

Заключительный этап

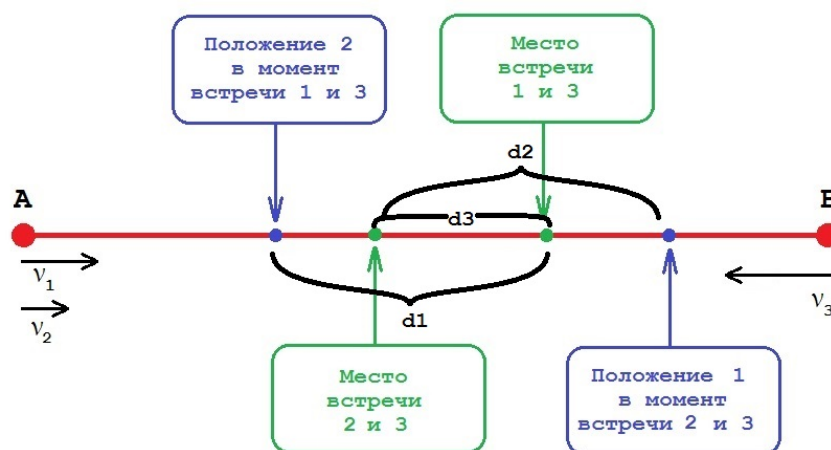
Индивидуальный предметный тур

Информатика. 8–11 класс

Задача III.1.1.1. Интеллектуальная ГИС (12 баллов)

Вы являетесь тестером в команде разработчиков новой интеллектуальной геоинформационной системы. Данная система помимо прямых расчетов расположения отслеживаемых объектов, должна уметь определять необходимые данные (расстояния, координаты и т. п.) путем сопоставления множества косвенных данных.

В данный момент тестируется следующая ситуация: есть три объекта 1, 2 и 3. Объекты 1 и 2 одновременно стартуют из точки A в направлении точки B , объект 3 в этот же момент стартует из точки B в направлении точки A . Известно, что скорость объекта 1 (обозначим ее v_1) строго больше скорости объекта 2 (обозначим ее v_2), но сами значения этих скоростей не известны. Однако известна их разность $dv = v_1 - v_2$. Очевидно, что при своем движении, объект 3 сначала встретится с объектом 1, а затем через какое-то время встретится с объектом 2. Известны следующие расстояния: d_1 — расстояние между 1 и 2 в момент встречи третьего и первого, d_2 — расстояние между 1 и 2 в момент встречи третьего и второго, d_3 — расстояние между точками встречи. Требуется по заданным четырем параметрам определить расстояние между точками A и B . В момент встречи второго и третьего первый не успевает достичь точки B .



Формат входных данных

На вход подаются четыре положительных числа dv , d_1 , d_2 , d_3 через пробел. Все числа целые в пределах от 1 до 10^9 . $d_3 < d_1 < d_2$. Входные данные таковы, что в момент встречи второго и третьего первый еще не успевает достичь точки B .

Формат выходных данных

Вывести одно число — расстояние между точками AB . Ответ следует выводить с округлением ровно до пяти знаков после десятичной точки.

Примеры

Пример №1

Стандартный ввод
3 9 12 5
Стандартный вывод
36.00000

Пример программы-решения

Ниже представлено решение на языке C++.

```

1  #include <bits/stdc++.h>
2  #define int long long
3  using namespace std;
4
5  int main(){
6      long double dv, d1, d2, d3;
7      cin >> dv >> d1 >> d2 >> d3;
8
9      cout << fixed << setprecision(5) << d1 * d2 / (d2 - d1);
10     return 0;
11 }
```

Пример программы-решения

Ниже представлено решение на языке Python.

```

1  from decimal import Decimal
2  a = list(map(int, input().split()))
3  d1 = Decimal(str(a[1]))
4  d2 = Decimal(a[2])
5  n = 5
6  a = d1 * d2 / (d2 - d1)
7  print(f'{a:.{n}f}')
```

Задача III.1.1.2. Самоорганизация в городе (15 баллов)

Рассмотрим следующую модель обмена ресурсами внутри некоторого города. Город состоит из n расположенных в один ряд кварталов. Каждый квартал изначально имеет a_i единиц запаса некоторого ресурса, причем никакие два квартала не обладают одинаковыми запасами. Для любых двух рядом расположенных кварталов возможна операция передачи одной единицы ресурса от большего по запасам к меньшему, но, само собой, отдающий квартал не должен стать беднее по запасам того,

кому он отдает. Основным требованием властей города является разнообразие запасов, что означает что ни в какой момент времени никакие два квартала не должны обладать одинаковыми по величине запасами. Таким образом, операция передачи может быть осуществлена только тогда, когда после ее выполнения новые величины запасов обоих участвовавших в операции передачи ресурса кварталов не совпадают ни с одним другим кварталом (в том числе и между собой). Если есть несколько пар рядом стоящих кварталов, имеющих возможность передачи, то выбирается пара с наибольшей разностью, а если и таких несколько, то среди них выбирается пара с минимальным значением запаса у получающего помощь квартала. За одну единицу времени возможен ровно один обмен. Такие операции поочередно производятся до тех пор, пока это не нарушает правила разнообразия. Требуется выяснить, каким станет распределение ресурса после окончания обменов.

Формат входных данных

В первой строке содержится число n — количество кварталов. Во второй строке находятся n натуральных чисел a_i через пробел — запасы ресурса у соответствующего квартала. Все величины запасов попарно различны. Все числа на входе в пределах от 1 до 1000.

Формат выходных данных

Вывести в одну строку через пробел n чисел — запасы в кварталах после того, как станет невозможно производить операции передачи ресурса.

Примеры

Пример №1

Стандартный ввод
6
2 10 9 3 6 12
Стандартный вывод
3 9 8 5 7 10

Пояснение к примеру 1

Приведем последовательность перераспределений:

- 0) 2 10 **9 3** 6 12
- 1) **2 10** 8 4 6 12
- 2) 3 9 8 4 **6 12**
- 3) 3 9 8 **4 7** 11
- 4) 3 9 8 5 **6 11**
- 5) 3 9 8 5 7 10

В нулевой строке исходные значения. Самая большая разница между 10 и 2, но эту операцию запрещено проводить, так как тогда во втором квартале будет 9, но 9 уже есть в третьем. Следующая по величине разница между 9 и 3, а так же между 12 и 6. Тогда производится помощь наиболее нуждающемуся в этих двух вариантах,

а именно $9 \rightarrow 3$. В первой строке результат передачи. Теперь самая большая разница между 2 и 10 и ничто не мешает произвести передачу $10 \rightarrow 2$. Во второй строке результат. Теперь снова есть два варианта $9 \rightarrow 3$ и $12 \rightarrow 6$, но теперь первый запрещен по причине того, что после этого получатся числа 4 и 8, но эти числа уже во второй строке есть. Тогда выполняется второй вариант $12 \rightarrow 6$, получается третья строка. По-прежнему нельзя $9 \rightarrow 3$, но и $11 \rightarrow 7$ нельзя, так как тогда будет повторение 8. Следующей по разности парой будет $7 \rightarrow 4$, которая и выполнит операцию. Получим четвертую строку. $3 \rightarrow 9$ снова запрещено, но теперь можно $11 \rightarrow 6$, и получим пятую строку. И теперь ни одной операции не получится произвести, то есть эта строка и будет ответом.

Пример программы-решения

Ниже представлено решение на языке C++.

```

1  #include <bits/stdc++.h>
2
3  using namespace std;
4  typedef long long ll;
5  typedef pair<int, int> pii;
6  typedef long double ld;
7
8  vector<int> mask;
9
10 bool ok_turn(int a, int b){
11     if(a > b)
12         swap(a, b);
13
14     if(mask[a+1] == 0 && mask[b-1] == 0 && a+2 != b)
15         return 1;
16
17     return 0;
18 }
19
20 int main(){
21     ios::sync_with_stdio(0), cin.tie(0), cout.tie(0);
22
23     int n;
24     cin >> n;
25     vector<int> v(n);
26     mask.resize(1001, 0);
27     for(int i = 0; i < n; i++){
28         cin >> v[i];
29         mask[v[i]] = 1;
30     }
31
32     while(1){
33         int big = -1, small = -1;
34
35         for(int i = 1; i < n; i++){
36             if(ok_turn(v[i-1], v[i])){
37                 int tbig = i-1;
38                 int tsmall = i;
39                 if(v[tbig] < v[tsmall])
40                     swap(tbig, tsmall);
41                 if(big == -1){
42                     big = tbig;

```



```

43         small = tsmall;
44     }
45     else{
46         if(v[big] - v[small] < v[tbig] - v[tsmall] ||
47            (v[big] - v[small] == v[tbig] - v[tsmall] && v[tsmall] < v[small])){
48             big = tbig;
49             small = tsmall;
50         }
51     }
52 }
53
54 if(big == -1) break;
55 mask[v[big]] = 0;
56 v[big]--;
57 mask[v[big]] = 1;
58 mask[v[small]] = 0;
59 v[small]++;
60 mask[v[small]] = 1;
61 }
62
63 for(auto q : v) cout<<q<<' ';
64 cout<<endl;
65 return 0;
66 }

```

Задача III.1.1.3. Проектирование электросети (15 баллов)

Рассмотрим проект модели снабжения мегаполиса электроэнергией. Имеется n источников, генерирующих электроэнергию для нужд мегаполиса и уже запланированная структура линий электропередач, соединяющая эти источники с мегаполисом. Совокупно источники и мегаполис будем называть объектами. Для каждого источника известно, какую мощность он генерирует, кроме того известно, какие объекты будут соединены между собой передающими линиями. Известно, что вся сеть имеет древовидную структуру, то есть она связная и не имеет в своем составе циклов. Считаем, что мегаполис потребит всю мощность, которую получится доставить. Потери мощности при передаче считаем равными нулю.

