирование) широко используется сегодня как инструмент изучения нашей планеты. Современные спутники дистанционного зондирования фиксируют интенсивность электромагнитного излучения, отраженного от поверхности Земли, не только в видимом свете, но и в тех частях спектра электромагнитных волн, которые не различимы человеческим глазом: в ультрафиолетовом, ближнем инфракрасном, средневолновом инфракрасном и дальнем инфракрасном. (Радарные спутники также фиксируют радиоволновое излучение.) Электромагнитные волны разной длины по-разному отражаются от поверхности Земли и поэтому несут разную информацию о ней. Аппаратура современных спутников обычно измеряет и записывает яркость излучения в разных диапазонах длин волн (в разных спектральных каналах) независимо. Число таких каналов может быть различно. Как правило, их не меньше 3-4-х. А иногда — 10 и больше.

Используя три любых спектральных канала, можно изучать снимок как цвет-

ное изображение на компьютере. При этом зафиксированную аппаратурой спутника яркость электромагнитного излучения в одном канале представляют как яркость красного цвета, в другом — как яркость зеленого цвета, в третьем — как синего. Такие изображения Земли из космоса называют мультиспектральными (многоканальными). Если складывать мультиспектральное изображение из каналов в диапазоне видимого света и отображать яркость в сине-голубом диапазоне синим цветом, в желто-зеленом — зеленым цветом и в красном диапазоне — красным, то такое изображение («в естественных цветах») будет похоже на обычную цифровую фотографию — как если бы сами фотографировали Землю со спутника цифровым фотоаппаратом. Примерно в таком виде космические снимки размещаются на геопорталах общего пользования — таких как «Яндекс карты» и «Google карты».

Однако, можно отображать на компьютере синим, зеленым и красным цветами не обязательно яркости излучения в видимой части электромагнитного спектра — можно использовать яркости в тех диапазонах, которые человеческий глаз не различает — в ультрафиолетовом и инфракрасном. Часто это позволяет нам увидеть такую информацию о нашей планете, которую даже космонавты не могут увидеть с орбиты собственными глазами. Конечно, такие изображения не будут выглядеть как привычные нам фотографии — это будут изображения в условных цветах. Но часто, они несут больше информации о поверхности нашей планеты, чем изображения в видимой части спектра.

Рассмотрите приведенные ниже мультиспектральные изображения участка территории Самарской области, сделанные спутником Landsat 8 13-го сентября 2019 года. Аппаратура этого спутника фиксирует яркость электромагнитного излучения в 10-ти разных спектральных каналах. Описание этих каналов можно найти, например, [здесь](#) или [здесь](#).

На изображениях цифрами обозначены различные объекты (однородные участки местности), список которых приведен ниже. Используя знания о географии нашей страны и информацию о спектральных каналах данного спутника и времени года, поставьте в соответствие каждому номеру участка его характеристику. Про использование различных комбинаций спектральных каналов спутников Landsat для изучения Земли можно прочесть, например, [здесь](#) и [здесь](#).

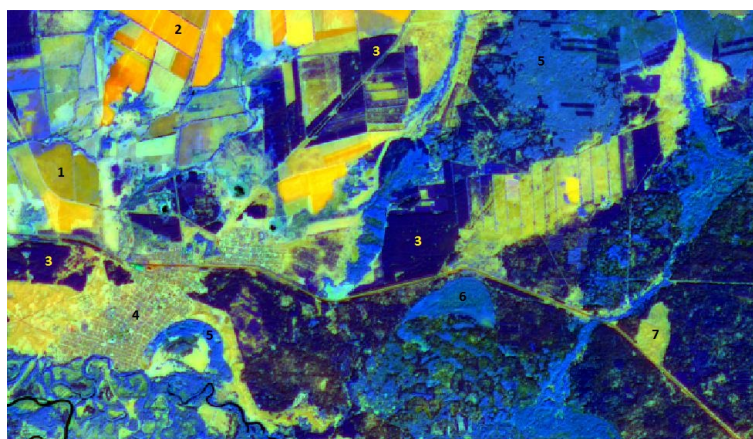
Изображение 1: красным цветом представлена интенсивность излучения в канале 4, зеленым — в канале 3, синим — в канале 2.



Изображение 2: красным цветом представлена интенсивность излучения в канале 6, зеленым — в канале 5, синим — в канале 4.



Изображение 3: красным цветом представлена интенсивность излучения в канале 11, зеленым — в канале 6, синим — в канале 5.



Сопоставьте значения из двух списков:

Участок 1	а. Вырубка (скорее всего, санитарная)
Участок 2	б. Сжатое поле
Участок 3	с. Лиственный лес (возможно, с небольшой примесью хвойных)
Участок 4	д. Болото, поросшее лиственными деревьями
Участок 5	е. Распаханное поле
Участок 6	ф. Хвойный лес (искусственные насаждения)
Участок 7	г. Населенный пункт (деревня)

Ответ: 1 — б; 2 — ; 3 — ф; 4 — г; 5 — с; 6 — д; 7 -а.

