**Module 2 -> Function and Arrays.**

# Arrays-Binary Search

Take as input N, the size of array. Take N more inputs and store that in an array. Take as input a number M. Write a function which returns the index on which M is found in the array, in case M is not found -1 is returned. Print the value returned.You can assume that the array is sorted, but you’ve to optimize the finding process. For an array of size 1024, you can make 10 comparisons at maximum.

1.It reads a number N.  
2.Take Another N numbers as input in Ascending Order and store them in an Array.  
3.Take Another number M as input and find that number in Array.  
4.If the number M is found, index of M is returned else -1 is returned.Print the number returned.

**Input Format**

**Constraints**

N cannot be Negative. Range of Numbers N and M can be between -1000000000 to 1000000000

**Output Format**

**Sample Input**

5

3

5

6

9

78

6

**Sample Output**

2

**Explanation**

Array = {3, 5, 6, 9, 78}, target number = 6 . Index of number 6 in the given array = 2. Write Binary search to find the number in given array as discuss in the class.

# Arrays-Reverse an Array

Take as input N, the size of array. Take N more inputs and store that in an array. Write a function that reverses the array. Print the values in reversed array.

1.It reads a number N.  
2.Take Another N numbers as input and store them in an Array.  
3.Reverse the elements in the Array.  
4.Print the reversed Array.

**Input Format**

First-line contains a single integer n denoting the size of the array.  
Next, N line contains a single integer denoting the elements of the array.

**Constraints**

N cannot be Negative. Range of Numbers can be between -1000000000 to 1000000000.

**Output Format**

Print the elements of the reversed array

**Sample Input**

5

0

4

6

8

9

**Sample Output**

9

8

6

4

0

**Explanation**

In the sample case , arr=[0,4,6,8,9] is reversed to arr=[9,8,6,4,0].

# Inverse of an array

Take as input N, a number. Take N more inputs and store that in an array. Write a recursive function which inverses the array. Print the values of inverted array

**Input Format**

Enter a number N and take N more inputs

**Constraints**

None

**Output Format**

Display the values of the inverted array in a space separated manner

**Sample Input**

5

0 2 4 1 3

**Sample Output**

0 3 1 4 2

**Explanation**

Swap element with index

for eg : element 4 at index 2 becomes element 2 at index 4

# Calculate The Sum

Raj is a very smart kid who recently started learning computer programming. His coach gave him a cyclic array A having N numbers, and he has to perform Q operations on this array. In each operation the coach would provide him with a number X. After each operation, every element of the cyclic array would be replaced by the sum of itself and the element lying X positions behind it in the cyclic array. All these replacements take place simultaneously. For example, if the cyclic array was [a, b, c, d], then after the operation with X = 1, the new array would be [a+d, b+a, c+b, d+c]. He needs to output the sum of the elements of the final array modulus 10^9+7. He made a program for it but it's not very efficient. You know he is a beginner, so he wants you to make an efficient program for this task because he doesn't want to disappoint his coach.

**Input Format**

The first line of each test file contains a integer N. The next line contains N space separated integers which represent the elements of the cyclic array. The third line contains a integer Q representing the number of operations that will be applied to the array. Finally, Q lines follow, each one containing an integer X .

**Constraints**

1 <= N <= 100000  
1 <= Ai<= 10^9  
0 <= Q <= 1000000  
0 <= X < N

**Output Format**

Your program should output to the standard output stream the sum of the elements of the final array modulus 10^9+7.

**Sample Input**

5

1 2 3 4 5

2

1

0

**Sample Output**

60

**Explanation**

After the 1st operation (X = 1), the array would be [1+5, 2+1, 3+2, 4+3, 5+4] =[6, 3, 5, 7, 9]  
After 2nd operation (X = 0), the array would be [6+6, 3+3, 5+5, 7+7, 9+9] =[12, 6, 10, 14, 18]  
Thus the correct answer would equal to = (12+6+10+14+18) % (10^9+7) = 60

# Arrays-Max Value In Array

Take an input N, the size of array. Take N more inputs and store that in an array. Write a function which returns the maximum value in the array. Print the value returned.

1.It reads a number N.

2.Take Another N numbers as input and store them in an Array.