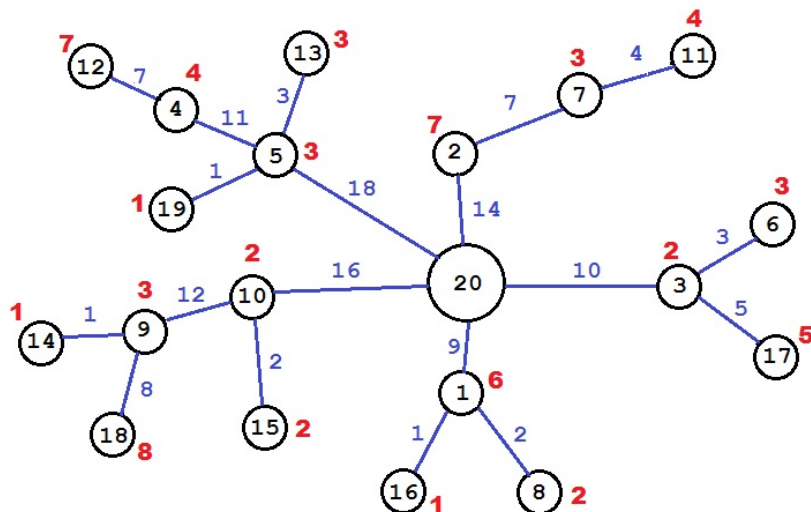
Требуется рассчитать пропускную способность каждой линии в проектируемой сети, то есть наибольшую мощность, которую можно передать по этой линии. При этом нужно указать минимально необходимую пропускную способность, такую, чтобы все имеющиеся генерируемые мощности были доставлены до мегаполиса.

Формат входных данных

В первой строке содержится число n — количество источников электроэнергии. Источники пронумерованы от 1 до n . Мегаполис имеет номер $n + 1$. Далее в n следующих строках содержится по два числа a_i , b_j — обозначающих, что объекты номер a_i и b_j по проекту будут соединены линией. В последней строке содержатся n чисел gr_i через пробел, описывающих генерируемую мощность источника номер i . Все числа на входе в пределах от 1 до 1000. Гарантируется, что заданный граф является деревом, то есть связан и не содержит циклов.

Формат выходных данных

Для каждой линии в соответствующей ей отдельной строке указать необходимую минимальную пропускную способность. Порядок линий на выходе должен совпадать с порядком линий на входе.



Примеры

Пример №1

Стандартный ввод	
19	
12 4	
19 5	
5 13	
4 5	
5 20	
2 20	
20 10	
7 2	
10 15	
10 9	
18 9	
14 9	
1 20	
1 8	
16 1	
20 3	
3 6	
17 3	
7 11	
6 7 2 4 3 3 3 2 3 2 4 7 3 1 2 1 5 8 1	

Стандартный вывод
7
1
3
11
18
14
16
7
2
12
8
1
9
2
1
10
3
5
4

Пояснение к примеру

На рисунке приведена схема для первого примера. Красным обозначены генерируемые мощности, синим — необходимые минимальные пропускные способности линий. В частности, видно, что восточный блок источников 3, 6 и 17 генерирует суммарную мощность равную 10, а юго-западный подблок 9, 14, 18 требует для себя линии с пропускной способностью 12, и т. д.

Пример программы-решения

Ниже представлено решение на языке C++.

```

1  #include <bits/stdc++.h>
2
3  using namespace std;
4  typedef long long ll;
5  typedef pair<int, int> pii;
6  typedef long double ld;
7
8  int n, a, b;
9  vector<vector<pii> > G;
10 vector<int> gp, ans;
11
12 void dfs(int a, int p){
13     for(auto q : G[a])
14         if(q.first != p){
15             dfs(q.first, a);
16             ans[q.second] = gp[q.first];
17             gp[a] += gp[q.first];
18         }
19 }
20
21 int main(){

```

```

22     ios::sync_with_stdio(0), cin.tie(0), cout.tie(0);
23
24     cin >> n;
25     n++;
26     G.resize(n+2);
27     gp.resize(n+1);
28     ans.resize(n-1);
29     for(int i = 0; i < n-1; i++){
30         cin >> a >> b;
31         G[a].push_back({b, i});
32         G[b].push_back({a, i});
33     }
34
35     for(int i = 1; i < n; i++)
36         cin >> gp[i];
37
38     dfs(n, n);
39
40     for(int i = 0; i < ans.size(); i++)
41         cout << ans[i] <<endl;
42
43     return 0;
44 }

```

Пример программы-решения

Ниже представлено решение на языке Python.

```

1  import sys
2  def dfs(u, par):
3      global getId, a, dp, ans
4      for elem in a[u]:
5          if elem != par:
6              dfs(elem, u)
7              dp[u] += dp[elem]
8              ans[getId[u][elem]] = dp[elem]
9  sys.setrecursionlimit(100000)
10 n = int(input())
11 n += 1
12 a = []
13 getId = []
14 ans = [0] * (n + 1)
15 dp = [0] * (n + 1)
16 for i in range(n + 1):
17     a.append([])
18     add = []
19     for j in range(n + 1):
20         add.append(-1)
21     getId.append(add)
22 for i in range(n - 1):
23     sd = input().split()
24     u = int(sd[0])
25     v = int(sd[1])
26     a[u].append(v)
27     a[v].append(u)
28     getId[u][v] = i + 1
29     getId[v][u] = i + 1
30 oo = list(map(int, input().split()))
31 for i in range(len(oo)):

```

```

32     dp[i + 1] = oo[i]
33     dfs(n, -1)
34     for i in range(1, n):
35         print(ans[i])

```

Задача III.1.1.4. Матрица кратчайших расстояний (25 баллов)

Рассмотрим уже известную модель дорожной сети, получаемую при анализе снимка местности. Изображение состоит только из точек двух типов. Точка, отмеченная 1, изображает дорожное полотно, все остальное изображается точками 0. При этом ширина дороги всегда ровно 1 точка. Нигде на изображении нет квадрата 2×2 , состоящего из точек 1. Будем считать, что в точках, являющихся пересечением трех или четырех дорог находятся населенные пункты. Пронумеруем эти населенные пункты от 1 до K , сверху вниз и слева направо (см. пример, $K = 9$).

												1		2		
				3												
						4								5		
		6		7		8										
						9										

В данном случае нужно решить классическую задачу теории графов для не самого классического способа задания этих графов — при помощи упрощенного рисунка. А именно, нужно для каждой пары населенных пунктов определить кратчайшее расстояние между ними при движении по дорогам, или выяснить, что по дорогам попасть из одного в другой не получится. Будем считать, что перемещаться можно только из одной клетки помеченной 1 в соседнюю по стороне, так же помеченную 1, при этом длина такого перемещения равна 1.

Формат входных данных

Первая строка содержит два числа N и M через пробел — размер изображения. Далее в N строках содержится по M символов 0 или 1 без пробелов между ними, соответствующих изображению. Нигде на изображении нет квадрата 2×2 , состоящего из точек 1. N и M находятся в пределах от 5 до 50. Гарантируется, что есть хотя бы один населенный пункт.

Формат выходных данных

Вывести K строк, по K чисел в каждой — матрицу кратчайших расстояний. В i -й строке на j -й позиции должно находиться кратчайшее расстояние между пунктами номер i и номер j . Обратите внимание на формат вывода: каждое расстояние

требуется выводить в формате пять знаков, недостающие символы нужно заменять пробелами. Если из одного пункта попасть в другой нельзя, нужно выводить в таком случае -1 .