Четвертая попытка. Задачи 9-11 класса

Привет, участник Олимпиады КД НТИ! Следующие 10 заданий по географии предназначены для учащихся 8-9 классов.

Время на решение ограничено временем открытия модуля (2 суток). После этого ответы на задачи приниматься не будут.

Если в процессе решения задач у тебя возникнет подозрение на ошибку в задании или какой-то существенный вопрос по условию, напиши авторам задачи через сообщения нашей группы vk.com/nticontest. В сообщении укажи ссылку на задачу.

Задача I.2.4.1. (10 баллов)

С помощью общедоступных геопорталов Яндекс Карты (<https://maps.yandex.ru/>), Google Maps (<https://www.google.com/maps>), Bing Maps (<https://www.bing.com/maps>), а также программы Google Планета Земля Pro определите как можно точнее географические координаты самой южной точки второго по величине континента Земли.

Вследствие приливов и отливов в океане де факто самой южной точкой суши этого материка при разном уровне воды временно оказываются разные участки литорали (приливно-отливной зоны) и прибрежных скал (которые в прилив могут не соединяться с материком). Внимательно изучите космические снимки высокого разрешения, доступные на всех, перечисленных выше ресурсах. В программе Google Планета Земля Pro (<https://www.google.com/intl/ru/earth/desktop/>) изучите доступные снимки высокого разрешения за последние 10 лет. Выберите наиболее южную точку при самом низком уровне воды (при максимальном отливе) из тех, которые видны на доступных космических снимках высокого разрешения.

Ответ выразите в градусах, минутах и секундах. Секунды округлите до целого числа. Впишите ответ в поля ниже.

_____ 1 _____ 2 _____ 3 (северной/южной широты) _____ 4
 _____ 5 _____ 6 _____ 7 (восточной/западной долготы) _____ 8

Ответ: 1 — 34; 2 — 50; 3 — 3 или 4; 4 — южной широты; 5 — 19; 6 — 59; 7 — 25 ± 1 ; 8 — восточной долготы.

Задача I.2.4.2. (10 баллов)

С помощью геопортала Яндекс Карты (<https://maps.yandex.ru/>), сервиса «Мои места» геопортала Google Карты (<https://www.google.com/maps>) или программы Google Планета Земля Pro (<https://www.google.com/intl/ru/earth/desktop/>) измерьте как можно точнее расстояние по поверхности Земли между Останкинской и Шуховской телебашнями в городе Москве. Ответ выразите в километрах и округлите до десятых долей.

Ответ: 11.4.

Задача I.2.4.3. (10 баллов)

С помощью Конструктора карт (<https://yandex.ru/map-constructor>) геопортала Яндекс Карты (<https://maps.yandex.ru/>), сервиса «Мои места» геопортала Google Карты (<https://www.google.com/maps>), портала Kosmosnimki.RU (<https://kosmosnimki.ru/>), другого аналогичного сервиса или программы Google Планета Земля Pro (<https://www.google.com/intl/ru/earth/desktop/>) определите площадь

пятого по размеру острова в составе Шантарского архипелага (Охотское море).

Полученное значение площади представьте в квадратных километрах, округлите до целых единиц и введите в поле ниже.

Ответ: 29.

Задача I.2.4.4. (10 баллов)

Точка с географическими координатами 64°01'40» северной широты и 122°32'20» восточной долготы расположена в пределах участка земной поверхности характерного внешнего вида площадью более 5 квадратных километров. Общая площадь аналогичных участков в данном районе составляет десятки квадратных километров.

С помощью геопорталов Яндекс Карты(<https://maps.yandex.ru/>), Kosmosnimki.RU (<https://kosmosnimki.ru/>), геопортала Google Карты(<https://www.google.com/maps>), программы Google Планета Земля Pro (<https://www.google.com/intl/ru/earth/desktop/>) или других аналогичных инструментов рассмотрите космические снимки на данную территорию.

Что из себя представляет данная территория? В результате какого природного процесса или воздействия человека она приобрела такой вид? Выберите только один вариант из представленных ниже.

Выберите один вариант из списка

1. Гарь после недавнего лесного пожара.
2. Участки выхода на поверхность вечной мерзлоты.
3. Озера, покрытые льдом.
4. Земельные участки, покрытые снегом.
5. Участки выхода на поверхность меловых отложений.
6. Участки выхода на поверхность известняков.
7. Массив песчаных дюн.
8. Солончак.
9. Заброшенный карьер после добычи полезных ископаемых.
10. Ледник.

Ответ: 7.

Задача I.2.4.5. (10 баллов)

На приведенных ниже спутниковых изображениях Вы видите различные участки территории нашей страны, где расположены государственные природные заповедники (с прилегающими территориями). Введите названия этих заповедников по-русски в соответствующие поля ниже. Введите только название в одно слово — не вводите ни самого слова «заповедник», ни слов «государственный», «природный», «биосферный» и пр., а также не ставьте название заповедника в кавычки.

