

SoftspaceX Test

1. Write a program to insert a New Element, Array (10, 20, 30, 40, 50). New Element will auto detect its index. Using pointers.

Sample Input

5

25

10 20 30 40 50

Sample Output

10 20 25 30 40 50

And perform Deletion

Input

30

Output

10 20 25 40 50

2. Write a program to insert a New Element, Array (String, String, String, String). User will Define the Index. If the Array range is full then print not inserted element.

Sample Input

10

5

2

25

10 20 30 40 50

Sample Output

10 20 25 30 40 50

3. Write a program to find prime numbers from 2 to n. User
Defend Library. Also build that same function using
recursion. Using Pointers.

Sample Input

2 50

Sample Output

2 3 ..Prime ..47

4. A queue is an abstract data type that maintains the order in which elements were added to it, allowing the oldest elements to be removed from the front and new elements to be added to the rear. This is called a *First-In-First-Out* (FIFO) data structure because the first element added to the queue (i.e., the one that has been waiting the longest) is always the first one to be removed.

A basic queue has the following operations:

Enqueue: add a new element to the end of the queue.

Dequeue: remove the element from the front of the queue and return it.

In this challenge, you must first implement a queue using *two stacks*. Then process queries, where each query is one of the following types:

- 1 x: Enqueue element into the end of the queue.
- 2: Dequeue the element at the front of the queue.
- 3: Print the element at the front of the queue.

Input Format

The first line contains a single integer, n , denoting the number of queries.

Each line of the subsequent lines contains a single query in the

form described in the problem statement above. All three queries start with an integer denoting the query , but only query 1 is followed by an additional space-separated value, x , denoting the value to be enqueued.

Constraints

- $1 \leq q \leq 10^5$
- $1 \leq type \leq 3$
- $1 \leq |x| \leq 10^9$
- It is guaranteed that a valid answer always exists for each query of type 3.

Output Format

For each query of type 3, print the value of the element at the front of the queue on a new line.

Sample Input

STDIN	Function
10	q = 10 (number of queries)
1 42	1st query, enqueue 42
2	dequeue front element
1 14	enqueue 14
3	print the front element
1 28	enqueue 28
3	print the front element
1 60	enqueue 60
1 78	enqueue 78
2	dequeue front element
2	dequeue front element

Sample Output

```
14
14
```

5. Solve This

Given two numbers N and M . N indicates the number of elements in the array $A[]$ ($1 - indexed$) and M indicates number of queries. You need to perform two types of queries on the array $A[]$.

You are given M queries. Queries can be of two types, type **1** and type **2**.

- Type 1 queries are represented as **1 i j** : Modify the given array by removing elements from i to j and adding them to the front.
- Type 2 queries are represented as **2 i j** : Modify the given array by removing elements from i to j and adding them to the back.

Your task is to simply print $|A[1] - A[N]|$ of the resulting array after the execution of M queries followed by the resulting array.

Note While adding at back or front the order of elements is preserved.

Input Format

First line consists of two space-separated integers, N and M .

Second line contains N integers, which represent the elements of the array.

M queries follow. Each line contains a query of either type 1 or type 2 in the form *type i j*

Constraints

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

$$1 \leq A[i] \leq 10^9$$

$$1 \leq i \leq j \leq N$$

Output Format

Print the absolute value i.e. $abs(A[1] - A[N])$ in the first line.

Print elements of the resulting array in the second line. Each element should be separated by a single space.

Sample Input

```
8 4
1 2 3 4 5 6 7 8
1 2 4
2 3 5
1 4 7
2 1 4
```

Sample Output

```
1
2 3 6 5 7 8 4 1
```

6. Write a program to perform linear Search and Binary Search using Pointers.

7. Solve This as well

1

2 3 2

3 4 5 4 3

4 5 6 7 6 5 4

Using classes and objects