Примеры

Пример №1

Стандартный ввод									
8 16									
0000100001111111									
1111111001010101									
0000100001010001									
0000001001100101									
0000001110111101									
1111111010000101									
1010101110100111									
1110111000111000									
Стандартный вывод									
0	2	-1	-1	10	-1	-1	-1	-1	
2	0	-1	-1	12	-1	-1	-1	-1	
-1	-1	0	-1	-1	-1	-1	-1	-1	
-1	-1	-1	0	-1	5	3	1	2	
10	12	-1	-1	0	-1	-1	-1	-1	
-1	-1	-1	5	-1	0	2	4	5	
-1	-1	-1	3	-1	2	0	2	3	
-1	-1	-1	1	-1	4	2	0	1	
-1	-1	-1	2	-1	5	3	1	0	

Пример программы-решения

Ниже представлено решение на языке C++.

```

1  #include <bits/stdc++.h>
2
3  using namespace std;
4  typedef long long ll;
5  typedef pair<int, int> pii;
6  typedef long double ld;
7
8  int n, m;
9  vector<string> S;
10 vector<vector<int>> > M, H;
11 int dx[4] = {0, -1, 0, 1};
12 int dy[4] = {-1, 0, 1, 0};
13
14 bool intr(int x, int y){
15     return (x >= 0 && y >= 0 && x < n && y < m);
16 }
17
18 bool np(int x, int y){
19     int r = 0;
20     for(int i = 0; i < 4; i++){
21         int nx = x + dx[i];

```

```

22     int ny = y + dy[i];
23     if(intr(nx, ny))
24         r += S[nx][ny] - '0';
25 }
26
27 return (r >= 3 && S[x][y] == '1');
28 }
29
30 void dfs(int x, int y, int d, int p){
31     if(M[x][y] == 0)
32         M[x][y] = 1;
33     for(int i = 0; i < 4; i++){
34         int nx = x + dx[i];
35         int ny = y + dy[i];
36
37         if(intr(nx, ny) && S[nx][ny] == '1'){
38             if(M[nx][ny] == 0)
39                 dfs(nx, ny, d+1, p);
40             else
41                 if(M[nx][ny] < 0 && -M[nx][ny] != p){
42                     int np = -M[nx][ny];
43                     H[np][p] = min(H[np][p], d+1);
44                     H[p][np] = min(H[p][np], d+1);
45                 }
46         }
47     }
48 }
49
50 int main(){
51     ios::sync_with_stdio(0), cin.tie(0), cout.tie(0);
52
53     cin >> n >> m;
54     S.resize(n);
55     for(int i = 0; i < n; i++)
56         cin >> S[i];
57
58     M.resize(n, vector<int>(m, 0));
59
60     int K = 0;
61     for(int i = 0; i < n; i++)
62         for(int j = 0; j < m; j++)
63             if(np(i, j)){
64                 K--;
65                 M[i][j] = K;
66             }
67
68     H.resize(-K+1, vector<int>(-K+1, 1e9));
69     for(int i = 1; i <= -K; i++)
70         H[i][i] = 0;
71
72     for(int i = 0; i < n; i++)
73         for(int j = 0; j < m; j++)
74             if(M[i][j] < 0)
75                 dfs(i, j, 0, -M[i][j]);
76
77
78     for (int k = 1; k <= -K; k++)
79         for (int i = 1; i <= -K; i++)
80             for (int j = 1; j <= -K; j++)
81                 H[i][j] = min (H[i][j], H[i][k] + H[k][j]);

```

```

82
83     for(int i = 1; i <= -K; i++)
84         for(int j = 1; j <= -K; j++)
85             if(H[i][j] == 1e9)
86                 H[i][j] = -1;
87
88     for(int i = 1; i <= -K; i++){
89         for(int j = 1; j <= -K; j++)
90             cout<<setw(5)<<H[i][j];
91         cout<<endl;
92     }
93     return 0;
94 }

```

Пример программы-решения

Ниже представлено решение на языке Python.

```

1  import queue
2  dx = [-1, 0, 1, 0]
3  dy = [0, 1, 0, -1]
4  n, m = list(map(int, input().split()))
5  a = [['0'] * (m + 2) for i in range(n + 2)]
6  check = [[-1] * (m + 2) for i in range(n + 2)]
7  for i in range(1, n + 1):
8      om = input()
9      for j in range(m):
10         a[i][j + 1] = om[j]
11  yk = 1
12  ff = []
13  for i in range(1, n + 1):
14      for j in range(1, m + 1):
15         if a[i][j] == '1':
16             cc = ord(a[i - 1][j]) + ord(a[i + 1][j]) + ord(a[i][j - 1]) + ord(a[i][j +
17             ↪ 1]) - ord('0') - ord('0') - ord('0') - ord('0')
18             if cc >= 3:
19                 check[i][j] = yk
20                 ff.append([i, j])
21                 yk += 1
22  for k in range(len(ff)):
23      Q = queue.Queue()
24      Q.put(ff[k])
25      dist = [['1000000000'] * (m + 2) for i in range(n + 2)]
26      dist[ff[k][0]][ff[k][1]] = 0
27      while Q.empty() == False:
28         u = Q.get()
29         for i in range(4):
30             if (u[0] + dy[i] >= 1 and u[0] + dy[i] <= n and u[1] + dx[i] >= 1 and u[1]
31             ↪ + dx[i] <= m and a[u[0] + dy[i]][u[1] + dx[i]] == '1' and dist[u[0] +
32             ↪ dy[i]][u[1] + dx[i]] > dist[u[0]][u[1]] + 1):
33                 dist[u[0] + dy[i]][u[1] + dx[i]] = dist[u[0]][u[1]] + 1
34                 Q.put([u[0] + dy[i], u[1] + dx[i]])
35  for i in range(len(ff)):
36      if dist[ff[i][0]][ff[i][1]] == 1000000000:
37         st = repr(-1).rjust(5)
38         print(st, end=' ', flush = True)
39     else:
40         st=repr(dist[ff[i][0]][ff[i][1]]).rjust(5)

```



```

38         print(st,end='',flush = True)
39     print()

```

Задача III.1.1.5. Агломерация (33 баллов)

Агломерация — это «компактное скопление населенных пунктов, главным образом городов, местами срастающихся, объединенных в сложную многокомпонентную динамическую систему с интенсивными производственными, транспортными и культурными связями». Изначально отдельно образовавшиеся поселения со временем разрастаются и сливаются в одно многоуровневое образование — городскую агломерацию. При этом процесс слияния может иметь рекурсивную структуру, когда несколько агломераций сливаются в одну агломерацию более высокого уровня.

В этой задаче нужно построить обработку снимка городской агломерации, находящую ее уровень. Идея очевидна — если в агломерацию входит как часть агломерация уровня $n - 1$ и нет частей-агломераций уровня n , то сама эта рассматриваемая агломерация имеет уровень n .

Но не все так просто — в данной задаче уточним понятие агломерации следующим образом: под агломерацией будем понимать любую квадратную область плоскости, целиком занятую отдельными поселениями, причем любое поселение, имеющееся в этом квадрате, не должно выходить за его пределы. Любой отдельный населенный пункт назовем агломерацией первого уровня. На снимке такой населенный пункт имеет вид квадрата, все ячейки которого обозначаются одним и тем же символом. Из низкоуровневых агломераций путем слияния получаются более высокоуровневые. Для заданного снимка квадратной области, целиком заполненной населенными кварталами, требуется определить его уровень.

	1	2	3	4	5	6
1	a	b	e	e	a	g
2	c	d	e	e	h	h
3	a	a	j	j	h	h
4	a	a	j	j	k	k
5	d	d	m	m	k	k
6	d	d	m	m	n	o

Рассмотрим пример. Каждый единичный квадрат обозначен своим символом и является жилым кварталом. Если несколько рядом стоящих кварталов обозначены одним символом, они образуют населенный пункт, целиком занимающий квадратную область. Каждый такой населенный пункт образует агломерацию первого уровня. Всего таких на рисунке 15. Агломераций второго уровня на рисунке две, укажем их левый верхний и правый нижний углы: (1, 1) — (2, 2) и (3, 1) — (6, 4). Агломераций

третьего уровня одна, ее координаты $(1, 1) — (4, 4)$. Весь квадрат в примере является агломерацией уровня четыре.

Отдельно следует отметить, что разные населенные пункты могут быть обозначены одним символом, при условии, что они не имеют общую границу ненулевой длины.

Формат входных данных

В первой строке находится число n — размер стороны квадрата заданной агломерации ($1 \leq n \leq 100$). В следующих n строках содержится по n маленьких букв из множества $a \dots z$ без пробелов, описывающих населенные пункты в составе агломерации. Буквы одного вида, являющиеся соседними по стороне всегда образуют квадраты. Некоторые пары различных квадратов могут быть обозначены одинаковыми буквами, но при этом данные пары квадратов не имеют общих границ ненулевой длины.

Формат выходных данных

Вывести одно число — уровень представленной во входных данных агломерации.

