The next line consists of the array of size m

OUTPUT :

Output consists of a single line of integers

CONSTRAINTS:

1<=m<=100
0<=A[i]<=100

NOTE: The indexing of the array starts with 0.

SAMPLE INPUT

5
4 5 3 7 1

SAMPLE OUTPUT

4 2 0 1 3

Answer: (penalty regime: 0 %)

```c
#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 min=0;
    for(int a=0;a<n;a++){
        for(int b=0;b<n;b++){
            if(arr[b]<arr[min])
            min=b;
        }
        printf("%d ",min);
        arr[min]=max;
    }
}
```

	Input	Expected	Got	
✓	5 4 5 3 7 1	4 2 0 1 3	4 2 0 1 3	✓

Passed all tests! ✓

Output the required number of pairs.

Constraints

$1 \leq n \leq 10^6$

$1 \leq a_i \leq 10^9$

SAMPLE INPUT

5

1 3 1 4 3

SAMPLE OUTPUT

2

Explanation

The 2 pair of indices are *(1, 3)* and *(2,5)*.

Answer: (penalty regime: 0 %)

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

5

123 146 454 542 456

100 328 248 689 200

SAMPLE OUTPUT

No

Answer: (penalty regime: 0 %)

```
#include<stdio.h>
int main (){
    int n,min1,min2,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++)
    scanf("%d",&pat[i]);
    for(int j=0;j<n-1;j++){
        min1=j,min2=j;
        for(int k=j;k<n;k++){
            if(vac[k]<vac[min1])
            min1=k;
            if(pat[k]<pat[min2])
            min2=k;
        }
        temp=vac[min1];
        vac[min1]=vac[j];
        vac[j]=temp;
        temp=pat[min2];
        pat[min2]=pat[j];
        pat[j]=temp;
    }
    for(int i=0;i<n;i++)
    {

        if(vac[i]<=pat[i]){
            flag=0;
            break;
        }
    }
    if(flag==0)
    printf("No");
    else
    printf("Yes");
}
```

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 %)

```c
#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-1;j++){
            min=j;
            for(int k=j;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-1;b--)
        maxsum+=arr[b];
        printf("%d\n",(maxsum-minsum));
    }
}
```