

# The Troublesome Frog

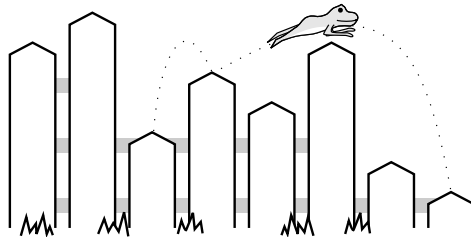
**Input File:** *frogin.txt*

**Output File:** *frogout.txt*

**Time Limit:** 1 second

Bazza and Shazza do not like bugs. They wish to clear out all the bugs on their garden fence. They come up with a brilliant idea: they buy some sugar frogs and release them near the fence, letting them eat up all the bugs.

The plan is a great success and the bug infestation is gone. But strangely, they now have a sugar frog infestation. Instead of getting rid of the frogs, Bazza and Shazza decide to set up an obstacle course and watch the frogs jump along it for their enjoyment.



The fence is a series of  $N$  fence posts of varying heights. Bazza and Shazza will select three fence posts to create the obstacle course, where the *middle* post is *strictly higher* than the other two. The frogs are to jump up from the left post to the middle post, then jump down from the middle post to the right post. The three posts do not have to be next to each other as frogs can jump over other fence posts, regardless of the height of those other posts.

The *difficulty* of an obstacle course is the height of the first jump plus the height of the second jump. The height of a jump is equal to the difference in height between its two fence posts. Your task is to help Bazza and Shazza find the most difficult obstacle course for the frogs to jump.

## Input

Your program should read from the file `frogin.txt`. The file will describe a single fence.

- The first line of input will contain one integer  $N$ : the number of fence posts.
- The next  $N$  lines will each contain one integer  $h_i$ : the height of the  $i$ th fence post.

You are guaranteed that there will be at least one valid obstacle course: that is, there will be at least one combination of three fence posts where the middle post is strictly higher than the other two.

## Output

Your program should write to the file `frogout.txt`. Your output file should contain one line with one integer: the greatest difficulty of any possible obstacle course.

**Sample Input 1**

8  
60  
70  
30  
50  
40  
60  
20  
10

**Sample Output 1**

80

**Explanation**

This case corresponds to the diagram on the previous page. In the diagram the frog is shown jumping between the third, fourth and eighth posts, for a difficulty of  $(50 - 30) + (50 - 10) = 60$ .

In fact, the most difficult course uses the third, sixth and eighth posts. The jump heights are 30 and 50, giving a total difficulty of 80.

**Sample Input 2**

5  
9  
9  
3  
4  
2

**Sample Output 2**

3

**Explanation**

The only valid obstacle course in this example uses the third, fourth and fifth posts, which give jumps of height 1 and 2, summing to 3.

**Constraints**

To evaluate your solution, the judges will run your program against several different input files. All of these files will adhere to the following bounds:

- $3 \leq N \leq 100\,000$  (the number of fence posts)
- $1 \leq h_i \leq 100\,000$  (the height of each post)

As some of the test cases will be quite large, you may need to think about how well your solution scales for larger input values. However, not all the cases will be large. In particular:

- For 30% of the marks,  $N \leq 300$ .
- For an additional 30% of the marks,  $N \leq 3\,000$ .
- For the remaining 40% of the marks, no special constraints apply.