Примеры

Пример №1

Стандартный ввод
6 abeeag cdeehh aajjhh aajjkk ddmmkk ddmmno
Стандартный вывод
4

Пример программы-решения

Ниже представлено решение на языке C++.

```

1  #include <bits/stdc++.h>
2
3  using namespace std;
4  typedef long long ll;
5  typedef pair<int, int> pii;
6  typedef long double ld;
7
8  int n;
9  vector<vector<int>> > u, d, l, r;
10
11 bool sq(int rx, int ry, int D){

```

```

12     int lx = rx - D + 1;
13     int ly = ry - D + 1;
14     if(r[lx][ly] >= D && d[lx][ly] >= D && u[rx][ry] >= D && l[rx][ry] >= D)
15         return 1;
16     return 0;
17 }
18
19 int main(){
20     ios::sync_with_stdio(0), cin.tie(0), cout.tie(0);
21
22     string s;
23     cin >> n;
24     vector<string> v(n+2, string(n+2, '#'));
25     for(int i = 1; i <= n; i++){
26         cin >> s;
27         s = "#" + s + "#";
28         v[i] = s;
29     }
30
31     u.resize(n+2, vector<int>(n+2, 0));
32     for(int i = 1; i <= n; i++)
33     for(int j = 1; j <= n; j++){
34         u[i][j] = (v[i][j] != v[i][j+1]);
35         u[i][j] += u[i-1][j] * (u[i][j] != 0);
36     }
37
38     l.resize(n+2, vector<int>(n+2, 0));
39     for(int j = 1; j <= n; j++)
40     for(int i = 1; i <= n; i++){
41         l[i][j] = (v[i][j] != v[i+1][j]);
42         l[i][j] += l[i][j-1] * (l[i][j] != 0);
43     }
44
45     d.resize(n+2, vector<int>(n+2, 0));
46     for(int i = n; i >= 1; i--)
47     for(int j = n; j >= 1; j--){
48         d[i][j] = (v[i][j] != v[i][j-1]);
49         d[i][j] += d[i+1][j] * (d[i][j] != 0);
50     }
51
52     r.resize(n+2, vector<int>(n+2, 0));
53     for(int j = n; j >= 1; j--)
54     for(int i = n; i >= 1; i--){
55         r[i][j] = (v[i][j] != v[i-1][j]);
56         r[i][j] += r[i][j+1] * (r[i][j] != 0);
57     }
58
59     vector<vector<int>> > dp(n+2, vector<int>(n+2, 0));
60
61     for(int u = 1; u <= n; u++)
62     for(int i = n; i >= 1; i--)
63     for(int j = n; j >= 1; j--){
64         if(i - u + 1 >= 1 && j - u + 1 >= 1){
65             dp[i][j] = max({dp[i][j], dp[i][j-1], dp[i-1][j], dp[i-1][j-1]});
66             if(sq(i, j, u)) dp[i][j]++;
67         }
68
69     cout << dp[n][n]<<endl;
70
71     return 0;

```

72 }

Пример программы-решения

Ниже представлено решение на языке Python.

```

1  n = int(input())
2  a = [['#' * (n + 2) for i in range(n + 2)]
3  for i in range(1, n + 1):
4      sd = input()
5      for j in range(1, n + 1):
6          a[i][j] = sd[j - 1]
7  ans = []
8  k = [['0' * (n + 2) for i in range(n + 2)]
9  for i in range(n + 2):
10     ans.append(k)
11  for sz in range(1, n + 1):
12     for i in range(1, n - sz + 2):
13         for j in range(1, n - sz + 2):
14             y1 = i
15             x1 = j
16             x2 = j + sz - 1
17             y2 = i + sz - 1
18             xx1 = x1
19             xx2 = x2
20             ch = 1
21             while x1 <= x2:
22                 if (a[y1 - 1][x1] == a[y1][x1] or a[y2][x1] == a[y2 + 1][x1]):
23                     ch = 0
24                     break
25                 x1 += 1
26
27             while y1 <= y2:
28                 if (a[y1][xx1] == a[y1][xx1 - 1] or a[y1][xx2] == a[y1][xx2 + 1]):
29                     ch = 0
30                     break
31             y1 += 1;
32             ans[sz][i][j] = ch + max(ans[sz - 1][i][j], max(ans[sz - 1][i][j + 1],
33                                     ↪ max(ans[sz - 1][i + 1][j], ans[sz - 1][i + 1][j + 1])))
33  print(ans[n][1][1])

```

География. 8-11 класс

Задача III.1.2.1. (10 баллов)

Поставьте в соответствие географические координаты из списка ниже и название крупнейших и/или широко известных островов или архипелагов, на территории которых находится соответствующая географическая точка.

Сопоставьте значения из двух списков

37° 40' с.ш. и 13° 45' в.д.	Сицилия
36° 25' с.ш. и 140° 00' в.д.	Мадагаскар
42° 05' ю.ш. и 146° 10' в.д.	Остров Калимантан
15° 00' ю.ш. и 48° 25' в.д.	Тасмания
69° 30' с.ш. и 41° 50' з.д.	Фолклендские острова
51° 35' ю.ш. и 60° 00' з.д.	Врангеля
71° 15' с.ш. и 179° 50' з.д.	Галапагосские острова
01° 00' с.ш. и 115° 00' в.д.	Гавайи
00° 35' ю.ш. и 90° 20' з.д.	Гренландия
19° 30' с.ш. и 155° 35' з.д.	Хонсю

Ответ: Все приведенные в списке острова достаточно известны. Предполагается, что участники должны представлять, в какой части света и в каком океане они находятся. В списке нет островов, которые были бы расположены достаточно близко, чтобы для их различения было бы необходимо знание точных географических координат. Для соотнесения координат с островами из списка достаточно представлять себе, как в целом устроена система географических координат и в каких пределах они изменяются. Зная это, участники должны, в отсутствии доступа к картам и гео-порталам (типа «Яндекс Карт»), определить, какое сочетание координат абсолютно невозможно для данного острова/архипелага.

37° 40' с.ш. и 13° 45' в.д.	Сицилия
36° 25' с.ш. и 140° 00' в.д.	Хонсю
42° 05' ю.ш. и 146° 10' в.д.	Тасмания
15° 00' ю.ш. и 48° 25' в.д.	Мадагаскар
69° 30' с.ш. и 41° 50' з.д.	Гренландия
51° 35' ю.ш. и 60° 00' з.д.	Фолклендские острова
71° 15' с.ш. и 179° 50' з.д.	Врангеля
01° 00' с.ш. и 115° 00' в.д.	Остров Калимантан
00° 35' ю.ш. и 90° 20' з.д.	Галапагосские острова
19° 30' с.ш. и 155° 35' з.д.	Гавайи

Задача III.1.2.2. (10 баллов)

Поставьте в соответствие каждому озеру из списка ниже тип климата в его окрестностях. В список включены одни из крупнейших и известнейших озер своих стран и континентов.

Сопоставьте значения из двух списков

Онежское (Россия)	умеренный, переходный от морского к континентальному
Байкал (Россия)	тропический, сухой
Танганьика (восточная Африка)	тропический, высокогорный
Мар-Чикита (Южная Америка)	умеренный, резко континентальный
Маракайбо (Южная Америка)	субэкваториальный, чередование влажного и сухого сезонов
Титикака (Южная Америка)	субарктический
Большое Медвежье (Канада)	субтропический, засушливый
Эйр (Австралия)	субэкваториальный, два влажных сезона

Ответ:

Онежское (Россия)	умеренный, переходный от морского к континентальному
Байкал (Россия)	умеренный, резко континентальный
Танганьика (восточная Африка)	субэкваториальный, два влажных сезона
Мар-Чикита (Южная Америка)	субтропический, засушливый
Маракайбо (Южная Америка)	субэкваториальный, чередование влажного и сухого сезонов
Титикака (Южная Америка)	тропический, высокогорный
Большое Медвежье (Канада)	субарктический
Эйр (Австралия)	тропический, сухой

Задача III.1.2.3. (26 баллов)

На приведенных ниже спутниковых изображениях Вы видите часть (относительно небольшую) одной из крупнейших горных систем Земли. Второе изображение показывает фрагмент территории первого в более крупном масштабе. Обратите внимание на масштабные линейки в левом нижнем углу каждого изображения.

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) — зеле-

ным, видимый красный свет — синим. В таком цветовом синтезе хорошо видны различия в растительности и влажности поверхности.

При этом лишённые растительности поверхности (голые скалы, песок) окрашены, как правило, в различные оттенки синего и фиолетового. Аналогично могут выглядеть территории городов с плотной застройкой. (Что не удивительно: кирпич и бетон являются одними из основных строительных материалов.) Лед и снег выглядят белыми или голубыми. Очень похоже выглядят солончаки и пересыхающие мелководные соленые озера.

