

Problem Solving

Problem-01: Be Positive

(Problem Source: <https://codeforces.com/problemset/problem/2149/A>)

Given an array a of n elements, where each element is equal to -1 , 0 , or 1 .
In one operation,
you can choose an index i and increase a_i by 1 (that is, assign $a_i := a_i + 1$).
Operations can be performed any number of times, choosing any indices.

The goal is to make the product of all elements in the array strictly positive with the minimum number of operations, that is, $a_1 \cdot a_2 \cdot a_3 \cdot \dots \cdot a_n > 0$.

Find the minimum number of operations.

It is guaranteed that this is always possible.

Input

Each test contains multiple test cases.

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

The description of the test cases follows:

The first line of each test case contains one integer n ($1 \leq n \leq 8$) — the length of the array a .

The second line contains n integers a_1, a_2, \dots, a_n , where $-1 \leq a_i \leq 1$ — the elements of the array a .

Output

For each test case, output one integer — the minimum number of operations required to make the product of the elements in the array strictly positive.

Example

input

3

3

-1 0 1

4

-1 -1 0 1

5

-1 -1 -1 0 0

output

3

1

4

The Solution

```
1  #include <iostream>
2  using namespace std;
3
4  int main() {
5      int t;
6      cin >> t;
7      while (t--> 0) {
8          int n,e;
9          cin >> n;
10         int result = 0, temp = 0;
11         for (int i = 0; i < n; i++)
12         {
13             cin >> e;
14             if (e == 0) result++;
15             else if (e == -1) temp++;
16         }
17         if (temp % 2 == 1)
18         {
19             result = result + 2;
20         }
21         cout << result << "\n";
22     }
23     return 0;
24 }
25
```

Problem-02: Sublime Sequence

(Problem Source: <https://codeforces.com/problemset/problem/2149/A>)

Farmer John has an integer x .

He creates a sequence of length n by alternating integers x and $-x$, starting with x .

For example, if $n=5$, the sequence looks like: $x, -x, x, -x, x$.

He asks you to find the sum of all integers in the sequence.

Input

The first line contains an integer t ($1 \leq t \leq 100$) — the number of test cases.

The only line of input for each test case is two integers x and n ($1 \leq x, n \leq 10$).

Output

For each test case, output the sum of all integers in the sequence.

Problem-02: Sublime Sequence

Input

The first line contains an integer t ($1 \leq t \leq 100$) — the number of test cases.

The only line of input for each test case is two integers x and n ($1 \leq x, n \leq 10$).

Output

For each test case, output the sum of all integers in the sequence.

Example

input	
4	
1	4
2	5
3	6
4	7

output	
0	
2	
0	
4	

The Solution


```
1  #include <iostream>
2  using namespace std;
3
4  int main() {
5      int t;
6      cin >> t;
7      while (t--> 0) {
8          int x, n;
9          cin >> x >> n;
10         if (n % 2 == 0) cout << 0 << endl;
11         else cout << x << endl;
12     }
13     return 0;
14 }
15
```

Problem-03: Maple and Multiplication

(Problem Source: <https://codeforces.com/problemset/problem/2139/A>)

Maple has two positive integers a and b . She may perform the following operation any number of times (possibly zero) to make a equal to b :

- Choose any positive integer x , and multiply either a or b by x .

Your task is to determine the minimum number of operations required to make a equal to b . It can be proven that this is always possible.

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 and only line of each test case contains two positive integers a and b ($1 \leq a, b \leq 1000$) — the numbers Maple currently has.

Output

For each test case, output a single integer representing the minimum number of operations Maple needs to make a equal to b .

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 and only line of each test case contains two positive integers a and b ($1 \leq a, b \leq 1000$) — the numbers Maple currently has.

Output

For each test case, output a single integer representing the minimum number of operations Maple needs to make a equal to b .

Example

input
3
1 2
10 3
1000 1000

output
1
2
0

The Solution

```
1  #include <iostream>
2  using namespace std;
3
4  int main() {
5      int t;
6      cin >> t;
7      while (t--) {
8          int a, b;
9          cin >> a >> b;
10         if (a == b) cout << 0 << endl;
11         else if (a % b == 0 || b % a == 0) cout << 1 << endl;
12         else cout << 2 << endl;
13     }
14     return 0;
15 }
16
```

Problem-04: Binary Imbalance

(Problem Source: <https://codeforces.com/problemset/problem/1902/A>)

You are given a string s , consisting only of characters '0' and/or '1'.

In one operation,

you choose a position i from 1 to $|s|-1$, where $|s|$ is the current length of string s .

Then you insert a character between the i -th and the $(i+1)$ -st characters of s .

If $s_i = s_{i+1}$, you insert '1'. If $s_i \neq s_{i+1}$, you insert '0'.

Is it possible to make the number of zeroes in the string strictly greater than the number of ones, using any number of operations (possibly, none)?

Input

The first line contains a single integer t ($1 \leq t \leq 100$) — the number of testcases.

The first line of each testcase contains an integer n ($1 \leq n \leq 100$).

The second line contains a string s of length exactly n , consisting only of characters '0' and/or '1'.

Output

For each testcase, print "YES" if it's possible to make the number of zeroes in s strictly greater than the number of ones, using any number of operations (possibly, none). Otherwise, print "NO".

Input

The first line contains a single integer t ($1 \leq t \leq 100$) — the number of testcases.

The first line of each testcase contains an integer n ($1 \leq n \leq 100$).

The second line contains a string s of length exactly n , consisting only of characters '0' and/or '1'.

Output

For each testcase, print "YES" if it's possible to make the number of zeroes in s strictly greater than the number of ones, using any number of operations (possibly, none). Otherwise, print "NO".

Example

input
3
2
00
2
11
2
10

output
YES
NO
YES

The Solution

Solution-1

```
1  #include <bits/stdc++.h>
2  using namespace std;
3  int main() {
4      int t; cin >> t;
5      while (t--) {
6          int n; string s;
7          cin >> n >> s;
8          cout << (s.find('0') != string::npos ? "YES\n" : "NO\n");
9      }
10 }
11
```

Solution-2

```
1  #include <bits/stdc++.h>
2  using namespace std;
3  int main() {
4      int t; cin >> t;
5      while (t--) {
6          int n, temp=0;
7          cin >> n;
8          char s[n];
9          for(int i=0; i<n; i++)
10             {
11                 cin>>s[i];
12                 if(s[i]=='0')
13                     {
14                         temp++;
15                     }
16             }
17             if(temp>0) cout<<"Yes";
18             else cout<<"No";
19         }
20     }
```