CS F211
Data Structures and Algorithms
Assignment - 2

Arrays and Binary Search

Allowed Language: **C**

January 17, 2024

General Tips

- Try to use functions as much as possible in your code. Functions increase reusability and the pass-by-value feature provides a significant help sometimes. Modularizing your code also helps you to debug efficiently.
- Use `scanf` to read characters/strings from STDIN. Avoid using `getchar`, `getc` or `gets`. Try to read up about character suppression in `scanf` as it will be very helpful in some of the problems.
- Use `printf` instead of `putc`, `putchar` or `puts` to print character/string output on STDOUT.
- Indent your code appropriately and use proper variable names. These increase readability and writability of the code. Also, Use comments wherever necessary.
- Use a proper IDEs like Sublime Text or VSCode as they help to run and test your code on multiple test-cases easily.
- **Note:** Kindly try to do all of these questions by yourself at least once. Spend some time thinking about it, or trying to code it instead of directly asking help of your friends or searching it up online. This helps you understand the question, allowing you to solve further questions which are not in the scope of this Assignment yourself.

Nom's Peanut Butter Quest: A Tale of Binary Search

Once upon a bamboo-filled valley, there lived a peculiar panda named Nom. Nom was not your typical bamboo-munching panda; instead, Nom had an extraordinary fondness for peanut butter. While other pandas spent their days leisurely enjoying bamboo shoots, Nom tirelessly roamed the forest in search of anything remotely related to peanut butter.

One sunny day, as Nom was sauntering through the lush bamboo groves, an ancient scroll fell from a tree. Intrigued, Nom unrolled the scroll to discover a series of mysterious symbols and problems. It turned out to be a set of binary search problems, hidden in the heart of the bamboo forest by a wise old turtle.

For every problem Nom solves, it unrolls the scroll further to uncover a series of maps. Every map had a treasure mark. According to the legend of the old turtle, every treasure had hundreds of jars of peanut butter.

The panda, fueled by his love for peanut butter, decided to embark on a journey to solve the problems and claim his treasure. Are you guys ready to help Nom collect 'em all?!

A: Hunger Games

In search of peanut butter, Nom gets very tired walking for days. Soon he realises that his first problem was somewhere he would never have expected. In the Bamboo Forest! Struck by hunger Nom decides to chomp on some bamboo. But only bamboo of a certain height can fulfill his hunger, not bigger, not smaller. He sees a bunch of bamboos lined up from left to right in ascending order of their heights. Help Nom find the perfect bamboo to beat his hunger.
Note: Assume that bamboo of the height Nom loves, always exists in the bunch.

Input

The first line contains a two integer N and H ($1 \leq N \leq 10^6$) and ($1 \leq H \leq 10^9$) - the number of bamboos in the bunch and the height of the bamboo Nom loves respectively.
The second line contains an array of size N containing $(a_0, a_1, \ldots, a_{N-1})$ where ($1 \leq a_i \leq 10^9$) - heights of different bamboos sorted in ascending order.

Output

A single integer, position of the bamboo in the bunch (array). Assume 1-based indexing.
Your solution should run in *O(logN)* time complexity.

```
input
5 24
12 23 24 56 37

output
3
```

```
input
10 230
99 102 123 126 210 211 230 300 309 415

output
7
```

B: Peanut Butter Summit

After a full belly, Nom entered the mountains. As he ascended, the path became steeper and more treacherous. To navigate the complex landscape, our panda friend faced the challenge of identifying the peak index, the point where the mountain slope reached its summit. Furthermore, the peak was known as the peanut butter peak since ages. Now, in order to get there, Nom has to find the best strategy to obtain the index of the peak, thus reaching it. The problem is as follows. You are given an array with N integers, such that there exists an i:

- $a_1 < a_2 < \ldots < a_i$
- $a_i > a_{i+1} > \ldots > a_n$

You have to help Nom find the index of the peak. Assume 1-based indexing.

Input

The first line contains a positive integer N $(1 \leq N \leq 10^6)$.
The second line contains N integers $a_1, a_2, \ldots, a_N$ $(-10^9 \leq a_i \leq 10^9)$.

Output

A single positive integer indicating the peak index of the mountain.
Your solution should run in *O(logN)* time complexity.

