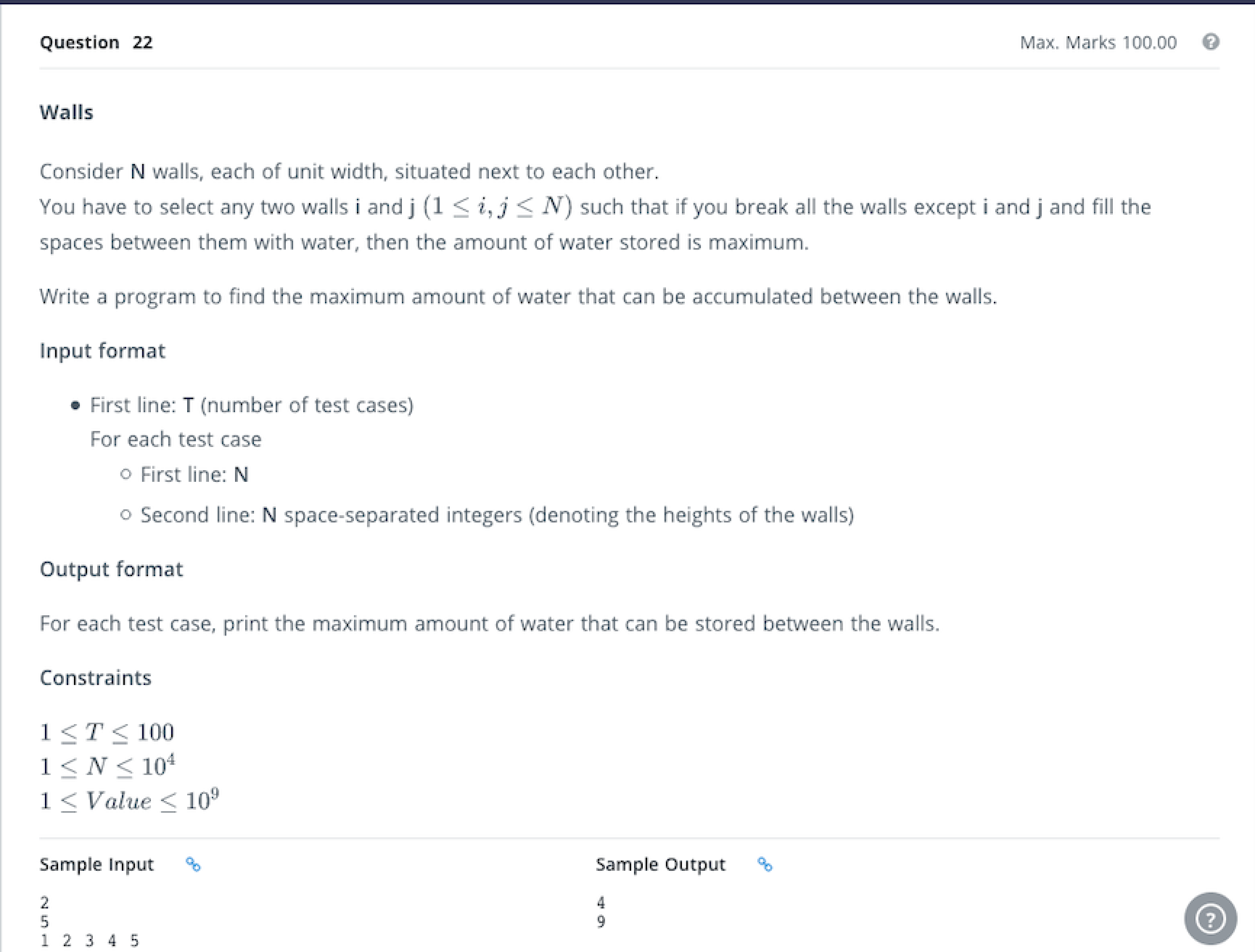


Question 22 Max. Marks 100.00

Walls

Consider N walls, each of unit width, situated next to each other.
You have to select any two walls i and j $(1 \leq i, j \leq N)$ such that if you break all the walls except i and j and fill the spaces between them with water, then the amount of water stored is maximum.

Write a program to find the maximum amount of water that can be accumulated between the walls.

Input format

- First line: T (number of test cases)
  For each test case
  - First line: N
  - Second line: N space-separated integers (denoting the heights of the walls)

Output format

For each test case, print the maximum amount of water that can be stored between the walls.