3.calculate the max value in the array and return that value.

**Input Format**

First line contains integer n as size of array. Next n lines contains a single integer as element of array.

**Constraints**

N cannot be Negative. Range of Numbers can be between -1000000000 to 1000000000

**Output Format**

Print the required output.

**Sample Input**

4

2

8

6

4

**Sample Output**

8

**Explanation**

Arrays= {2, 8, 6, 4} => Max value = 8 .

**Alex Goes Shopping**

# Alex Goes Shopping

It is Alex’s birthday and she wants to go shopping. She only has ‘A’ units of money and she wants to spend all of her money. However, she can only purchase one kind of item. She goes to a shop which has ‘n’ types items with prices A0,A1,A2,…,An-1. The shopkeeper claims that he has atleast ‘k’ items she can choose from. Help her find out if the shopkeeper is correct or not.

**Input Format**

The first line contains an integer ‘n’ denoting the number of items in the shop. The second line contains ‘n’ space-separated integers describing the respective price of each item. The third line contains an integer ‘q’ denoting the number of queries. Each of the subsequent lines contains two space-separated integers ‘A’ and ‘k’

**Constraints**

1 <=n, Ai, A <= 105 where 0<=i<="" br="" style="outline: 0px; box-sizing: inherit;">

1 <= q <= 2\*n  
1 <= k <= n  
The array may contain duplicate elements.

**Output Format**

For each query, print Yes on a new line if the shopkeeper is correct; otherwise, print No instead.

**Sample Input**

4

100 200 400 100

5

100 2

200 3

500 4

600 4

800 4

**Sample Output**

Yes

Yes

No

No

Yes

**Explanation**

In query 1, Alex has 100 units of money. The shopkeeper claims that she can choose to buy from 2 kinds of items i.e. item 1 and item 4 each priced at 100.

In query 2, The shopkeeper claims that she can choose to buy from 3 kinds of items ie item 1 and item 4 each priced at 100(she can buy 1 from either of the two), or item 2 priced at 200(she can buy one)

In query 3, she has 500 units of money. She can either buy item 1 or item 4 ( 5 of each kind respectively). Thus, she has only 2 kinds of items to choose from.

In query 5, she has 800 units of money. She can either buy item 1 or item 4 ( 8 of each kind respectively) or item 2(she can buy 4 of these) or item 3(2 of these). Thus, she has 4 kinds of items to choose from

# Arrays-Target Sum Pairs

Take as input N, the size of array. Take N more inputs and store that in an array. Take as input “target”, a number. Write a function which prints all pairs of numbers which sum to target.

**Input Format**

The first line contains input N. Next N lines contains the elements of array and (N+1)th line contains target number.

**Constraints**

Length of the arrays should be between 1 and 1000.

**Output Format**

Print all the pairs of numbers which sum to target. **Print each pair in increasing order**.

**Sample Input**

5

1

3

4

2

5

5

**Sample Output**

1 and 4

2 and 3

**Explanation**

Find any pair of elements in the array which has sum equal to target element and print them.

# Arrays-Target Sum Triplets

Take as input N, the size of array. Take N more inputs and store that in an array. Take as input “target”, a number. Write a function which prints all triplets of numbers which sum to target.

**Input Format**

First line contains input N.  
Next line contains N space separated integers denoting the elements of the array.  
The third line contains a single integer T denoting the target element.

**Constraints**

Length of Array should be between 1 and 1000.

**Output Format**

Print all the triplet present in the array in a new line each. The triplets must be printed as A, B and C where A,B and C are the elements of the triplet ( A<=B<=C) and all triplets must be printed in sorted order. Print only unique triplets.

**Sample Input**

9

5 7 9 1 2 4 6 8 3

10

**Sample Output**

1, 2 and 7

1, 3 and 6

1, 4 and 5

2, 3 and 5

**Explanation**

Array = {5, 7, 9, 1, 2, 4, 6 ,8 ,3}. Target number = 10. Find any three number in the given array which sum to target number.

# Maximum Sum Path in Two Arrays

You are provided two sorted arrays. You need to find the maximum length of bitonic subsequence. You need to find the sum of the maximum sum path to reach from beginning of any array to end of any of the two arrays. You can switch from one array to another array only at common elements.