---

input
6
1 2 4 6 4 3

output
4

---

input
10
12 14 17 20 23 21 17 12 9 3

output
5

C: The Legend of 7

For his third problem, Nom meets Thala. In the dressing room there were N Jerseys hung, numbered from 1 to N in a sorted manner from left to right. Bhajji being the mischievous guy he is, decides to steal one random Jersey numbered k and then swaps the bundles of the remaining Jerseys from the right of k to its left and vice versa. For example, if initially there were 8 Jerseys $[1, 2, 3, 4, 5, 6, 7, 8]$ and Bhajji steals Jersey 5 and swaps the bundles, the final order of the Jerseys would be $[6, 7, 8, 1, 2, 3, 4]$.

Thala's Jersey Number is 7. Thala is worried that Bhajji might have stolen his Jersey. You're given an integer N and an array of size $N - 1$ containing the order of the Jerseys after Bhajji stole one. Help Nom find out whether Thala's Jersey is safe or not. If it is present, output its current index in the array, otherwise -1. Assume 0-based indexing.

Input

The first line contains a single integer N $(7 \leq N \leq 10^6)$ - the number of Jerseys.
The second line contains an array of size $N - 1$ containing $(a_0, a_1, \ldots, a_{N-2})$ where $(1 \leq a_i \leq N)$.

Output

A single integer, index of Thala's Jersey in the array, if present. -1 otherwise.
Your solution should run in *O(logN)* time complexity.

```
input
8
6 7 8 1 2 3 4

output
1
```

```
input
7
4 5 6 7 1 2

output
3
```

---

input
12
8 9 10 11 12 1 2 3 4 5 6

output
-1

D: Root for Nom

Further ahead in his journey for peanut butter, Nom suddenly got warped back in the 1910s. He ended up meeting Einstein. Einstein was solving the Hamiltonian Equation at that moment, and was in serious need of help of someone to calculate square roots for him. Nom agrees and opens his laptop, but realises that due to the time travel, he lost all the inbuilt operators. You have to help Nom find all the square roots without using any inbuilt functions or operators.

Further Einstein tells you both to find the square root upto 6 digits after decimal. No more, no less and no rounding off.

Input

The first line contains a positive integers N ($1 \leq N \leq 2^{31} - 1$) - the number whose square root you have to find.

Output

A floating-point number with 6 digits after the decimal.
Your solution should run in *O(logN)* time complexity.

---

input
245

output
15.652475

---

input
1

output
1.000000

---

input
6969

output
83.480536

E: Final Destination

For the final challenge, Nom comes to the top of a cliff. Below he sees a $N \times N$ matrix-like ground filled with landmines. But, there is a possibility that there could be one block without any mines. In order to obtain the final jar of peanut butter, he must jump on that one block only, about which is written in the scroll, if it exists. Otherwise, he will lose all the previous peanut butter jars. It is given that each row is sorted and the last block of i-th row is strictly lesser than the first block of $(i+1)$-th row. Find out whether or not Nom should jump down.

Input

The first line contains a single integer N $(1 \leq N \leq 2 * 10^3)$ - dimensions of the matrix-like land.
The second line contains a single integer K $(-10^9 \leq K \leq 10^9)$ - the block on which Nom should land.
Consequent N lines contain N integers $(a_0, \ldots, a_{N-1})$ separated with a space $(-10^9 \leq a_i \leq 10^9)$.

Output

"YES" if the block exists, otherwise "NO".
Your solution should run in *O(log(N×N))* time complexity.

```
input
3
34
2 20 21
25 30 31
34 37 40

output
YES
```

```
input
4
16
2 20 21 23
25 30 31 32
34 37 40 42
44 48 49 50

output
NO
```

F: How to become a Millionaire

Now that Nom has got all of his peanut butter, he went to sleep. Now, Kira wants you to help him with his financial problems. For the first problem, you've been given an array of size N consisting of annual revenues in millions. Here, Kira will ask you Q queries to find out the revenue from element l of the array to element r (both inclusive). Assume 1-based indexing.

Input

The first line contains two integers N and Q ($1 \leq N, Q \leq 10^6$).
The second line contains N elements $a_1, a_2, \ldots, a_N$ ($-10^9 \leq a_i \leq 10^9$).
Next Q lines contain two integers l and r ($1 \leq l < r \leq N$).

