

Solutions designed by:

Daniel Rozario

Md. Abdullah Al Mahfuz

Md Abu Raihan

Bornita Ghosh

Rafiul Islam Rafi

Atia Sanjida Saba

Problem Solving

Problem-05: Shortest Increasing Path

(Problem Source: <https://codeforces.com/problemset/problem/2147/A>)

You are at $(0, 0)$ in a rectangular grid and want to go to (x, y) .

In order to do so, you are allowed to perform a sequence of steps.

Each step consists of moving a positive integer amount of length in the positive direction of either the x or the y axis.

The first step must be along the x axis, the second along the y axis, the third along the x axis, and so on. Formally, if we number steps from one in the order they are done, then odd-numbered steps must be along the x axis and even-numbered steps must be along the y axis.

Additionally, each step must have a length **strictly greater** than the length of the previous one.

Output the minimum number of steps needed to reach (x, y) , or -1 if it is impossible.

Input

Each test contains multiple test cases. The first line contains the number of test cases t ($1 \leq t \leq 10^4$). The description of the test cases follows.

The first and only line of each case contains two integers x and y ($1 \leq x, y \leq 10^9$).

Output

For each test case, output the minimum number of steps to reach (x, y) or -1 if it is impossible.

Input

Each test contains multiple test cases. The first line contains the number of test cases t ($1 \leq t \leq 10^4$).

The description of the test cases follows.

The first and only line of each case contains two integers x and y ($1 \leq x, y \leq 10^9$).

Output

For each test case, output the minimum number of steps to reach (x, y) or -1 if it is impossible.

Example

input

```
10
1 2
5 6
4 2
1 1
2 1
3 3
5 1
5 4
752 18572
95152 2322
```

output

```
2
2
3
-1
-1
-1
-1
-1
2
3
```

The Solution

```
1 #include <bits/stdc++.h>
2 using namespace std;
3
4 int main() {
5     int t;
6     cin>>t;
7     while(t--)
8     {
9         long x, y;
10        cin>>x;
11        cin>>y;
12
13        if(x==y || x==y+1 || y==1)
14        {
15            cout<<-1<<endl;
16        }
17        else if(x<y)
18        {
19            cout<<2<<endl;
20        }
21        else cout<<3;
22    }
23
24 }
25 }
```

Problem-06: Shift Sort

(Problem Source: <https://codeforces.com/problemset/problem/2140/A>)

You are given a binary string* s of length n and you are allowed to perform the following operation any number of times (including zero):

- Choose 3 indices $1 \leq i < j < k \leq n$ and right shift or left shift the values on s_i, s_j, s_k cyclically.

For the binary string 110110, if we choose $i = 1, j = 2, k = 3$ and perform a right shift cyclically, the string becomes 011110; if we choose $i = 4, j = 5, k = 6$ and perform a left shift cyclically, the string becomes 110101.

Determine the minimum number of operations required to sort the given binary string.

*A *binary string* is a string that consists only of the characters 0 and 1.

Input

Each test contains multiple test cases. The first line contains the number of test cases t ($1 \leq t \leq 100$). The description of the test cases follows.

The first line of each test case contains a single integer n ($3 \leq n \leq 100$) — the length of the string.

The second line contains a binary string s of length n .

Output

For each test case, output a single integer — the minimum number of operations required to sort the given binary string.

Input

Each test contains multiple test cases. The first line contains the number of test cases t ($1 \leq t \leq 100$). The description of the test cases follows.

The first line of each test case contains a single integer n ($3 \leq n \leq 100$) — the length of the string.

The second line contains a binary string s of length n .

Output

For each test case, output a single integer — the minimum number of operations required to sort the given binary string.