**Input Format**

First line contains integer t which is number of test case. For each test case, it contains two integers n and m which is the size of arrays and next two lines contains n and m space separated integers respectively.

**Constraints**

1<=t<=100 1<=n,m<=100000

**Output Format**

Print the maximum path.

**Sample Input**

1

8 8

2 3 7 10 12 15 30 34

1 5 7 8 10 15 16 19

**Sample Output**

122

**Explanation**

122 is sum of 1, 5, 7, 8, 10, 12, 15, 30, 34

# Pair of Roses

Deepak has a limited amount of money that he can spend on his girlfriend. So he decides to buy two roses for her. Since roses are of varying sizes, their prices are different. Deepak wishes to completely spend that fixed amount of money on buying roses for her.  
As he wishes to spend all the money, he should choose a pair of roses whose prices when summed up are equal to the money that he has.  
Help Deepak choose such a pair of roses for his girlfriend.  
  
NOTE: If there are multiple solutions print the solution that minimizes the difference between the prices i and j. After each test case, you must print a blank line.

**Input Format**

The first line indicates the number of test cases T.  
Then, in the next line, the number of available roses, N is given.  
The next line will have N integers, representing the price of each rose, a rose that costs less than 1000001.  
Then there is another line with an integer M, representing how much money Deepak has.  
There is a blank line after each test case.

**Constraints**

1≤ T ≤100  
2 ≤ N ≤ 10000  
Price[i]<1000001

**Output Format**

For each test case, you must print the message: ‘Deepak should buy roses whose prices are i and j.’, where i and j are the prices of the roses whose sum is equal do M and i ≤ j. You can consider that it is always possible to find a solution. If there are multiple solutions print the solution that minimizes the difference between the prices i and j.

**Sample Input**

2

2

40 40

80

5

10 2 6 8 4

10

**Sample Output**

Deepak should buy roses whose prices are 40 and 40.

Deepak should buy roses whose prices are 4 and 6.

**Explanation**

Find two such kinds of price of roses which has sum up to equal to Deepak's Money.

# Painter's Partition Problem Java

Given **K** painters to paint **N** boards where each painter takes 1 unit of time to paint 1 unit of boards i.e. if the length of a particular board is 5, it will take 5 units of time to paint the board. Compute the minimum amount of time to paint all the boards.

**Note that:**

* *Every painter can paint only contiguous segments of boards.*
* *A board can only be painted by 1 painter at maximum.*

**Input Format**

First line contains **K** which is the number of painters. Second line contains **N** which indicates the number of boards. Third line contains N space separated integers representing the length of each board.

**Constraints**

1 <= K <= 10  
1 <= N <= 10  
1<= Length of each Board <= 10^8

**Output Format**

Output the minimum time required to paint the board.

**Sample Input**

2

2

1 10

**Sample Output**

10

# Aggressive Cows

Farmer John has built a new long barn, with N (2 <= N <= 100,000) stalls. The stalls are located along a straight line at positions x1,…,xN (0 <= xi <= 1,000,000,000).

His C (2 <= C <= N) cows don't like this barn layout and become aggressive towards each other once put into a stall. To prevent the cows from hurting each other, FJ wants to assign the cows to the stalls, such that the minimum distance between any two of them is as large as possible. What is the largest minimum distance?

**Input Format**

First line contains ***N*** and ***C***, separated by a single space, representing the total number of stalls and number of cows respectively. The next line contains ***N*** integers containing the indexes of stalls.

**Constraints**

2 <= N <= 100,000  
0 <= xi <= 1,000,000,000  
2 <= C <= N

**Output Format**

Print one integer: the largest minimum distance.

**Sample Input**

5 3

1 2 8 4 9

**Sample Output**

3

**Explanation**

Problem Credits - (Spoj)[http://www.spoj.com/problems/AGGRCOW/]

# Help Ramu

Ramu often uses public transport. The transport in the city is of two types: cabs and rickshaws. The city has n rickshaws and m cabs, the rickshaws are numbered by integers from 1 to n, the cabs are numbered by integers from 1 to m.