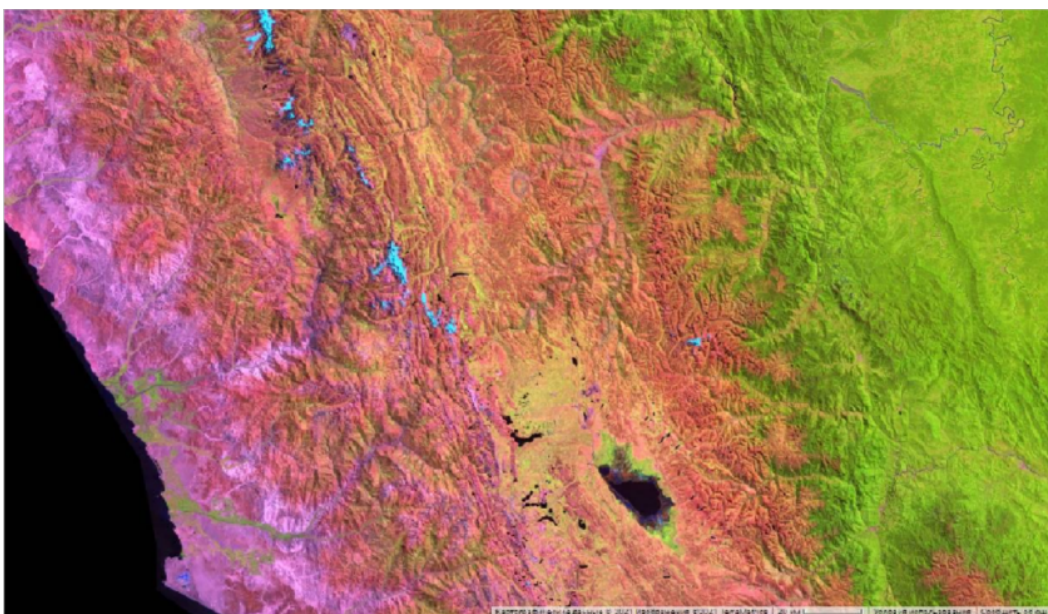
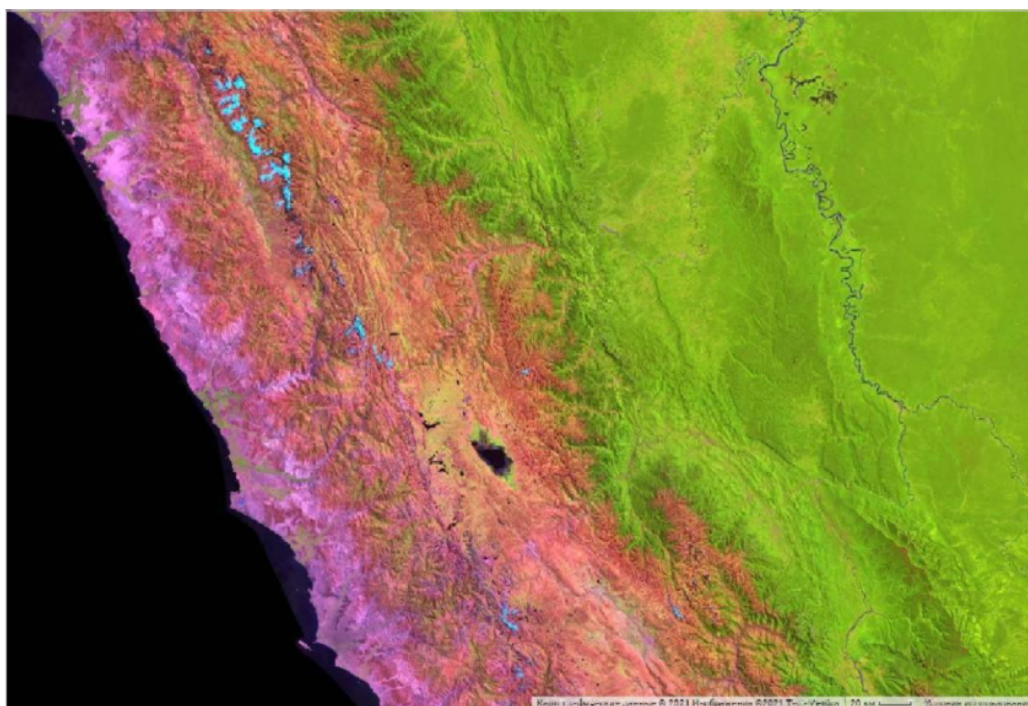
Чистые воды водоемов и водотоков выглядят темными, почти черными, так как равномерно поглощают большую часть солнечных лучей во всех указанных участках спектра. Если в воде имеется взвесь минеральных частиц или высокая плотность планктонных организмов, такие акватории выглядят как оттенки синего и голубого 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 (от темного до травянисто-зеленого), как правило, соответствуют древесной растительности, а более светлые, желто-зеленые — травянистой и кустарниковой. Однако, разнообразие растительности велико и по одному только оттенку не всегда можно однозначно судить о ее характере.

В умеренном климате, как хорошо известно жителям России, листопадные леса выглядят совершенно по-разному летом и осенью, когда деревья меняют цвет листвы и сбрасывают их. В сухом тропическом и субтропическом климате растительность также меняется в течение года (многие деревья сбрасывают листья в сухой сезон), древесная растительность может приобретать бурые оттенки в середине и конце сухого сезона.

Открытый грунт, в зависимости от его состава и характера, отображается различными оттенками красного и фуксинового (magenta). Частично покрытые растительностью участки (имеющие несомкнутый растительный покров) могут иметь различные промежуточные оттенки (розоватые, бурые и пр.). Так, травянистая растительность на самых разных широтах также может выгорать в сухой сезон, если он имеется. В указанном цветовом синтезе на космических снимках она может приобретать различные красные и бурые оттенки (в значительной мере за счет того, что в спектре преобладает сигнал от открытых участков грунта). По той же причине красноватые и светло-коричневые оттенки может иметь и тундровая растительность.

Рассмотрите внимательно и проанализируйте данные спутниковые изображения. По очертаниям местности и характеру растительности попробуйте определить, какую горную страну Вы на нем видите.



На основании анализа изображения составьте ее краткое описание данной части горной страны по следующему плану.

- Если можете, укажите название и территориальное расположение горной страны. (2 балла)
- Определите ее высотность (высокие, средние, низкие горы). Обоснуйте. (2 балла)
- Сделайте предположение о происхождении этих гор (тектонические, вулканические, эрозионные). Обоснуйте Ваше предположение. (2 балла)
- Кратко опишите растительность данной части этой горной страны и прилегающих территорий в пределах представленных снимков. Описывайте отдельно растительность для разных участков территории, привязывая ее к положению

в рельефе и к другим особенностям местности. Поясните, по каким дешифровочным признакам и дополнительным данным Вы сделали заключение о характере растительности. (10 баллов)

- Объясните описанные Вами различия в растительности на разных участках. (6 баллов)
- В какой климатической зоне (зонах) расположена данная горная страна? Почему Вы так считаете? (4 балла)

В скобочках указано максимальное количество баллов, начисляемых за соответствующий пункт описания. За характеристики, не подкрепленные соответствующим обоснованием, баллы не начисляются. Баллы также могут быть снижены за излишне краткое и нелогичное обоснование, фактические ошибки и логические провалы в рассуждении.

Ответ:

1. Название горной страны: Анды (1 балл).
Расположение: вдоль западного побережья Южной Америки, на снимке — территория Перу (1 балл); балл начисляется даже, если конкретная страна не была упомянута.
2. Это высокие горы. (1 балл) На это указывают, в частности, вершины, покрытые снегом (в тропическом климате) (1 балл)
3. Анды — горы складчатые по происхождению (1 балл). Кроме известной из курса географии информации о тектонике плит и зону субдукции (что тоже принимается как ответ), о складчатом происхождении гор свидетельствует их структура, хорошо видная на снимках: вытянутые параллельные друг другу и побережью океана хребты. (1 балл) Вулканическое происхождение не является правильным ответом, хотя вулканы в Андах, конечно есть. Эрозионное происхождение также неверный ответ, хотя эрозионные процессы, как везде в горах, активны.
4. В описании растительности оцениваются следующие моменты (засчитывается любое упоминание, даже косвенное и даже в косноязычной формулировке).
 - Отмечено различие характера растительности на равнине и в горах. (1 балл)
 - Упомянут нивальный пояс (ледники и снежники), в том числе в разделе про высотность (1 балл)
 - Упомянут горно-пустынный (безлесый) пояс, здесь называемый пуна (puna); в качестве верных принимаются любые описания того, что в горах выше леса, но ниже снегов есть зона с травянистой и кустарниковой растительностью, зона с открытым грунтом или без растительности (1 балл)
 - Упомянут горно-лесной пояс, здесь называемый юнга (yungas) (1 балл); засчитывается, если участники отделяют его от лесов на равнине
 - Отмечены различия в растительности западного и восточного макросклонов горной страны и территорий к западу (пустыня Сечура) и востоку (влажный тропический лес бассейна Амазонки) от гор. (1 балл)
 - Отмечены отличия растительности вдоль рек, в том числе упомянуты сельхозугодья на месте лесов (1 балл)

Каждый из 6-ти выводов о растительности должен быть обоснован дешифровочными признаками. В качестве таковых учитываются как спектральные ха-

рактеристики (цвета различных частей изображения), так и структуру участков территории, их привязка к объектам и прочие топологические характеристики. Наличие обоснования по дешифровочным признакам добавляет по одному баллу к каждому подпункту выше.

5. Оценивается каждое верное указание на факторы, влияющие на распределение растительности, в частности:

- Отмечена связь растительности с рельефом. (1 балл)
- Отдельно — если описана высотная поясность, за которую считается любое описание изменений растительности с высотой. (1 балл)
- Еще один балл — за употребление самого термина «высотная поясность». (1 балл)
- Указана хозяйственная деятельность человека как фактор. (1 балл)
- Указана роль холодного Перуанского течения, из-за которого западнее гор на побережье находится пустыня. (1 балл).

За данный пункт начисляется не больше 4-х баллов. Если участник упомянул все подпункты, то все равно начисляется 4 балла (рельеф не учитывается отдельно, если описана и названа высотная поясность).

6. Упомянут тропический пояс. (1 балл)

Упомянут субэкваториальный пояс к востоку от гор. (1 балл)

Упомянуто про горный климат и/или зону высотной поясности. (1 балл)

Указано на разную влажность климата с разных сторон гор из-за холодного Перуанского течения. (1 балл)

Задача III.1.2.4. (20 баллов)

На приведенных ниже спутниковых изображениях Вы видите участок территории в западной части острова Мадагаскар. Эти изображения земной поверхности получены со спутников Sentinel-2. Они не являются единовременным снимком, а показывают для каждого пикселя медианное значение отражающей способности данного участка земной поверхности со всех (относительно) безоблачных снимков за определенный месяц 2020 года. Обработка сделана с помощью онлайн сервиса Google Earth Engine (<https://earthengine.google.com/>). Алгоритм фильтрации облачности не идеален: на некоторых снимках (мартовском) заметны остатки теней от облаков. 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. Различные типы земной поверхности (растительного и почвенного покрова) выглядят на этих снимках примерно так же, как и на снимках Landsat. Можно воспользоваться описанием из предыдущей задачи. Рассмотрите внимательно и проанализируйте данные спутниковые изображения. Изучите, как меняется характер растительности данной местности в течение года.