Output

Q lines each with the cumulative revenue of their respective queries.

```
input
5 3
2 4 0 -1 4
1 3
2 3
1 4

output
6
4
5
```

```
input
4 4
3 -2 -1 7
1 4
2 3
3 4
2 4

output
7
-3
6
4
```

G: How to Become a Better Millionaire

Now that Kira knows which years have led to the highest revenues, he has made a plan for each year. He wants you to calculate which plans to consider so that he can become the richest. You will be given with an array of revenue plans. You have to find a contiguous subarray with the highest revenue in order to make him rich.

Input

The first line contains a positive integer $N(1 \leq N \leq 10^6)$ - number of revenue plans.
The second line contains N plans $a_1, a_2, \ldots, a_N$ $(-10^9 \leq a_i \leq 10^9)$.

Output

A single integer - highest possible revenue which Kira can get.

---

input
6
2 -1 -2 4 6 10

output
20

---

input
8
1 2 3 -2 0 0 3 2

output
9

---

input
3
2 -2 2

output
2

H: Scam 2024

Kira finds out that life is not very simple. He wanted to get even richer. So kira realised that revenue doesn't matter. All the big money lies in stocks. So he decided to start trading. He wants your help with basic trading initially, where he has a list of prices of a stock for N days and you have to find out the maximum profit he can make out of that certain stock.
Note: Kira can buy the stock only on 1 day and can sell it any day after the day he bought it. Although, it is not necessary that Kira buys a stock.

Input

The first line contains a positive integer $N(1 \leq N \leq 10^6)$ - the number of days.
The second line contains N integers $a_1, a_2, \ldots, a_N$ $(1 \leq a_i \leq 10^9)$ - prices across N days of the stock.

Output

A single integer telling the maximum profit Kira can make

---

input
5
7 1 4 5 2

output
4

---

input
7
5 6 9 1 2 3 4

output
4

---

input
5
7 6 5 4 3

output
0

I: Uncontrollable Kira

Kira got a hang of stocks and started following Harshad Mehta. Now he decided, that he will not stop at just one stock. He will buy as many as he wants, but he can hold onto just one stock at a time. Although, he can buy and sell anything at any time. Given an array of N stock prices per day, find the maximum profit Kira can make this way.

Input

The first line contains a positive integer $N(1 \leq N \leq 10^6)$ - the number of days.
The second line contains N integers $a_1, a_2, \ldots, a_N$ $(1 \leq a_i \leq 10^9)$ - prices across N days of the stock.

Output

A single integer telling the maximum profit Kira can make

```
input
9
5 2 3 5 6 7 2 4 2

output
7
```

```
input
7
3 4 5 2 5 4 2

output
5
```

```
input
9
1 2 3 4 5 4 3 2 1

output
4
```

J: TheHackerCat 2

Vidyateja is back with his websites, and since he was so generous the last time, you trust him again. This time he makes a malicious notes website and sends it to you on Whatsapp. You open it, and this time, all the electronic gadgets and home appliances you own, which has any type of Operating System, gets hacked. But, being the generous guy he is, he gives you a small matrix to fill and a number N. This matrix can only be filled in a clockwise spiral order using numbers from 1 to N^2.

Input

The first line contains a single integer $N(1 \leq N \leq 1000)$.

Output

A matrix of size $N \times N$.

---

input
3 (See *Figure 1* for clarity)

output
1 2 3
8 9 4
7 6 5

---

input
2

output
1 2
4 3

---

input
1

output
1

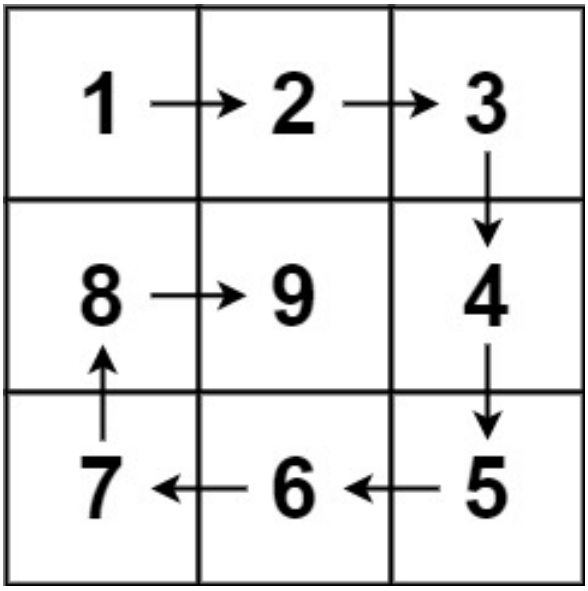

Figure 1: Example 1