Public transport is not free. There are 4 types of tickets:

A ticket for one ride on some rickshaw or cab. It costs c1 ruppees;  
A ticket for an unlimited number of rides on some rickshaw or on some cab. It costs c2 ruppees;  
A ticket for an unlimited number of rides on all rickshaws or all cabs. It costs c3 ruppees;  
A ticket for an unlimited number of rides on all rickshaws and cabs. It costs c4 ruppees.

Ramu knows for sure the number of rides he is going to make and the transport he is going to use. He asked you for help to find the minimum sum of ruppees he will have to spend on the tickets.

**Input Format**

Each Test case has 4 lines which are as follows:

The first line contains four integers c1, c2, c3, c4 (1 ≤ c1, c2, c3, c4 ≤ 1000) — the costs of the tickets.  
The second line contains two integers n and m (1 ≤ n, m ≤ 1000) — the number of rickshaws and cabs Ramu is going to use.  
The third line contains n integers ai (0 ≤ ai ≤ 1000) — the number of times Ramu is going to use the rickshaw number i.  
The fourth line contains m integers bi (0 ≤ bi ≤ 1000) — the number of times Ramu is going to use the cab number i.

**Constraints**

1 <= T <= 1000 , where T is no of testcases  
1 ≤ c1, c2, c3, c4 ≤ 1000  
1 ≤ n, m ≤ 1000  
0 ≤ ai , bi ≤ 1000

**Output Format**

For each testcase , print a single number - the minimum sum of rupees Ramu will have to spend on the tickets in a new line.

**Sample Input**

2

1 3 7 19

2 3

2 5

4 4 4

4 3 2 1

1 3

798

1 2 3

**Sample Output**

12

1

**Explanation**

For the first testcase ,  
The total cost of rickshaws = min( min(2 \* 1, 3) + min(5 \* 1, 3), 7) = min(5, 7) = 5  
The total cost of cabs = min( min(4 \* 1, 3) + min(4 \* 1, 3) + min(4 \* 1, 3) , 7) = min ( 9, 7) = 7  
Total final cost = min( totalCabCost + totalRickshawCost , c4) = min( 5 + 7, 19) = min ( 12, 19) = 12  
We print 12.

# Book Allocation Problem

You are given number of pages in n different books and m students. The books are arranged in ascending order of number of pages. Every student is assigned to read some consecutive books. The task is to assign books in such a way that the maximum number of pages assigned to a student is minimum.

**Input Format**

First line contains integer t as number of test cases. Next t lines contains two lines. For each test case, 1st line contains two integers n and m which represents the number of books and students and 2nd line contains n space separated integers which represents the number of pages of n books in ascending order.

**Constraints**

1 < t < 50  
1< n < 100  
1< m <= 50  
1 <= Ai <= 1000

**Output Format**

Print the maximum number of pages that can be assigned to students.

**Sample Input**

1

4 2

12 34 67 90

**Sample Output**

113

**Explanation**

1st students : 12 , 34, 67 (total = 113)  
2nd students : 90 (total = 90)  
Print max(113, 90)

# Product of Array Except Self

Given an array arr of n integers where n > 1, return an array output such that output[i] is equal to the product of all the elements of arr except arr[i].

**Input Format**

First line contains integer **N** as size of array.  
Next line contains a **N** integer as element of array.

**Constraints**

**arr[i]<=10000000**

**Output Format**

print output array

**Sample Input**

4

1 2 3 4

**Sample Output**

24 12 8 6

# KTH ROOT

You are given two integers n and k. Find the greatest integer x, such that, x^k <= n.

**Input Format**

First line contains number of test cases, T. Next T lines contains integers, n and k.

**Constraints**

1<=T<=10  
1<=N<=10^15  
1<=K<=10^4

**Output Format**

Output the integer x

**Sample Input**

2

10000 1

1000000000000000 10

**Sample Output**

10000

31

**Explanation**

For the first test case, for x=10000, 10000^1=10000=n

# Murthal Parantha

The coding blocks members went to the success party of their first ever online boot-camp at Murthal. They ordered **P** number of paranthas. The stall has **L** cooks and each cook has a rank **R**. A cook with a rank **R** can cook 1 parantha in the first R minutes 1 more parantha in the next 2R minutes, 1 more parantha in 3R minutes and so on(he can only cook a complete parantha) ( For example if a cook is ranked 2.. he will cook one parantha in 2 minutes one more parantha in the next 4 mins and one more in the next 6 minutes hence in total 12 minutes he cooks 3 paranthas. In 13 minutes also he can cook only 3 paranthas as he does not have enough time for the 4th parantha). Calculate the minimum time needed to cook all the paranthas.