Март 2020:



Май 2020:



Август 2020:



Октябрь 2020:



На основании анализа изображения составьте ее краткое описание данной местности по следующему плану. Поясните, по каким дешифровочным признакам и дополнительным данным Вы сделали каждое из Ваших заключений о данной местности.

- Опишите рельеф данной местности. (2 балла)
- Опишите растительность в различных участках местности в пределах снимков. (6 баллов)
- Укажите, какой тип (типы) растительности, по вашему мнению, является естественным(и) природным(и) для данной территории, а какой (какие) возник(ли) в результате хозяйственной деятельности человека. (4 балла)
- Кратко охарактеризуйте климат данной территории. Предположите, как меняется погода в течение года. Обоснуйте Ваши предположения. (4 балла)
- Предположите, какая экономическая (хозяйственная) деятельность ведется на данной территории. Какие, предположительно, источники дохода могут иметь местные жители. Обоснуйте Ваши предположения. (2 балла)

- Предположите, с чем может быть связана очевидная прямолинейность границ между различными типами растительного покрова. (2 балла)

В скобках указано максимальное количество баллов, начисляемых за соответствующий пункт описания. За характеристики и предположения, не подкрепленные соответствующим обоснованием, баллы не начисляются. Баллы также могут быть снижены за излишне краткое и нелогичное обоснование, фактические ошибки и логические провалы в рассуждении.

Ответ:

1. Рельеф данной местности равнинный. (1 балл)

Обоснование: на плоский рельеф указывает отсутствие теней от гор или врезанных ложбин; явное отсутствие связанных с рельефом границ типов растительности (все они либо антропогенные, либо связаны с водотоками); долина небольшого водотока (возможно, временного) в центральной части снимка неглубокая и не врезанная, частично занята той же растительностью, что и вокруг. (1 балл)

Обоснование отсутствует или неверное. (0 баллов)

2. В описании растительности оцениваются явные указания на следующие типы растительности. При описании дешифровочных признаков важен акцент на сезонной динамике.

- Сухой (частично листопадный) тропический лес в северо-восточной части снимка. (1 балл) Дешифровочные признаки: ярко-зеленый весной (точнее, местной осенью, в начале сухого сезона), но все равно темнее травянистой растительности и посевов на полях; существенно темнеет к маю-июню (там это зима, разгар сухого сезона); к концу сухого сезона (август-октябрь) местами буреет: это часть деревьев теряет листву (зеленый цвет сохраняется, так как часть деревьев — вечнозеленые); возможно, бурые пятна маркируют более нарушенные человеком участки. (Перечисление хотя бы части признаков — 1 балл, без объяснений — 0 баллов.)
- Обрабатываемые поля и пастбища (могут использоваться для посевов во влажный сезон и для выпаса в сухой) (1 балл.) Дешифровочные признаки: в марте, после сезона дождей выглядят желто-зелеными, светлее леса, или розовыми (если поля распаханы под посев или недавно убраны); в мае только часть полей остаются зелеными (с посевами), фиолетовые пятна в юго-восточном углу снимка — палы (выжигания на пастбищах), к концу сухого сезона растительность на полях и пастбищах выгорает от солнца — преобладают красноватые и бурые тона от открытого грунта и сухой растительности (здесь отличия от леса очень заметны); обрабатываемые поля, как правило прямоугольной формы, с четкими границами; пастбища — без четких границ. (Перечисление хотя бы части признаков — 1 балл, без объяснений — 0 баллов.)
- Участки недавно вырубленного и, как правило, сожженного леса (подсечно-огневое земледелие). (1 балл) Дешифровочные признаки: яркие фуксиновые пятна относительно небольшого размера, на границе с еще сохранившимся лесом или даже отдельные в глубине массива; к концу сухого сезона становятся неотличимы от старых полей. (Перечисление хотя бы части признаков — 1 балл, без объяснений — 0 баллов.)

3. Естественная растительность — сухие тропические леса (1 балл), поля и пастбища созданы человеком на месте лесов, местами зарастают отдельными дере-

вьями и кустами (вторичное зарастание) (1 балл).

Объяснение: свидетельства антропогенного характера полей — прямоугольные границы; различная сезонная динамика на соседних участках (разное использование), в то время как динамика всего лесного массива примерно одна и та же на крупных участках; травяные палы — также свидетельства искусственного происхождения.

(Перечисление хотя бы части признаков — 1 балл, без объяснений — 0 баллов.)

Еще один балл начисляется, если участник обратит внимание, что лесной массив тоже подвергается воздействию человека: достаточно хорошо различимая сетка просек — свидетельство ведения в прошлом лесного хозяйства.

4. Климат — сухой тропический (1 балл).

Сухой сезон с марта по октябрь и влажный с ноября по март. Точность до месяца не важна. (1 балл).

Обоснование: яркая зелень на мартовских снимках — свидетельство недавнего влажного сезона (1 балл); постепенное выгорание растительности на серии снимков — свидетельство сухого сезона (1 балл).

5. Земледелие (1 балл)

Скотоводство (1 балл)

Лесное хозяйство или туризм (лесной массив — природный резерват) (1 балл).

Хотя в целом два балла за пункт, допускается выставление дополнительного балла за догадку о туризме.

6. Прямолинейные границы обрабатываемых полей. (1 балл)

Границы охраняемой природной территории (резервата), где запрещают вырубать лес под поля и пастбища (все равно изредка вырубают, но в целом эти границы видно, что соблюдаются). (1 балл)

Задача III.1.2.5. (14 баллов)

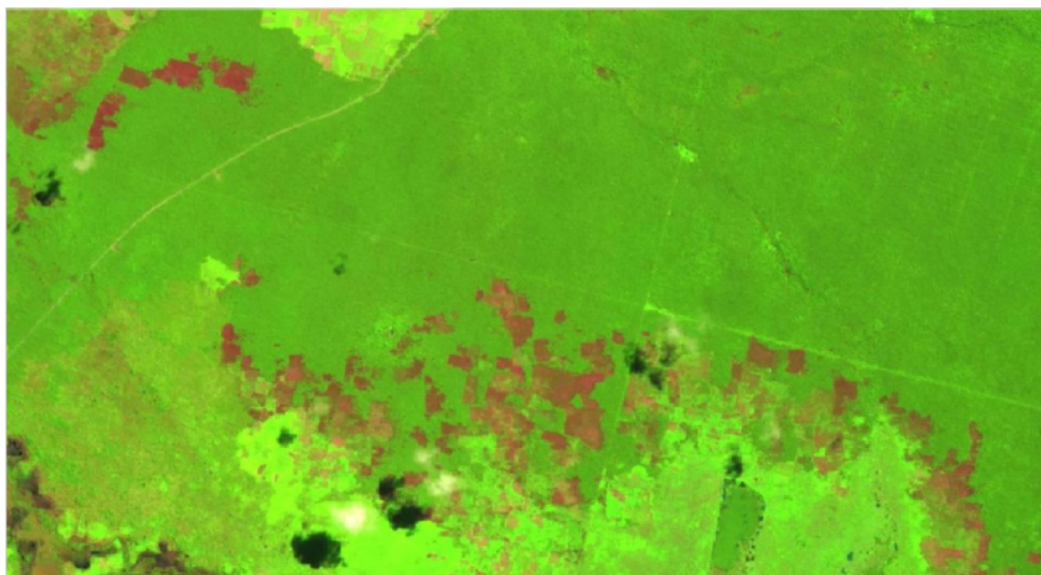
На приведенных ниже спутниковых изображениях Вы видите тот же участок территории в западной части острова Мадагаскар, что и в предыдущей задаче.

Эти изображения земной поверхности также получены со спутников Sentinel-2 и тоже являются композитами, полученными с помощью онлайн сервиса Google Earth Engine (<https://earthengine.google.com/>).

Все, написанное про такие композиты в предыдущей задаче, в полной мере относится и к данным изображениям. Однако, эти композиты получены путем обработки снимков за март месяц разных годов. С их помощью можно проследить динамику изменений растительности за последние несколько лет.

Рассмотрите внимательно данные спутниковые изображения. Проанализируйте, как меняется характер растительности данной местности за последние годы.