Эти изображения земной поверхности получены со спутников Landsat 7 и Landsat 8. Они не являются единовременным снимком, а получены с помощью синтеза ря-

да безоблачных снимков за 2019 год, обработанных таким образом, чтобы компенсировать всегда имеющиеся различия в положении солнца, положении спутника, прозрачности атмосферы и пр. Обработка проведена лабораторией GLAD (Global Land Analysis & Discovery — <https://glad.umd.edu/>) Географического факультета Университета Мэриленда (США). Данное изображение сделано с помощью портала Глобальные изменения лесного покрова (<http://earthenginepartners.appspot.com/science-2013-global-forest>).

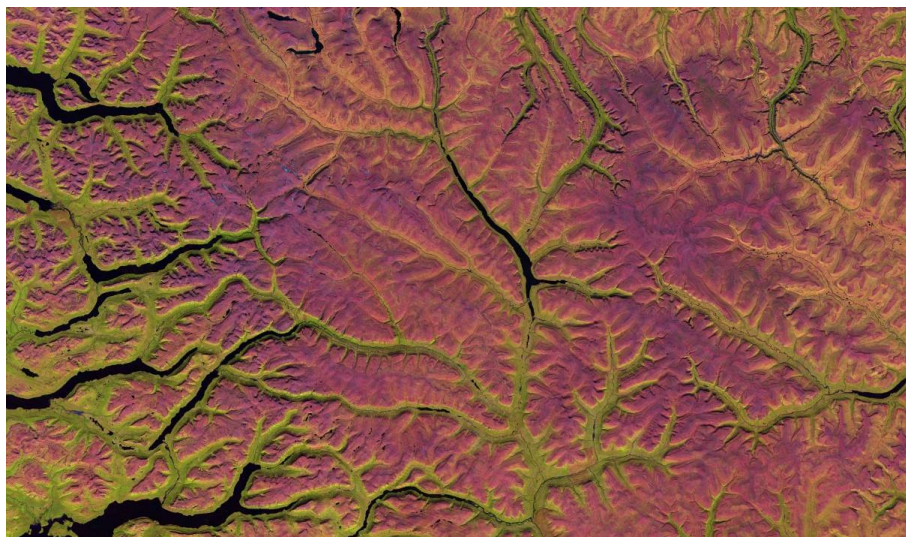
Изображение представляет интенсивность электромагнитного излучения, отраженного от поверхности земли и принятого аппаратурой спутников, в условных цветах: не видимый человеческим глазом коротковолновый инфракрасный свет представлен красным, ближний инфракрасный (также не различимый глазом) — зеленым, видимый красный свет — синим. В таком цветовом синтезе хорошо видны различия в растительности и влажности поверхности.

При этом чистые воды водоемов и водотоков выглядят темными, почти черными, так как равномерно поглощают большую часть солнечных лучей во всех указанных участках спектра. Если в воде имеется взвесь минеральных частиц или высокая плотность планктонных организмов, такие акватории выглядят как оттенки синего и голубого тонов. Такой же цвет могут иметь и мелководные участки, где сквозь толщу воды проходит солнечный свет, отраженный от дна водоема и/или от водной растительности.

Участки суши, покрытые сомкнутой растительностью, как правило, выглядят при данном цветовом синтезе как различные оттенки зеленого. При этом, более темные и насыщенные тона, как правило, соответствуют древесной растительности, а более светлые — травянистой и кустарниковой.

Открытый грунт, в зависимости от его состава и характера, отображается различными оттенками красного и фиолетового. Частично покрытые растительностью участки (имеющие несомкнутый растительный покров) могут иметь различные промежуточные оттенки (розоватые, бурые и пр.). Голые скалы и песок, как правило, имеют различные оттенки голубого и синего. Близкие оттенки имеют и искусственные поверхности, созданные человеком: здания, сооружения, дороги с твердым покрытием и пр. (Что не удивительно: кирпич и бетон — основные строительные материалы.) Солончаки, а также лед и снег — практически белые или голубоватые (обычно — если покрыты водой). Песок тоже может выглядеть очень светлыми, почти такими же белыми, как снег или соль. Но песок может иметь и розовый цвет.

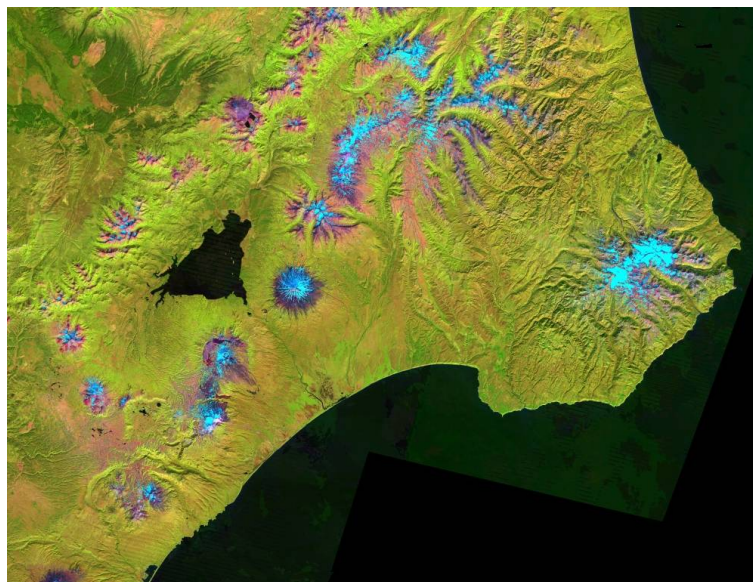
Заповедник 1:



Заповедник 2:



Заповедник 3:



Заповедник на космическом снимке 1: _____¹.