**Input Format**

First line contains P, the number of pratha ordered. In the next line the first integer denotes the number of cooks L and L integers follow in the Next line each denoting the rank of a cook.

**Constraints**

P <= 1000  
L <= 50  
1 <= R <= 8

**Output Format**

Print an integer which tells the number of minutes needed to get the order done.

**Sample Input**

10

4

1 2 3 4

**Sample Output**

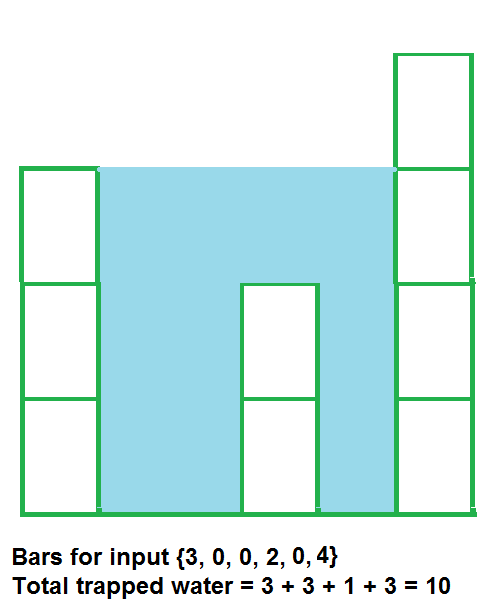
12

**Explanation**

First cook with rank 1 cooks 4 paranthas in 10 minutes (1+2+3+4).  
Second cook with rank 2 cooks 3 paranthas in 12 minutes (2+4+6)  
Third cook with rank 3 cooks 2 paranthas in 9 minutes (3+6) Fourth cook with rank 4 only needs to cook one last remaining parantha. He can do that in 4 minutes.  
Since these cooks cook parallely, the total time taken will be the maximum of the four i.e. 12 minutes.

# Rain Water Trapping

You are given an input array whose each element represents the height of a line towers. The width of every tower is 1. It starts raining. Water is filled between the gap of towers if possible. You need to find how much water filled between these given towers.

Example : 

**Input Format**

The first line consists of number of test cases T. Each test case consists an integer N as number of towers and next line contains the N space separated integers.

**Constraints**

1 <= N <= 1000000 1 <= t <= 10 0 <= A[i] <= 10000000

**Output Format**

Print how much unit of water collected among towers for each test case.

**Sample Input**

2

6

3 0 0 2 0 4

12

0 1 0 2 1 0 1 3 2 1 2 1

**Sample Output**

10

6

# Arrays-Linear Search

Take as input N, the size of an array. Take N more inputs and store that in an array. Take another number’s input as M. Write a function which returns the index on which M is found in an array, in case M is not found -1 is returned. Print the value returned.

1. It reads a number N.  
   2.Take Another N numbers as an input and store them in an Array.
2. Take another number M as an input.
3. If M is found in the Array the index of M is returned else -1 is returned and print the value returned.

**Input Format**

**Constraints**

N cannot be Negative. Range of Numbers can be between -1000000000 to 1000000000. M can be between -1000000000 to 1000000000.

**Output Format**

**Sample Input**

5

2

4

6

9

17

17

**Sample Output**

4

**Explanation**

Given array = {2, 4, 6, 9, 17}. Target number = 17. Index = 4.

# Kartik Bhaiya And Strings

Kartik Bhaiya has a string consisting of only 'a' and 'b' as the characters. Kartik Bhaiya describes perfectness of a string as the maximum length substring of equal characters. Kartik Bhaiya is given a number **k** which denotes the maximum number of characters he can change. Find the maximum perfectness he can generate by changing no more than **k** characters.

**Input Format**

The first line contains an integer denoting the value of K. The next line contains a string having only ‘a’ and ‘b’ as the characters.