Март 2016 года:



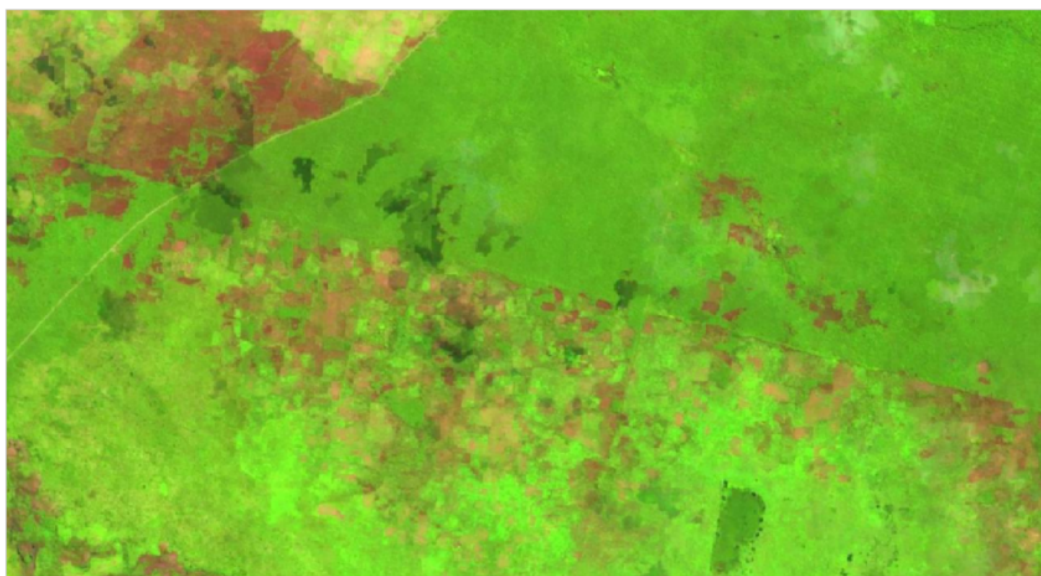
Март 2017 года:



Март 2018 года:



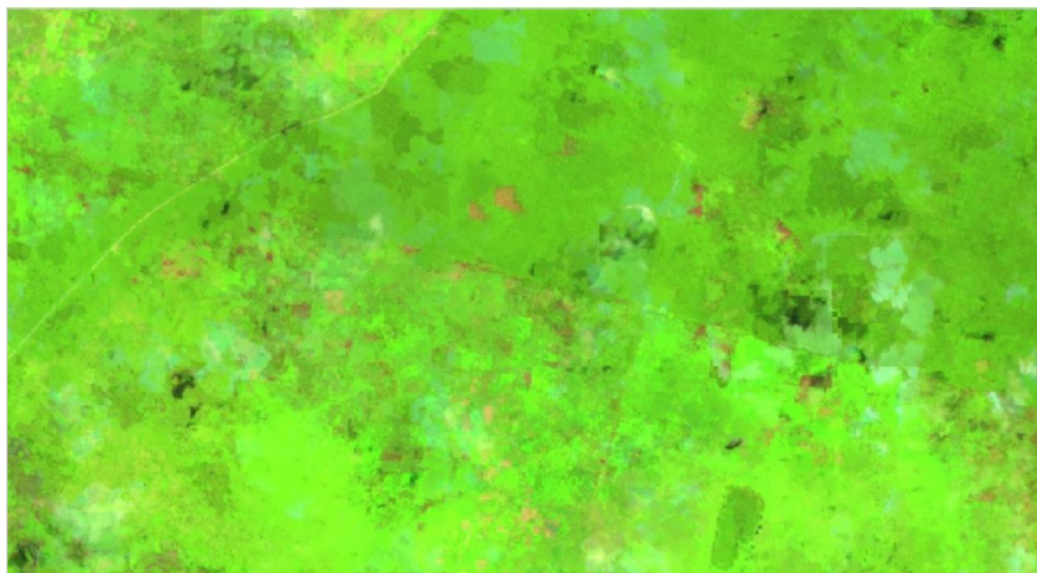
Март 2019 года:



Март 2020 года:



Март 2021 года (частично облачный):



- Опишите произошедшие изменения. Поясните, по каким дешифровочным признакам и дополнительным данным Вы сделали Ваши выводы. (6 баллов)
- Что, по Вашему мнению, является причиной / «двигателем» этих изменений? (4 балла)
- К каким проблемам это может привести? (4 балла)

В скобках указано максимальное количество баллов, начисляемых за соответствующий вопрос. За характеристики и предположения, не подкрепленные соответствующим обоснованием, баллы не начисляются. Баллы также могут быть снижены за излишне краткое и нелогичное обоснование, фактические ошибки и логические провалы в рассуждении.

Ответ:

1. Баллы начисляются за указание следующих изменений за последние годы, отраженных на снимках.

- Вырубка / сведение леса (1 балл)
- Создание полей / посадка культур на этих местах, расширение сельхозугодий (1 балл)
- Приостановка / замедление / прекращение вырубок/расчисток при достижении ими границы природного резервата / создание резервата (ООПТ — особо охраняемой природной территории) (1 балл)
- Посадка леса — не засчитывается, ее не было. (0 баллов)

Оцениваются дешифровочные признаки, по которым сделаны выводы:

- Появление красных / фуксиновых участков правильной геометрической формы на месте бывшего леса (1 балл)
- Травянистая растительность (посевы) на месте вырубленных участков на следующий год (1 балл)
- Ограничение распространения расчисток и полей видимой прямолинейной границей (границей резервата) (1 балл)

2. Баллы начисляются / не начисляются за указание на следующие процессы, выступающие в качестве движущей силы изменений растительности.

- Расширение сельхозугодий (1 балл)
- Подсечно-огневое земледелие, примитивная/экстенсивная модель сельского хозяйства (дополнительно 1 балл, даже если сельхозугодья уже указаны)
- Потребность в пропитание / собственное потребление местных жителей / натуральное хозяйство (1 балл)
- Значительный прирост населения (1 балл)
- Потребность в древесине не приводит здесь к полному сведению лесов, здесь не ведутся промышленные лесозаготовки. (0 баллов)
- «Рыночные интересы», промышленное производство товаров для рынка — не оказывает серьезного влияния — это почти полностью натуральное хозяйство (0 баллов)
- Засуха сама по себе не является «двигателем» изменений, скорее следствием (0 баллов)
- «Деятельность человека» в такой общей формулировке не засчитывается (0 баллов)
- Природные климатические изменения не являются сами по себе «двигателем» наблюдаемых процессов — за такой достаточно короткий период наблюдать значимое влияние климатических изменений невозможно (0 баллов)

3. Баллы начисляются / не начисляются за указание на следующие возможные проблемы.

- Истощение земель / разрушение почвенного покрова (1 балл)
- Усиление засушливости местного климата / опустынивание (1 балл)
- Обезлесивание / исчезновение лесов (1 балл)
- Парниковый эффект, глобальное потепление (1 балл)
- Потеря мест обитания животных и растений / деградация биологического разнообразия (1 балл)
- «Проблемы с экологией», сформулированные в общем виде не засчитываются (0 баллов)

- Снижение выработки кислорода и озона — не засчитываем, это не так. (0 баллов)

Всего 4 балла за пункт, но допускается начисление 5 баллов за выдающиеся ответы, указывающие на все перечисленные проблемы.

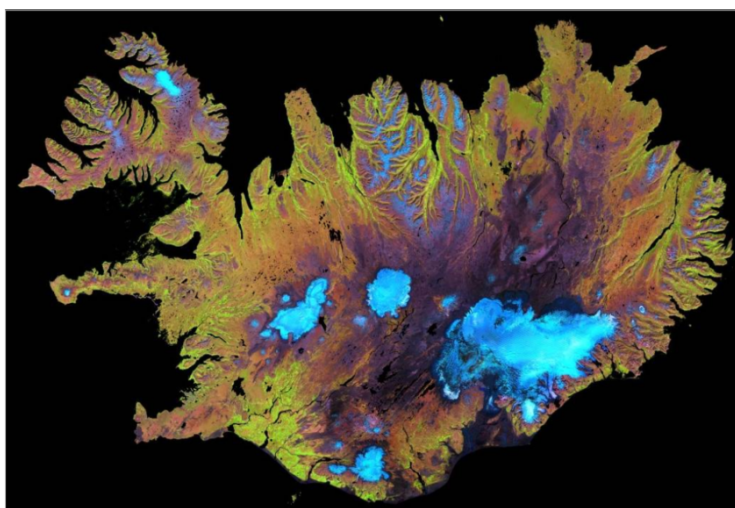
Задача III.1.2.6. (10 баллов)

Рассмотрите внимательно приведенные ниже спутниковые изображения различных островов/архипелагов. Определите, к какому из типов относится каждый остров по своему происхождению: (1) материковый, (2) вулканический, (3) коралловый, (4) искусственный. 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, острова двух или более типов, выберите тот, который занимает большую площадь в пределах снимка и расположен 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), обработанные 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>). Описание того, как выглядят на этих снимках различные типы земной поверхности (растительного и почвенного покрова, вода и пр.), возьмите из Задачи 3.