Заповедник на космическом снимке 2: _____².

Заповедник на космическом снимке 3: _____³.

Ответ: 1 — Путоранский; 2 — Костомукшский; 3 — Кроноцкий.

Задача I.2.4.6. (10 баллов)

Рассмотрите внимательно приведенные ниже спутниковые изображения различных природных или антропогенных объектов. Определите, к какой из трех категорий относятся объекты на каждом изображении: (1) вулканы, (2) места добычи каменного угля или (3) места добычи иных полезных ископаемых. Поставьте в таблице ниже галочку в соответствующей колонке для каждого изображения. В каждой строке таблицы должна стоять только одна галочка.

Если на некоторых изображениях присутствуют, по Вашему мнению, два или все три типа объектов, выберите занимающий большую площадь в пределах карты и расположенный вблизи центра изображения.

Эти изображения земной поверхности получены со спутников Landsat 5, Landsat 7 и Landsat 8. Они не являются единовременным снимком, а получены с помощью синтеза ряда безоблачных снимков за год (использовались снимки 2019 года), обработанных таким образом, чтобы компенсировать всегда имеющиеся различия в положении солнца, положении спутника, прозрачности атмосферы и пр. Обработка проведена лабораторией GLAD (Global Land Analysis & Discovery — <https://glad.umd.edu/>) Географического факультета Университета Мэриленда (США). Данное изображение сделано с помощью портала Глобальные изменения лесного покрова (<http://earthenginepartners.appspot.com/science-2013-global-forest>).

Изображение представляет интенсивность электромагнитного излучения, отраженного от поверхности Земли и принятого аппаратурой спутников, в условных цветах: не видимый человеческим глазом коротковолновый инфракрасный свет представлен красным, ближний инфракрасный (также не различимый глазом) — зеленым, видимый красный свет — синим. В таком цветовом синтезе хорошо видны раз-

личия в растительности и влажности поверхности.

При этом чистые воды водоемов и водотоков выглядят темными, почти черными, так как равномерно поглощают большую часть солнечных лучей во всех указанных участках спектра. Если в воде имеется взвесь минеральных частиц или высокая плотность планктонных организмов, такие акватории выглядят как оттенки синего и голубого тонов. Такой же цвет могут иметь и мелководные участки, где сквозь толщу воды проходит солнечный свет, отраженный от дна водоема и/или от водной растительности.

Участки суши, покрытые сомкнутой растительностью, как правило, выглядят при данном цветовом синтезе как различные оттенки зеленого. При этом, более темные и насыщенные тона, как правило, соответствуют древесной растительности, а более светлые — травянистой и кустарниковой.

Открытый грунт, в зависимости от его состава и характера, отображается различными оттенками красного и фиолетового. Частично покрытые растительностью участки (имеющие несомкнутый растительный покров) могут иметь различные промежуточные оттенки (розоватые, бурые и пр.). Красноватые тона может иметь и болотная растительность. Голые скалы и песок, как правило, имеют различные оттенки голубого и синего. Близкие оттенки имеют и искусственные поверхности, созданные человеком: здания, сооружения, дороги с твердым покрытием и пр. (Что не удивительно: кирпич и бетон — основные строительные материалы.) Солончаки, а также лед и снег — практически белые или голубоватые (обычно — если покрыты водой). Песок и бетон тоже могут выглядеть очень светлыми, почти такими же белыми, как снег или соль. Но песок может иметь и розовый цвет.

Поэтому для отнесения ландшафтов / объектов к той или иной категории недостаточно просто посмотреть их цвет на изображениях — необходимо проанализировать весь окружающий ландшафт, а также форму и текстуру антропогенных объектов.

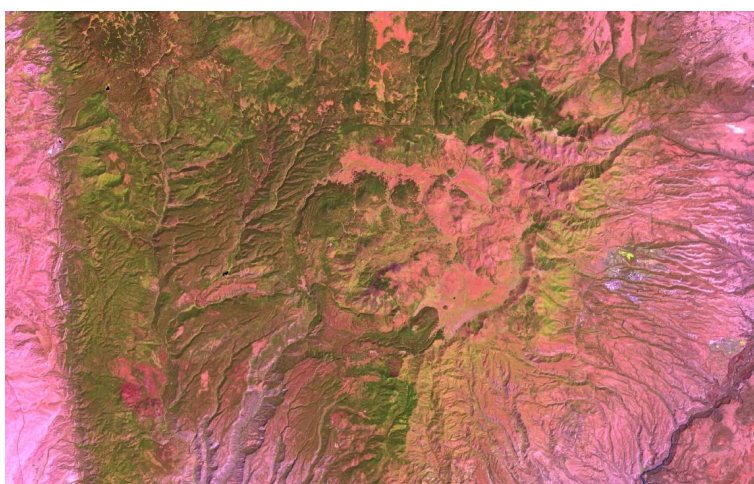
Изображение 1:



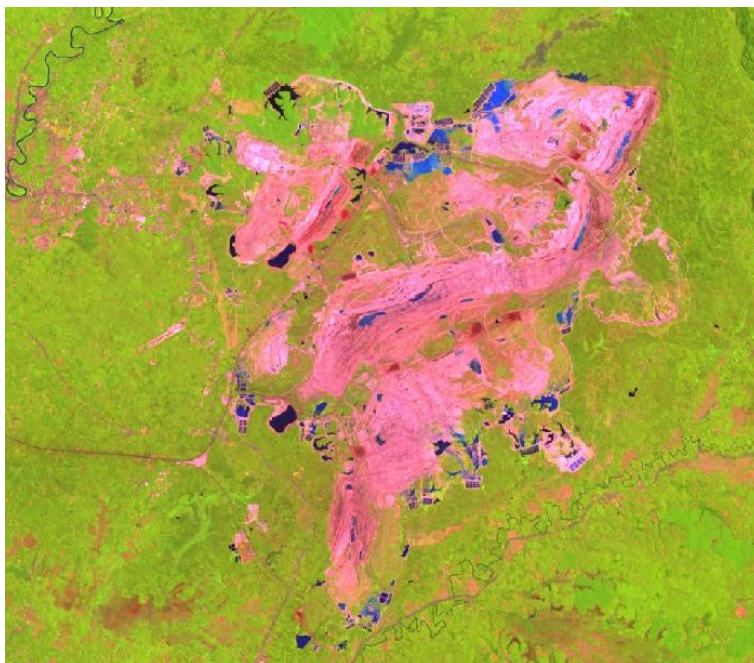
Изображение 2:



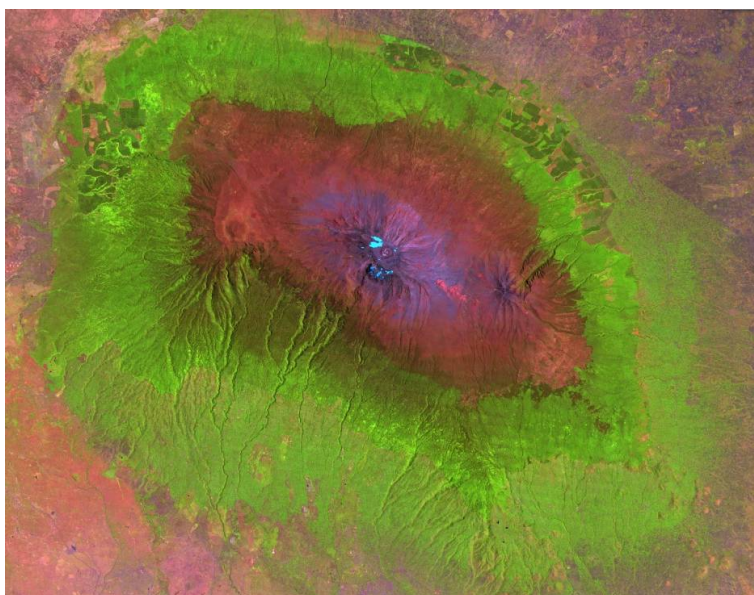
Изображение 3:



Изображение 4:



Изображение 5:



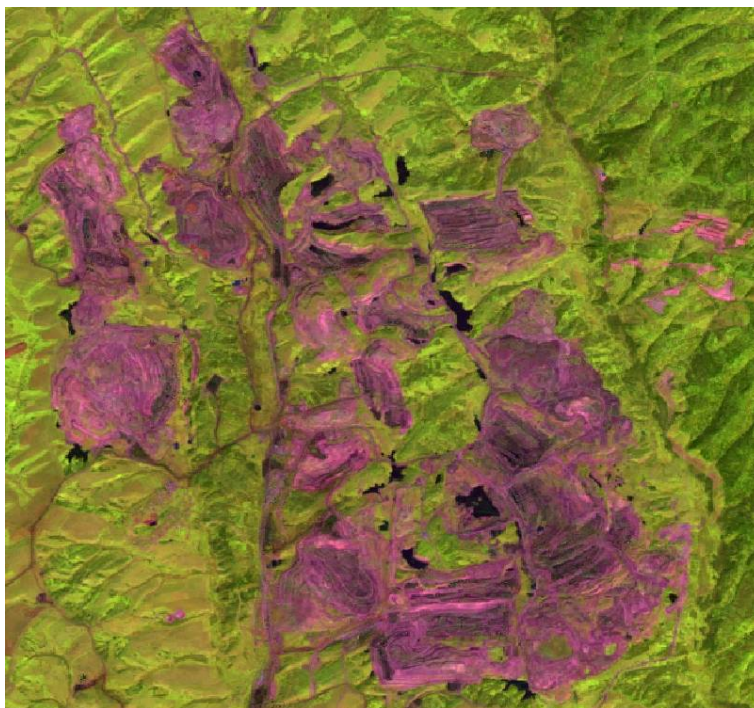
Изображение 6:



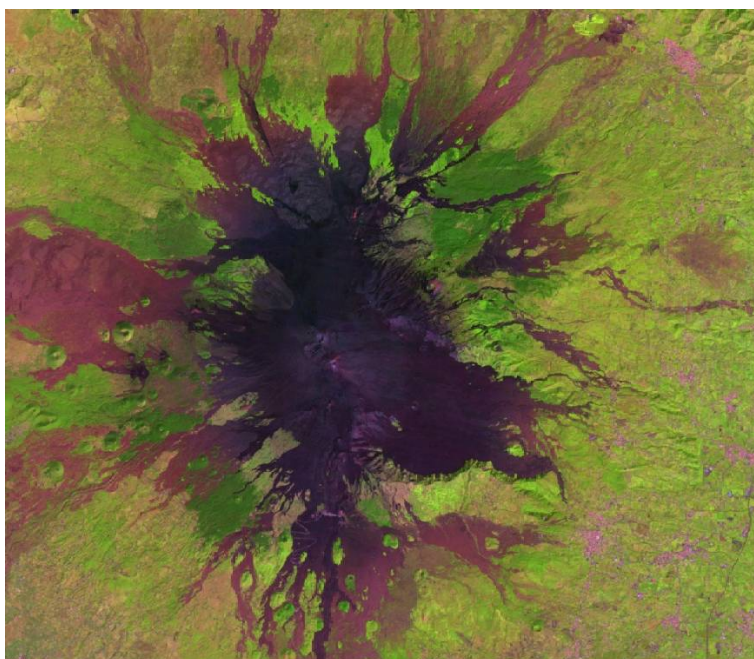
Изображение 7:



Изображение 8:



Изображение 9:



Отметьте верные ячейки

Спутниковые изображения	Вулканы	Добыча угля	Добыча иных полезных ископаемых
Изображение 1	1	2	3
Изображение 2	4	5	6
Изображение 3	7	8	9
Изображение 4	10	11	12
Изображение 5	13	14	15
Изображение 6	16	17	18
Изображение 7	19	20	21
Изображение 8	22	23	24
Изображение 9	25	26	27

Ответ: 3; 4; 7; 11; 13; 18; 21; 23; 25.

Задача I.2.4.7. (10 баллов)

Мировое производство электроэнергии из возобновляемых источников (исключая гидроэнергетику) растет в мире (<https://data.worldbank.org/indicator/EG.ELC.RNWX.KH?view=chart>) почти экспоненциально. Важнейший компонент этого роста — солнечная и ветровая генерация, считающиеся одними из самых "экологически чистых". Всемирный банк и Международная финансовая корпорация, образующие Группу Всемирного банка, поддерживают различные проекты, способствующие развитию альтернативной энергетики на международном уровне и на уровне отдельных стран. Среди таких проектов — Глобальный атлас солнечной энергии (Global Solar Atlas — <https://globalsolaratlas.info/>) и Глобальный атлас ветровой энергии (Global Wind Atlas — <https://globalwindatlas.info/>), объединенные в рамках единой платформы Energydata.info (<https://energydata.info/>).

В частности Глобальный атлас солнечной энергии (<https://globalsolaratlas.info/>) содержит картографическую информацию, полученную на основе, в том числе, космических снимков, о потенциале фотоэлектрической энергии для всей территории суши между 60 градусами северной широты и 45 градусами южной широты. Атлас оценивает как теоретический потенциал (усредненное по времени количество солнечной радиации, падающей на единицу поверхности в разных точках; в киловатт-часах на кв.м.в день), так и потенциал практически доступной энергии, которую можно получить с помощью однофазных кремниевых солнечных батарей, установленных под оптимальным углом к солнцу (в том числе, с учетом физических препятствий для установки фотоэлектрических модулей и законодательных ограничений, связанных с охраной природы и сельскохозяйственных земель).

Опираясь на данные этого портала, расположите страны из приведенного ниже списка по уменьшению среднего по стране теоретического потенциала (одного из вариантов) — количества солнечной энергии, падающей в среднем в день на квадратный метр поверхности фотоэлектрической панели, установленной под оптимальным углом к солнцу (чтобы получать максимум солнечной радиации в течении года) (global tilted irradiation).

Расположите элементы списка в правильном порядке

1. Китай.
2. Израиль.

3. Индия.
4. США (без почти всей Аляски).
5. Алжир.
6. Чили (севернее 45 параллели).
7. Сирия.
8. Намибия.
9. Египет.
10. Австралия.

Ответ: 8; 9; 5; 2; 6; 10; 7; 3; 4; 1.

