**Functions & Arrays**

**Q1. Fibonacci Number**

**Given a number N, figure out if it is a member of fibonacci series or not. Return true if the number is member of fibonacci series else false.**

**Fibonacci Series is defined by the recurrence**

F(n) = F(n-1) + F(n-2)

**where F(0) = 0 and F(1) = 1**

**Input Format :**

Integer N

**Output Format :**

true or false

**Constraints :**

0 <= n <= 10^4

**Sample Input 1 :**

5

**Sample Output 1 :**

true

**Sample Input 2 :**

14

**Sample Output 2 :**

false

**Q2. Find the Difference**

**Given an array/list A with N elements, you need to find difference of sum of elements at even indices (E) and sum of elements at odd indices(O).**

**Note : Array/List indexes start from 0.**

**Input Format :**

Line 1 : Size of the array/list i.e N

Line 2 : N integers i.e. elements of the array/list separated by space

**Output Format :**

Difference i.e. E - O.

**Input Constraints :**

**1<=N<=10^6**

**1 <= Ai <= 10^4**

**Sample Input :**

5

1 2 3 4 5

**Sample Output :**

3

**Sample Output Explanation :**

**Sum of elements at even indices = 1 + 3 + 5 = 9**

**Sum of elements at odd indices = 2 + 4 = 6**

**So output is (9 - 6) i.e. 3.**

**Q3. Linear Search**

**You have been given a random integer array/list(ARR) of size N, and an integer X. You need to search for the integer X in the given array/list using 'Linear Search'.**

**You have been required to return the index at which X is present in the array/list. If X has multiple occurrences in the array/list, then you need to return the index at which the first occurrence of X would be encountered. In case X is not present in the array/list, then return -1.**

**'Linear search' is a method for finding an element within an array/list. It sequentially checks each element of the array/list until a match is found or the whole array/list has been searched.**

**Input format :**

First line of each test case or query contains an integer 'N' representing the size of the array/list.

Second line contains 'N' single space separated integers representing the elements in the array/list.

Third line contains the value of X(integer to be searched in the given array/list)

**Output format :**

Print the index at which X is present or -1, otherwise.

Output for every test case will be printed in a separate line.

**Constraints :**

0 <= N <= 10^5

-2 ^ 31 <= X <= (2 ^ 31) - 1

Time Limit: 1 sec

**Sample Input 1:**

1

7

2 13 4 1 3 6 28

3

**Sample Output 1:**

4

**Sample Input 2:**

2

7

2 13 4 1 3 6 28

9

5

7 8 5 9 5

5

**Sample Output 2:**

-1

2

**Q4. Swap Alternate**

**You have been given an array/list(ARR) of size N. You need to swap every pair of alternate elements in the array/list.**

**Try to only change in the input array itself and print the final swapped array.**

**Input Format :**

First line contains an integer 'N' representing the size of the array/list.

Second line contains 'N' single space separated integers representing the elements in the array/list.

**Output Format :**

Print the elements of the resulting array in a single row separated by a single space.

Output for every test case will be printed in a separate line.

**Constraints :**

0 <= N <= 10^5

Time Limit: 1sec

**Sample Input 1:**

6

9 3 6 12 4 32

**Sample Output 1 :**

3 9 12 6 32 4

**Sample Input 2:**

9

9 3 6 12 4 32 5 11 19

**Sample Output 2 :**

3 9 12 6 32 4 11 5 19

**Sample Input 3:**

4

1 2 3 4

**Sample Output 3:**

2 1 4 3

**Q5. Arrange Numbers in Array**

**You have to make an empty array(ARR) and its size N.**

**Your task is to populate the array using the integer values in the range 1 to N(both inclusive) in the order - 1,3,5,.......,6,4,2.**

**Input Format :**

The first and the only line of each test case or query contains an integer 'N'.

**Output Format :**

For each test case, print the elements of the array/list separated by a single space.

Output for every test case will be printed in a separate line.

**Constraints :**

0 <= N <= 10^4

Time Limit: 1sec

**Sample Input 1 :**

6

**Sample Output 1 :**

1 3 5 6 4 2

**Explanation of Sample Input 1 :**

Since the value of N is 6, the number will be stored in the array in such a fashion that 1 will appear at 0th index, then 2 at the last index, in a similar fashion 3 is stored at index 1. Hence the array becomes 1 3 5 6 4 2.

**Sample Input 2 :**

9

**Sample Output 2 :**

1 3 5 7 9 8 6 4 2

**Sample Input 3:**

3

**Sample Output 3:**

1 3 2

**Search and Sort**

**Q6. Binary Search**

**You have been given a sorted(in ascending order) integer array/list(ARR) of size N and an element X. Write a function to search this element in the given input array/list using 'Binary Search'. Return the index of the element in the input array/list. If the element is not present in the array/list, then return -1.**

**Input format :**

The first line contains an Integer 'N', which denotes the size of the array/list.

The second line contains 'N' single space-separated integers representing the elements in the array/list.

The third line contains the value of X to be searched for in the array/list.

**Output Format :**

Print the index at which X is present for each test case, -1 otherwise.

