111-2 Data Structure

Homework 1

Question 1 (80%)

You are given a **0-indexed** array leaves of size n representing the positions of several lotus leaves on a river. leaves[i] can be -1, 0, or 1 where:

- -1 represents there is a female frog on the i^{th} leaf.
- 0 indicates there is a broken leaf (with a big hole on it) at the i^{th} position.
- 1 indicates there is a male frog on the i^{th} leaf.

Each male frog wants to make **a single jump** over broken leaves without falling into the river to find a female frog from position i to position j such that:

- 0 <= i, j <= n-1
- The male frog jumps over broken leaves **only**. Formally, for all k where min(i,j) < k < max(i,j), leaves[k] == 0.

Please print the **maximum** number of the broken leaves that can be jumped over by a male frog. In case it is **impossible** for a male frog to find a female frog, or there isn't any male frog on a leaf, print 0.

Input format

The first line of a test case has an integer n representing the number of the leaves. The second line contains n integers representing the status of the leaves.

Constraints

- 1 <= n <= 1000
- -1 <= leaves[i] <= 1

Output format

An integer.

Sample input 1

```
9
1 0 0 -1 0 0 0 0 1
```

Sample output 1

```
4
```

Sample input 2

```
4
0 0 1 -1
```

Sample output 2

```
0
```

Question 2 (20%)

Continuing from **Question 1**, now you are given a 2d-array leaves of size rows*cols representing the positions of several lotus leaves on a river.

Each male frog can only jump to the leaf on the same row/column where it stands, which means it **can't jump diagonally** to find a female frog.

All the other rules are the same as in Question 1.

Please print the **maximum** number of the broken leaves that can be jumped over by a male frog. In case it is **impossible** for a male frog to find a female frog, or there isn't any male frog on a leaf, print 0.

Input format

The first line of a test case has 2 integers rows, cols representing there are rows*cols leaves. The following rows lines each contains cols integers representing the status of the leaves.

Constraints

- 1 <= rows, cols <= 1000
- -1 <= leaves[i][j] <= 1

Output format

An integer.

Sample input 1

```
8 10
0 1 0 -1 0 0 1 0 0 1
1 0 0 0 0 -1 0 1 0 -1
0 0 -1 0 0 1 0 0 0 1
0 0 0 1 -1 -1 0 1 1 -1
-1 -1 0 0 0 -1 0 0 0 1
0 1 0 -1 0 0 0 0 0 1
1 0 0 0 0 -1 0 1 0 -1
0 0 0 1 -1 -1 -1 1 1 -1
```

Sample output 1

```
6
```

Sample input 2

```
5 4
1 -1 0 0
0 0 0 1
1 -1 -1 -1
-1 1 1 1
0 -1 -1 0
```

Sample output 2

```
0
```

Grading

Each question has **5 test cases**, and you'll get **0.2*the total score of the question** if you pass 1 test case of the question, for example:

- Question 1: 4/5 test cases are passed, get 4*0.2*80 = 64 points
- Question 2: 3/5 test cases are passed, get 3*0.2*20 = 12 points

and you'll get 76 points totally.

Please do not plagiarize, or you'll get 0 point.

Submission

You can only use C/C++ to write the program.

Please name your files as Q{question_id}_{student_id}, for example:

- Q1_123456.c or Q1_123456.cpp
- Q2_123456.c or Q2_123456.cpp

and then upload your files to e3.

If you have any questions, please send an e-mail to the teacher and all the TAs **via e3**.