Задача I.2.4.8. (10 баллов)

Участок земной поверхности вокруг точки с географическими координатами 11°12'06» северной широты и 35°06'28» восточной долготы после 2020 года, очевидно, круглогодично станет акваторией в результате строительства плотины и начала заполнения водохранилища. Однако, строительство данного гидроузла идет уже несколько лет. В результате этого данный участок уже несколько лет как бывает кратковременно покрыт водой во время паводков.

Воспользуйтесь порталом Global Surface Water Explorer (<https://global-surface-water.appspot.com/>) и/или порталами, которые позволяют рассматривать онлайн космические снимки за разные периоды времени (например, LandLook Viewer (<https://landlook.usgs.gov/viewer.html>) или GloVis (<https://glovis.usgs.gov/>) Геологической службы США, USGS, программой Google Планета Земля Pro (<https://www.google.com/intl/ru/earth/desktop/>)). Определите год (четырёхзначное число), когда участок вблизи указанной точки впервые был сезонно покрыт водой во время паводка. Лучше проверить Ваши выводы, по крайней мере, по двум разным источникам.

Ответ: 2017.

Задача I.2.4.9. (10 баллов)

Определите страну по описанию. В поле ниже впишите ее название по-русски. Не используйте значок удара.

Согласно Глобальному атласу солнечной энергии (<https://globalsolaratlas.info/>), данная страна входит в двадчатку мировых лидеров по практическому потенциалу фотоэлектрической энергии (правда, это без учета исключения из рассмотрения сельскохозяйственных земель и земель особо охраняемых природных территорий), хотя по теоретическому потенциалу (среднему количеству солнечной радиации, падающей на единицу поверхности ее территории) позиция страны далека от лидерской.

Согласно данным портала Всемирной лесной вахты (<https://www.globalforestwatch.org/>)(Global Forest Watch), леса (древесная растительность с сомкнутостью древесного полога 30% и выше) занимали, по состоянию и на 2000 около 60% территории этой страны. Хотя к 2010 году эта цифра несколько снизилась, по абсолют-

ным цифрам общей площади лесов страна продолжала входить в десятку мировых лидеров. Более четверти лесов страны входит в состав малонарушенных лесных территорий (intact forests landscapes), по данным на 2016 год.

Однако, потери лесного покрова страны в XXI веке значительные. Согласно данным того же портала Всемирной лесной вахты, с 2000 года страна лишилась более 8% своего лесного покрова. Более трех четвертей всех этих потерь — безвозвратные (на этих площадях можно говорить об обезлесивании). Особенно трагическим для лесов оказался 2019 год. В один этот год было потеряно более 1% лесов страны. Это был худший год за всю историю наблюдений (с 2000 года).

Площадь постоянно покрытых водой участков в стране также сокращается. По данным портала Freshwater Ecosystems Explorer (<https://www.sdg661.app/>), с 2000 года их количество уменьшилось примерно на треть. (Часть этих территорий заливается водой сезонно, часть — не заливается вовсе.)

Ответ: Боливия.

Задача I.2.4.10. (10 баллов)

Наблюдение Земли из космоса (дистанционное зондирование) широко используется сегодня как инструмент изучения нашей планеты. Современные спутники дистанционного зондирования фиксируют интенсивность электромагнитного излучения, отраженного от поверхности Земли, не только в видимом свете, но и в тех частях спектра электромагнитных волн, которые не различимы человеческим глазом: в ультрафиолетовом, ближнем инфракрасном, средневолновом инфракрасном и дальнем инфракрасном. (Радарные спутники также фиксируют радиоволновое излучение.) Электромагнитные волны разной длины по-разному отражаются от поверхности Земли и поэтому несут разную информацию о ней. Аппаратура современных спутников обычно измеряет и записывает яркость излучения в разных диапазонах длин волн (в разных спектральных каналах) независимо. Число таких каналов может быть различно. Как правило, их не меньше 3-4-х, а иногда — 10 и больше.

Используя три любых спектральных канала, можно изучать снимок как цветное изображение на компьютере. При этом зафиксированную аппаратурой спутника яркость электромагнитного излучения в одном канале представляют как яркость красного цвета, в другом — как яркость зеленого цвета, в третьем — как синего. Такие изображения Земли из космоса называют мультиспектральными (многоканальными). Если складывать мультиспектральное изображение из каналов в диапазоне видимого света и отображать яркость в сине-голубом диапазоне синим цветом, в желто-зеленом — зеленым цветом и в красном диапазоне — красным, то такое изображение («в естественных цветах») будет похоже на обычную цифровую фотографию — как если бы сами фотографировали Землю со спутника цифровым фотоаппаратом. Примерно в таком виде космические снимки размещаются на геопорталах общего пользования — таких как «Яндекс карты» и «Google карты».

