Programming Using C

week 08practice session coding

Name:K.Kamaleshwaran
Department:AIML-'B'
Roll No.:242501079

Question 1
Correct
Marked out of 1.00
Flag question

Coderz here is a simple task for you, you have given an array of size **N** and an integer **M**.

Your task is to calculate the ***difference between maximum sum and minimum sum of N-M*** elements of the given array.

Constraints:

1<=t<=**10**
1<=n<=**1000**
1<=a[i]<=**1000**

Input:

First line contains an integer **T** denoting the number of testcases.
First line of every testcase contains two integer **N** and **M**.
Next line contains **N** space separated integers denoting the elements of array.

Output:

For every test case print your answer in new line

SAMPLE INPUT

1
5 1
1 2 3 4 5

SAMPLE OUTPUT

4

Explanation

M is 1 and N is 5 so you have to calculate maximum and minimum sum using (5-1 =) 4 elements.
Maximum sum using the 4 elements would be (2+3+4+5=)14.
Minimum sum using the 4 elements would be (1+2+3+4=)10.
Difference will be 14-10=4.

Answer: (penalty regime: 0 %)

```
#include<stdio.h>
int main()
{
    int t;
    scanf("%d",&t);
    while(t--)
    {
        int n,m,d,min,temp;
        scanf("%d %d",&n,&m);
        d=n-m;
        int arr[n];
        for(int i=0;i<n;i++)
        {
            scanf("%d",&arr[i]);
        }
        for(int j=0;j<n;j++)
        {
            min=j;
            for (int k=j+1;k<n;k++)
            {
                if(arr[k]<arr[min])
                min=k;
            }
            temp=arr[min];
            arr[min]=arr[j];
            arr[j]=temp;
        }
        int maxsum=0,minsum=0;
        for(int a=0;a<d;a++)
        minsum+=arr[a];
        for(int b=n-1;b>=n-d;b--)
        maxsum+=arr[b];
        printf("%d\n",maxsum-minsum);
    }
}
```

	Input	Expected	Got	
✓	1 5 1 1 2 3 4 5	4	4	✓

Passed all tests! ✓

A new deadly virus has infected large population of a planet. A medical scientist has documented a sensitivity of virus which can cure the disease. Vaccine produced from this virus has various strength depending on viral blood cells count. A person is cured only if viral blood cells count is greater than or equal to strength of person. A doctor receives a new set of report which contains viral blood cells count of each infected patient. Practo stores all vaccine doctor has and their viral blood cells count. You need to determine if doctor can cure all patients with the vaccines he has. The number of vaccines and patients are equal.

Input Format

First line contains the number of vaccines . N. Second line contains N integers, which are strength of vaccines. Third line contains N integers, which are viral blood cells count of patients.

Output Format

Print a single line containing **"Yes"** or **"No"**.

Input Constraint

1 ≤ N ≤ 10

Strength of vaccines and viral blood cells count of patients fit in integer.

SAMPLE INPUT

```
3
123 146 454 542 456
100 328 248 689 200
```

SAMPLE OUTPUT

```
No
```

Answer (penalty regime: 0 %)

```
#include<stdio.h>
int main()
{
    int n,min2,min1,temp,flag=1;
    scanf("%d",&n);
    int vac[n],pat[n];
    for(int i=0;i<n;i++)
    scanf("%d",&vac[i]);
    for(int i=0;i<n;i++)
    {
        min1=i,min2=i;
        for(int k=i;k<n;k++)
        {
            if(vac[k]<vac[min1])
            min1=k;
            if(pat[k]<pat[min2])
            min2=k;
        }
        temp=vac[min1];
        vac[min1]=vac[i];
        vac[i]=temp;
        temp=pat[min2];
        pat[min2]=pat[i];
        pat[i]=temp;
    }
    for (int i=0;i<n;i++)
    {
        if(vac[i]<pat[i])
        {
            flag=0;
            break;
        }
    }
    if(flag==1)
    printf("Yes");
    else
    printf("No");
    return 0;
}
```

	Input	Expected	Got	
✓	3 123 146 454 542 456 100 328 248 689 200	No	No	✓

Passed all tests! ✓

You are given an array of n integer numbers $a_0, a_1, \ldots, a_{n-1}$. Calculate the number of pair of indices (i, j) such that $0 \le i < j \le n$ and $a_i \text{ or } a_j = 0$

Input format

First line is containing the number of array elements.
Second line is space separated integers $a_0, a_1, \ldots, a_{n-1}$

Output format

Output the required number of pairs

Constraints

$1 \le n \le 10^5$
$1 \le a_i \le 10^9$

SAMPLE INPUT

```
4
1 0 1 0
```

SAMPLE OUTPUT

```
3
```

Explanation

The 3 pair of indices are $(1, 3)$ and $(2,3)$

Answer: (penalty regime: 0 %)

```
#include<stdio.h>
int main()
{
    int n,count=0;
    scanf("%d",&n);
    int a[n];
    for(int i=0;i<n;i++)
    {
        scanf("%d",&a[i]);
    }
    for(int i=0;i<n-1;i++)
    {
        for(int j=i+1;j<n;j++)
        {
            if((a[i]|a[j])==0)
            count++;
        }
    }
    printf("%d",count);
    return 0;
}
```

	Input	Expected	Got	
✓	4 1 0 1 0	3	3	✓

Passed all tests! ✓

You are given an array A of non-negative integers of size n. Your task is to sort the array in non-decreasing order and print out the original indices of the new sorted array.

Example:

A=[4,3,2,7,1]

After sorting the new array becomes A=[1,2,3,4,7].

The required output should be "4 3 2 1 5"

INPUT :

The first line of input contains n,the size of the array.
The next line contains n of the array of size n.

OUTPUT :

Output consists of a single line of integers.

CONSTRAINTS:

1<=size<=**100**
0<=a[i]<=**100**

NOTE: The indexing of the array starts with 0.

SAMPLE INPUT

5
4 3 2 7 1

SAMPLE OUTPUT

4 3 2 1 5

Answer: (penalty regime: 0 %)

```
#include<stdio.h>
int main()
{
    int n;
    scanf("%d",&n);
    int arr[n];
    for(int i=0;i<n;i++)
    scanf("%d",&arr[i]);
    int max=arr[0];
    for(int i=0;i<n;i++)
    {
        if(arr[i]>max)
        max=arr[i];
    }
    max++;
    int smin=0;
    for(int a=0;a<n;a++)
    {
        for(int i=0;i<n;i++)
        {
            if(arr[i]<arr[smin])
            smin=i;
        }
        printf("%d ",smin+1);
        arr[smin]=max;
    }
}
```

	Input	Expected	Got	
✓	5 4 3 2 7 1	4 3 2 1 5	4 3 2 1 5	✓

Passed all tests! ✓