**Constraints :**

0 <= N <= 10^6

0 <= X <= 10^9

Time Limit: 1 sec

**Sample Input 1:**

7

1 3 7 9 11 12 45

3

**Sample Output 1:**

1

**Sample Input 2:**

7

1 2 3 4 5 6 7

9

**Sample Output 2:**

-1

**Q7. Bubble Sort**

**Provided with a random integer array/list(ARR) of size N, you have been required to sort this array using 'Bubble Sort'.**

**Note:**

Change in the input array/list itself. Print the elements.

**Input format :**

First line of each test case or query contains an integer 'N' representing the size of the array/list.

Second line contains 'N' single space separated integers representing the elements in the array/list.

**Output format :**

Print the elements of the array/list in sorted order separated by a single space.

Output for every test case will be printed in a separate line.

**Constraints :**

0 <= N <= 10^3

Time Limit: 1 sec

**Sample Input 1:**

1

7

2 13 4 1 3 6 28

**Sample Output 1:**

1 2 3 4 6 13 28

**Sample Input 2:**

2

5

9 3 6 2 0

4

4 3 2 1

**Sample Output 2:**

0 2 3 6 9

1 2 3 4

**Q8. Selection Sort**

**Provided with a random integer array/list(ARR) of size N, you have been required to sort this array using 'Selection Sort'.**

**Note:**

Change in the input array/list itself. Print the elements.

**Input format :**

First line of each test case or query contains an integer 'N' representing the size of the array/list.

Second line contains 'N' single space separated integers representing the elements in the array/list.

**Output format :**

Print the elements of the array/list in sorted order separated by a single space.

Output for every test case will be printed in a separate line.

**Constraints :**

0 <= N <= 10^3

Time Limit: 1 sec

**Sample Input 1:**

1

7

2 13 4 1 3 6 28

**Sample Output 1:**

1 2 3 4 6 13 28

**Sample Input 2:**

2

5

9 3 6 2 0

4

4 3 2 1

**Sample Output 2:**

0 2 3 6 9

1 2 3 4

**Q9. Insertion Sort**

**Provided with a random integer array/list(ARR) of size N, you have been required to sort this array using 'Insertion Sort'.**

**Note:**

Change in the input array/list itself. Print the elements.

**Input format :**

First line contains an integer 'N' representing the size of the array/list.

Second line contains 'N' single space separated integers representing the elements in the array/list.

**Output Format :**

Print the elements of the array/list in sorted order separated by a single space.

Output for every test case will be printed in a separate line.

**Constraints :**

0 <= N <= 10^3

Time Limit: 1 sec

**Sample Input 1:**

7

2 13 4 1 3 6 28

**Sample Output 1:**

1 2 3 4 6 13 28

**Sample Input 2:**

5

9 3 6 2 0

**Sample Output 2:**

0 2 3 6 9

**Sample Input 3:**

4

4 3 2 1

**Sample Output 4:**

1 2 3 4

**Q10. Merge Two Sorted Arrays**

**You have been given two sorted arrays/lists(ARR1 and ARR2) of size N and M respectively, merge them into a third array/list such that the third array is also sorted.**

**Input Format :**

First line contains an integer 'N' representing the size of the first array/list.

Second line contains 'N' single space separated integers representing the elements of the first array/list.

Third line contains an integer 'M' representing the size of the second array/list.

Fourth line contains 'M' single space separated integers representing the elements of the second array/list.

**Output Format :**

Print the sorted array/list(of size N + M) in a single row, separated by a single space.

Output for every test case will be printed in a separate line.

**Constraints :**

0 <= N <= 10^5

0 <= M <= 10^5

Time Limit: 1 sec

**Sample Input 1**

5

1 3 4 7 11

4

2 4 6 13

**Sample Output 1 :**

1 2 3 4 4 6 7 11 13

**Sample Input 2 :**

3

10 100 500

7

4 7 9 25 30 300 450

**Sample Output 2 :**

4 7 9 10 25 30 100 300 450 500

**Sample Input 3:**

4

7 45 89 90

0

**Sample Output 3:**

7 45 89 90

**CHOCOLATE QUESTION**

**Q. Maximize the sum**

**Given 2 sorted arrays (in increasing order), find a path through the intersections that produces maximum sum and return the maximum sum.**

**That is, we can switch from one array to another array only at common elements.**

**If no intersection element is present, we need to take sum of all elements from the array with greater sum.**

**Input Format :**

Line 1 : An integer M i.e. size of first array

Line 2 : M integers which are elements of first array, separated by spaces

Line 3 : An integer N i.e. size of second array

Line 4 : N integers which are elements of second array, separated by spaces

**Output Format :**

Maximum sum value

**Constraints :**

***1 <= M, N <= 10^6***

**Sample Input :**

6

1 5 10 15 20 25

5

2 4 5 9 15

**Sample Output :**

81

**Explanation :**

**We start from array 2 and take sum till 5 (sum = 11). Then we'll switch to array at element 10 and take till 15. So sum = 36. Now, no elements left in array after 15, so we'll continue in array 1. Hence sum is 81**