Однако, можно отображать на компьютере синим, зеленым и красным цветами не обязательно яркости излучения в видимой части электромагнитного спектра — можно использовать яркости в тех диапазонах, которые человеческий глаз не различает — в ультрафиолетовом и инфракрасном. Часто это позволяет нам увидеть такую информацию о нашей планете, которую даже космонавты не могут увидеть с орбиты собственными глазами. Конечно, такие изображения не будут выглядеть как

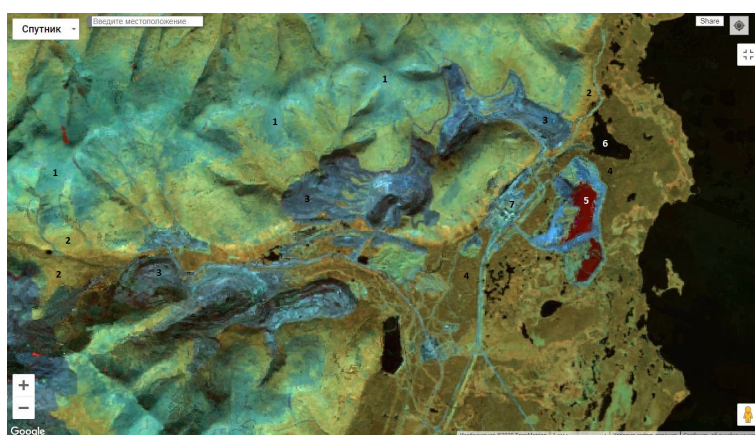
привычные нам фотографии — это будут изображения в условных цветах. Но часто, они несут больше информации о поверхности нашей планеты, чем изображения в видимой части спектра.

Рассмотрите приведенные ниже мультиспектральные изображения участка территории Мурманской области. Эти изображения земной поверхности получены со спутников Landsat 7 и Landsat 8. Они не являются единовременным снимком, а получены с помощью синтеза ряда безоблачных снимков летнего сезона за 2019 год, обработанных таким образом, чтобы компенсировать всегда имеющиеся различия в положении солнца, положении спутника, прозрачности атмосферы и пр. Обработка проведена лабораторией GLAD (Global Land Analysis & Discovery — <https://glad.umd.edu/>) Географического факультета Университета Мэриленда (США). Данные изображения сделаны с помощью портала Глобальные изменения лесного покрова (<http://earthenginepartners.appspot.com/science-2013-global-forest>).

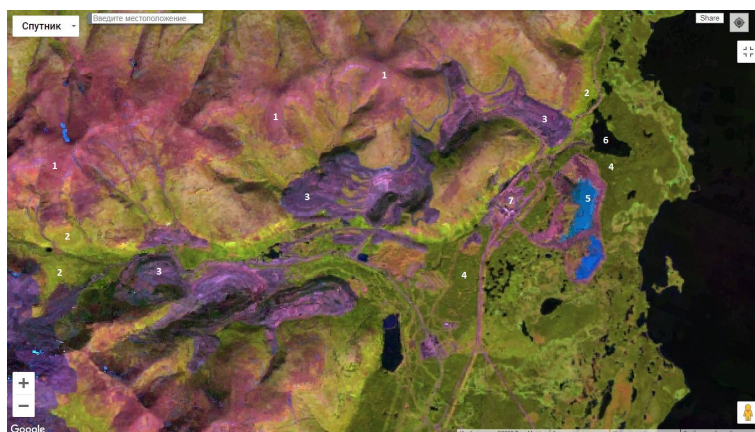
Описание спектральных каналов этих снимков можно найти, например, здесь (<https://www.usgs.gov/faqs/what-are-band-designations-landsat-satellites>) или здесь ([https://ru.wikipedia.org/wiki/Landsat-8#Operational_Land_Imager_\(OLI\)](https://ru.wikipedia.org/wiki/Landsat-8#Operational_Land_Imager_(OLI))). Обратите внимание, что нумерация сходных каналов у двух спутников отличается. Кроме того, у Landsat 8 есть каналы, которые отсутствуют у Landsat 7. При этом спектральные диапазоны соответствующих по спектру каналов обоих спутников практически одинаковы.

На изображениях цифрами обозначены различные объекты (однородные участки местности), список которых приведен ниже. Используя знания о географии нашей страны и информацию о спектральных каналах данных спутников, поставьте в соответствие каждому номеру участка его характеристику. Про использование различных комбинаций спектральных каналов спутников Landsat для изучения Земли можно прочесть, например, здесь (<http://gis-lab.info/qa/landsat-bandcomb.html>) и здесь (<https://habr.com/post/183416/>). Номера каналов приведены для Landsat 7.

Изображение 1: красным цветом представлена интенсивность излучения в канале 4, зеленым — в канале 5, синим — в канале 7.



Изображение 2: красным цветом представлена интенсивность излучения в канале 5, зеленым — в канале 4, синим — в канале 3.



Сопоставьте значения из двух списков:

Участок 1	а. Горная тундра
Участок 2	б. Промышленные строения (обогажительная фабрика)
Участок 3	с. Карьеры для добычи полезных ископаемых
Участок 4	д. Хвойный лес
Участок 5	е. Хвостохранилище (склад отработанной породы)
Участок 6	ф. Лиственный лес
Участок 7	г. Озеро с чистой водой

Ответ: 1 — а; 2 — б; 3 — с; 4 — д; 5 — е; 6 — ф; 7 — г.