Example

input

```
4  
3  
001  
4  
0110  
6  
110100  
6  
101011
```

output

```
0  
1  
2  
1
```

The Solution

```
1 #include <iostream>
2 using namespace std;
3 int main() {
4     int t;
5     cin >> t;
6     while (t--) {
7         int n;
8         cin >> n;
9         string s;
10        cin >> s;
11        int z = 0;
12        for (int i = 0; i < n; i++) if (s[i] == '0') z++;
13        int cnt = 0;
14        for (int i = 0; i < z; i++) if (s[i] == '1') cnt++;
15        cout << cnt << "\n";
16    }
17    return 0;
18 }
19 }
```

Problem-07: Only One Digit

(Problem Source: <https://codeforces.com/problemset/problem/2126/A>)

You are given an integer x . You need to find the smallest non-negative integer y such that the numbers x and y share at least one common digit. In other words, there must exist a decimal digit d that appears in both the representation of the number x and the number y .

Input

The first line contains an integer t ($1 \leq t \leq 1000$) – the number of test cases.

The first line of each test case contains one integer x ($1 \leq x \leq 1000$).

Output

For each test case, output one integer y – the minimum non-negative number that satisfies the condition.

Input

The first line contains an integer t ($1 \leq t \leq 1000$) — the number of test cases.

The first line of each test case contains one integer x ($1 \leq x \leq 1000$).

Output

For each test case, output one integer y — the minimum non-negative number that satisfies the condition.

Example

input

```
5  
6  
96  
78  
122  
696
```

output

```
6  
6  
7  
1  
6
```

The Solution

```
1 #include <bits/stdc++.h>
2 using namespace std;
3 int main() {
4     int t;
5     cin >> t;
6     while (t--) {
7
8         string s;
9         cin >> s;
10        sort(s.begin(), s.end());
11        cout << s[0] << endl;
12    }
13    return 0;
14 }
```

Problem-08

You are given an array of characters, and each character can only be one of theses: 'b', 'o', or 'd'.

Suppose, a word "bod" is written on a mirror. When looking at the *other side* of the mirror, the characters appear reversed and each character becomes its mirrored version.

Given the input array (which always contains any or all from the characters 'b', 'o' or 'd'), determine what the string looks like from the other side of the mirror.

Example:

Input	Output
Test Case: 4	
bod	bod
bdo	oobd
obbbo	oddoo
ddddd	bbbbb

The Solution

```
1 #include <iostream>
2 using namespace std;
3
4 int main() {
5     int t;
6     cout<<"Test Case: ";
7     cin>>t;
8     while(t--) {
9         string s;
10        cin >> s;
11
12        string ans = "";
13
14        for (int i = s.size() - 1; i >= 0; i--) {
15
16            if (s[i] == 'b') {
17                ans = ans + 'd';
18            }
19            else if (s[i] == 'd') {
20                ans = ans + 'b';
21            }
22            else {
23                ans = ans + 'o';
24            }
25        }
26
27        cout << ans;
28    } return 0;
29 }
```

Problem-09

You are given an array of **n** integers.

You will also be given an integer **target**.

Your task is to check whether **any two elements** in the array add up to the target value.

If such a pair exists → print **1**

Otherwise → print **0**

Example:

Input	Output
Test Case: 3	
Array Size: 5 Array Elements: 2 7 4 1 8 Target: 9	1
Array Size: 4 Array Elements: 1 2 3 4 Target: 8	0
Array Size: 5 Array Elements: 0 -1 2 -3 1 Target: -2	1

The Solution

```
1 #include <iostream>
2 using namespace std;
3
4 int main() {
5     int t;
6     cout<<"Test Case: ";
7     cin>>t;
8     while(t--) {
9         int n;
10        cout<<"Array size: ";
11        cin >> n;
12
13        int arr[n];
14        cout<<"Array elements: ";
15        for (int i = 0; i < n; i++) {
16            cin >> arr[i];
17        }
18
19        int target;
20        cout<<"Target: ";
21        cin >> target;
22
23        int found = 0;
24
25        for (int i = 0; i < n; i++) {
26            for (int j = i + 1; j < n; j++) {
27                if (arr[i] + arr[j] == target) {
28                    found = 1;
29                }
30            }
31        }
32
33        cout << found<<endl;
34    }
35
36    return 0;
37}
38
```