

<b>Status</b>	Finished
<b>Started</b>	Monday, 1 December 2025, 7:29 PM
<b>Completed</b>	Monday, 1 December 2025, 8:57 PM
<b>Duration</b>	1 hour 28 mins

Question **1**

Correct

coders 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:**

**$1 \leq t \leq 10$**

**$1 \leq n \leq 1000$**

**$1 \leq a[i] \leq 1000$**

**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**

## 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 %)

```

1 #include<stdio.h>
2 #include<stdlib.h>
3 int cmp(const void*a,const vo
4 B- int main(){
5     int T;
6     scanf("%d",&T);
7 B-     while(T--){
8         int N,M;
9         scanf("%d%d",&N,&M);
10        int a[N];
11 B-        for(int i=0;i<N;i++){
12            scanf("%d",&a[i]);
13        }
14        qsort(a,N,sizeof(int),cm
15        int k=N-M;
16        long long min=0,max=0;
17        for(int i=0;i<k;i++){ mi
18        for(int j=N-k;j<N;j++){
19            printf("%lld\n",max-min)
20        }
21        return 0;
22    }

```

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

Passed all tests!  $\frac{m\Gamma}{\sim}$

**Question 2**

Correct

a new deadly virus has infected large population of a planet. a brilliant scientist has discovered a new strain of virus which can cure this disease. Vaccine produced from this virus has various strength depending on midichlorians count. A person is cured only if midichlorians count in vaccine batch is more than midichlorians count of person. A doctor receives a new set of report which contains midichlorians count of each infected patient, Practo stores all vaccine doctor has and their midichlorians count. You need to determine if doctor can save 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 midichlorians count of patients.

**Output Format**

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

**Input Constraint**

**1 < N < 10**

Strength of vaccines and midichlorians count of patients fit in integer.

**SAMPLE INPUT**

5

123 146 454 542 456

100 328 248 689 200

**SAMPLE OUTPUT**

No

**Answer:** (penalty regime: 0 %)

```
1 #include<stdio.h>
2 #include<stdlib.h>
3 int cmp(const void*a,const vo
4 B- int main(){
5     int N;
6     scanf("%d",&N);
7     int v[N],p[N];
8     for(int i=0;i<N;i++){sc
9     for(int i=0;i<N;i++){sc
10    qsort(v,N,sizeof(int),cmp
11    qsort(p,N,sizeof(int),cmp
12 B-    for(int i=0;i<N;i++){
13 B-        if(v[i]<=p[i]){
14            printf("No");
15            return 0;
16        }
17    }
18 }
19 }
```



	Input	Expected	Got	
~	5 123 146 454 542 456 100 328 248 689 200	No	No	~

Passed all tests!

**Question 3**

Correct

you are given an array  $a$  of non-negative integers of size  $m$ . Your task is to sort the array in non-decreasing order and print out the original indices of the new sorted array.

**Example:**

A={4,5,3,7,1}

After sorting the new array becomes A={1,3,4,5,7}.

The required output should be "4 2 0 1 3"

**INPUT :**

The first line of input consists of the size of the array

The next line consists of the array of size m

**OUTPUT :**

Output consists of a single line of integers

**CONSTRAINTS:**

$1 \leq m \leq 106$

$0 \leq A[i] \leq 106$

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 %)

```
1 #include<stdio.h>
2 #include<stdlib.h>
3 typedef struct{int val,idx;}p
4 int cmp(const void*a,const vo
5         return((p*)a)->val-((p*)b
6 }
7 int main(){
8     int n;
9     scanf("%d",&n);
10    p a[n];
11   for(int i=0;i<n;i++){
12       scanf("%d",&a[i].
13       a[i].idx=i;
14   }
15   qsort(a,n,sizeof(p),c
16   for(int i=0;i<n;i++){
17       printf("%d ",a[i]
18   }
19   return 0;
20 }
21
```

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

Passed all tests!