**Constraints**

2 ≤ N ≤ 10^6

**Output Format**

A single integer denoting the maximum perfectness achievable.

**Sample Input**

2

abba

**Sample Output**

4

**Explanation**

We can swap the a's to b using the 2 swaps and obtain the string "bbbb". This would have all the b's and hence the answer 4.  
Alternatively, we can also swap the b's to make "aaaa". The final answer remains the same for both cases.

# Find Square Root

Implement squareroot(x).

Compute and return the square root of x.

If x is not a perfect square, return floor(sqrt(x)).  
**Note:** DO NOT USE SQRT FUNCTION FROM STANDARD LIBRARY

**Input Format**

An integer A .

**Constraints**

0 <= A <10000000000

**Output Format**

Squareroot of A.

**Sample Input**

25

**Sample Output**

5

# Divisible Subarrays

You are given N elements, *a1,a2,a3….aN*. Find the number of *good* sub-arrays.  
A good sub-array is a sub-array [*ai,ai+1,ai+2….aj*] such that (*ai+ai+1+ai+2+….+aj*) is divisible by N.

**Input Format**

The first line contains the number of test cases T. First line of each test case contains an integer N denoting the number of elements. Second line of each test case contains N integers, a1, a2, a3….aN, where ai denotes the ith element of the array.

**Constraints**

1<=T<=10  
1<=N<=10^5  
|ai|<=10^9

**Output Format**

Output a single integer denoting the number of good sub-arrays.

**Sample Input**

2

5

1 1 1 1 1

5

5 5 5 5 5

**Sample Output**

1

15

**Explanation**

In first test case, there is only one sub-array [1, 1, 1, 1, 1], such that 1+1+1+1+1=5, which is divisible by N=5

# Next Permutation

Given an array Arr[], Treat each element of the array as the digit and whole array as the number. Implement the next permutation, which rearranges numbers into the numerically next greater permutation of numbers.

If such arrangement is not possible, it must be rearranged as the lowest possible order ie, sorted in an ascending order.

**Note:** The replacement must be in-place, do not allocate extra memory.

**Input Format**

The First Line contains the Number of test cases T.  
Next Line contains an Integer N, number of digits of the number.  
Next Line contains N-space separated integers which are elements of the array 'Arr'.

**Constraints**

1 <= T <= 100  
1 <= N <= 1000  
0 <= Ai <= 9

**Output Format**

Print the Next Permutation for each number separated by a new Line.

**Sample Input**

2

3

1 2 3

3

3 2 1

**Sample Output**

1 3 2

1 2 3

**Explanation**

Possible permutations for {1,2,3} are {1,2,3} , {1,3,2} , {2,1,3} , {2,3,1}, {3,1,2} and {3,2,1}. {1,3,2} is the immediate next permutation after {1,2,3}.  
For the second testcase , {3,2,1} is the last configuration so we print the first permutation as its next permutation.

# Sorting In linear time

Given an array having n elements representing balls of various colours like red, white or blue, sort them in-place so that balls of the same colour are adjacent, with the colours in the order red, white and blue.  
To represent the colour red, white, and blue we will use the integers 0, 1, and 2 respectively.  
**Note:**You are not suppose to use the sort function for this problem.

**Input Format**

First line contains integer n as size of array. Next n lines contains a single integer as element of array.

**Constraints**

None

**Output Format**

Print the balls in order so that balls of same colour are adjacent.

**Sample Input**

6

2 0 2 1 1 0

**Sample Output**

0 0 1 1 2 2

# **Maximum Circular Sum**

You are provided n numbers (both +ve and -ve). Numbers are arranged in a circular form. You need to find the maximum sum of consecutive numbers.

**Input Format**

First line contains integer t which is number of test case.  
For each test case, it contains an integer n which is the size of array and next line contains n space separated integers denoting the elements of the array.

**Constraints**

1<=t<=100  
1<=n<=1000  
|Ai| <= 10000

**Output Format**

Print the maximum circular sum for each testcase in a new line.

**Sample Input**

1

7

8 -8 9 -9 10 -11 12

**Sample Output**

22

**Explanation**

Maximum Circular Sum = 22 (12 + 8 - 8 + 9 - 9 + 10)