Constraints

$1 \leq T \leq 100$
$1 \leq N \leq 10^4$
$1 \leq Value \leq 10^9$

Sample Input

```
2
5
1 2 3 4 5
```

Sample Output

```
4
9
```

Question 21 Max. Marks 100.00

Ladders

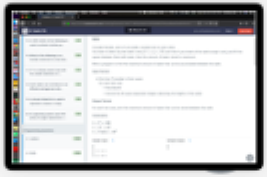
You are visiting a building of N floors. **On every floor, only one ladder of specified length is present.** If the length of the ladder is x units, you can reach x floors above from the current one. Refer to the example below for clear understanding.

> For example, if a person is on the 2^{nd} floor where a ladder of 3 units is present, he/she can reach any of 3rd/4th/5th floor as per his/her choice.

You can leave the ladder in between in order to change the ladder, but you can only start from the starting floor of the ladder.

> For example, if a person is on the 2^{nd} floor where a ladder of 3 units is present (let's denote ladder by L1), and there is a ladder on 3^{rd} floor of 3 units say L2, and a person starts from 2^{nd} floor with ladder L1, and reaches 4th floor, he cannot use the ladder L2 from 4th floor or any floor except the 3rd floor. Since the ladder L2 is present on 3rd floor, he can only start using it from that floor.

You are given Q questions. In each question, you will be given a floor number. For each question, you have to tell the least number of ladders required to reach given floor.
Initially, you are on the ground floor.



Input Format:
The first line contains an integer T , indicating the number of test cases.
For each test case:

number of ladders required to reach given floor.

Initially, you are on the ground floor.

Input Format:

The first line contains an integer T , indicating the number of test cases.

For each test case:

The first line contains an integer N, indicating number of floors in the building.

Next line contains N space separated positive integers which denote the length of ladder at each floor (First integer corresponds to ladder length on ground floor, second integer corresponds to ladder length on first floor and so on) .

Next line contains an integer Q, indicating number of questions.

Following Q lines contain an integer each, denoting the floor number for which answer is to be computed.

Output Format:

For each question, print the least number of ladders required required to reach given floor.

Answer for each question should come in a new line.

Input Constraints

$1 \leq T \leq 10$

$1 \leq N, Q \leq 10^5$

$1 \leq ladder\ length \leq N$

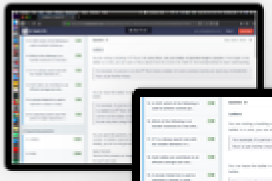
$1 \leq query\ value \leq N$

Sample Input

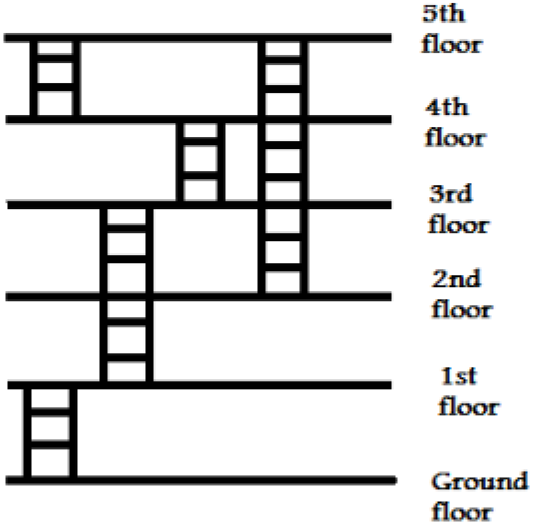
```
1
5
1 2 3 1 1
5
1
2
3
4
5
```

Sample Output

```
1
2
2
3
3
```



Explanation



We can reach 1st floor using only 1 ladder.
2nd floor can be reached using 2 ladders.
3rd floor can be reached using 2 ladders.
4th floor can be reached using 3 ladders.
5th floor can be reached using 3 ladders.

Note: Your code should be able to convert the sample input into the sample output. However, this is not enough to pass the challenge, because the code will be run on multiple test cases. Therefore, your code must solve this problem statement.

Time Limit: 2.0 sec(s) for each input file

Memory Limit: 256 MB

Source Limit: 1024 KB

Marking Scheme: Marks are awarded if any testcase passes

Allowed Languages: Bash, C, C++, C++14, Clojure, C#, D, Erlang, F#, Go, Groovy, Haskell, Java, Java 8, JavaScript(Rhino), JavaScript(Node.js), Juli (SBCL), Lua, Objective-C, OCaml, Octave, Pascal, Perl, PHP, Python, Python 3, Racket, Ruby, Rust, Scala, Swift, Swift-4.1, Visual Basic