Изображение 1:



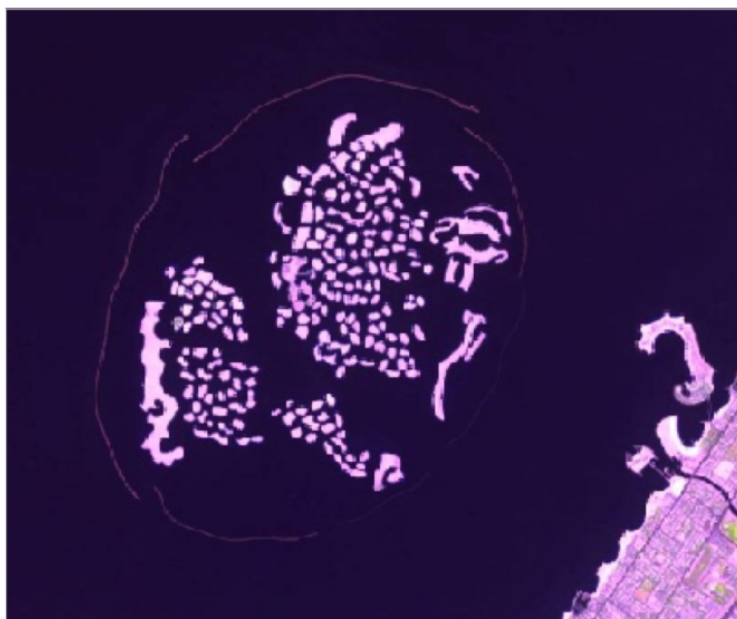
Изображение 2:



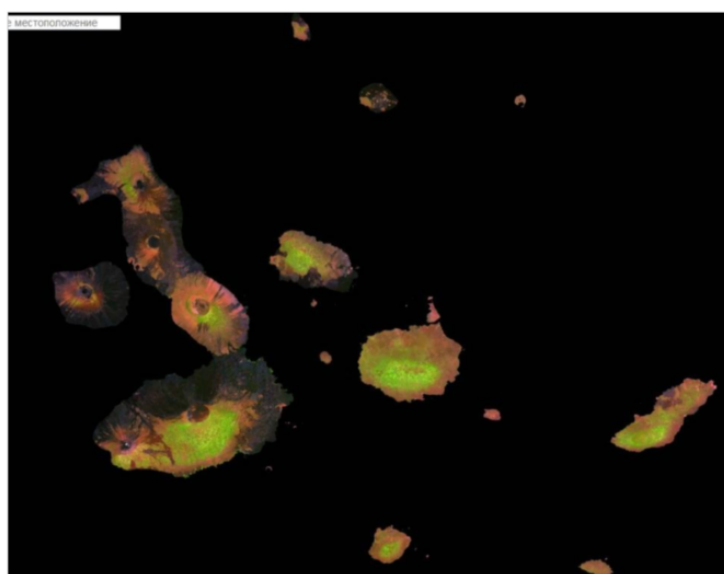
Изображение 3:



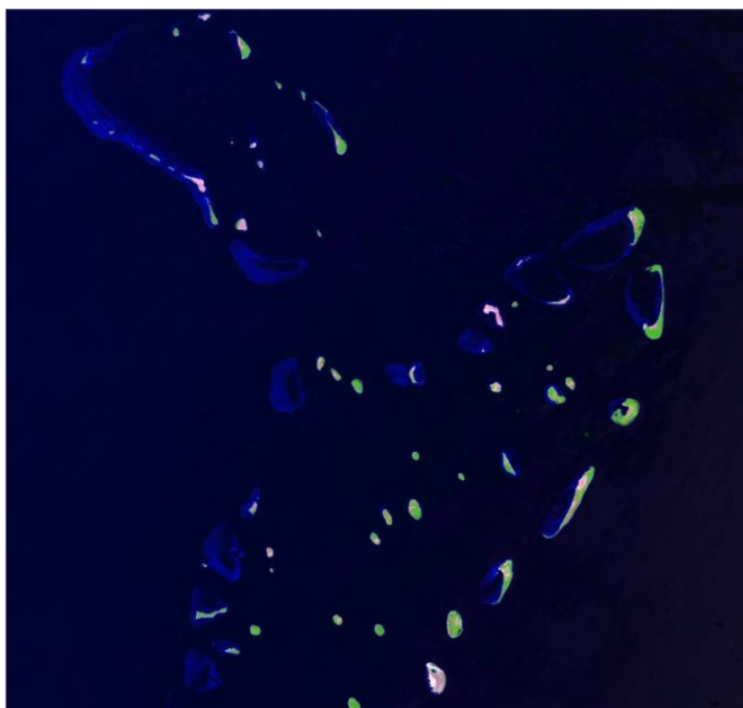
Изображение 4:



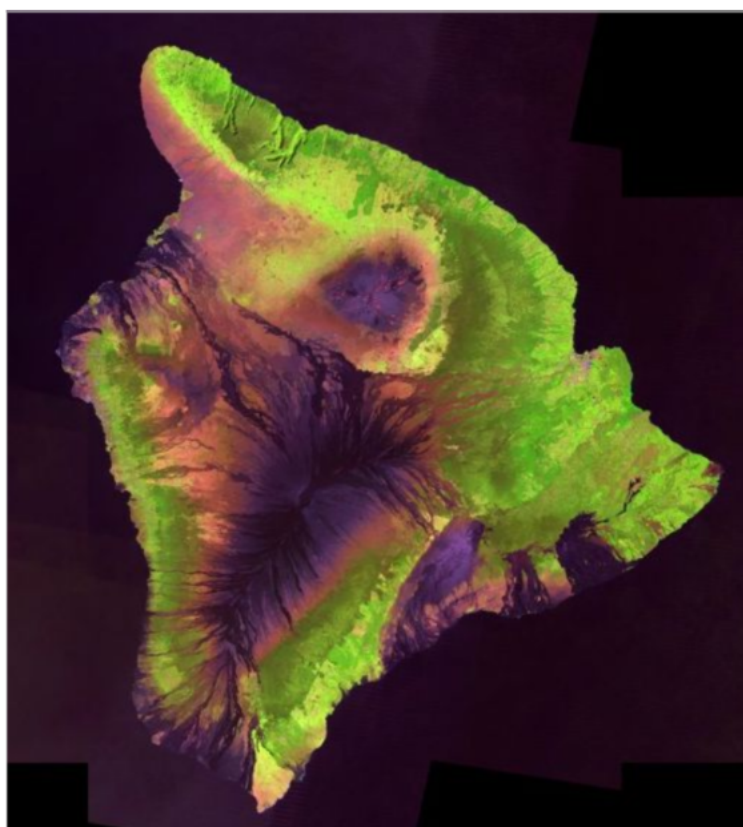
Изображение 5:



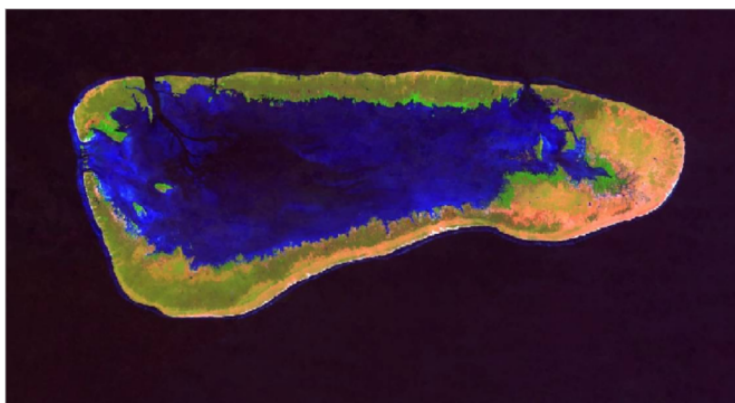
Изображение 6:



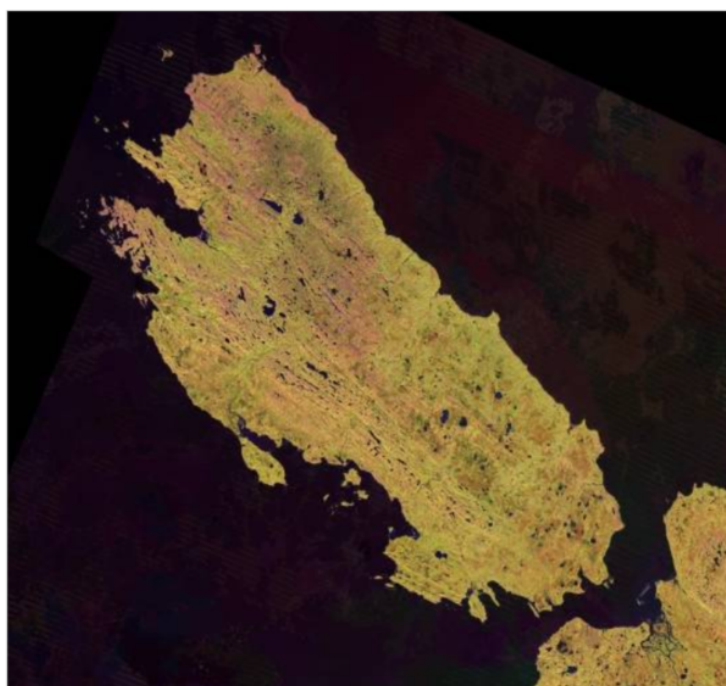
Изображение 7:



Изображение 8:



Изображение 9:



Ответ:

Спутниковые изображе-	материковый	вулканический	коралловый	искусственный
Изображение 1	☐	☒	☐	☐
Изображение 2	☒	☐	☐	☐
Изображение 3	☒	☐	☐	☐
Изображение 4	☐	☐	☐	☒
Изображение 5	☐	☒	☐	☐
Изображение 6	☐	☐	☒	☐
Изображение 7	☐	☒	☐	☐
Изображение 8	☐	☐	☒	☐
Изображение 9	☒	☐	☐	☐