

GE23131-Programming Using C-2024

Quiz navigation

1 2 3 4

Show one page at a time

Finish review

Status Finished
Started Wednesday, 25 December 2024, 3:47 PM
Completed Wednesday, 25 December 2024, 4:34 PM
Duration 46 mins 42 secs

Question 1
Correct
Marked out of 1.00
Flag question

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

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>
#include<stdlib.h>
4 int compare(const void*a, const void *b) {
5     return ((int *)a - (int *)b);
6 }
7
8 int main() {
9     int t,n,m,i;
10    int *arr;
11    scanf("%d",&t);
12    scanf("%d %d",&n,&m);
13    arr = (int *)malloc(n * sizeof(int));
14    for (i = 0; i < n; i++) {
15        scanf("%d",&arr[i]);
16    }
17    qsort(arr,n,sizeof(int),compare);
18    int maxSum = 0,minSum = 0;
19    for (i = 0; i < n-m; i++) {
20        minSum += arr[i];
21        maxSum += arr[n-1-i];
22    }
23    printf("%d\n",maxSum - minSum);
24    free(arr);
25 }
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41 }
```

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

Passed all tests! ✓

Question 2
Correct
Marked out of 1.00
Flag question

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

$1 \leq N \leq 10$

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 %)

```
#include<stdio.h>
#include<stdlib.h>
3
4 int compare(const void*a, const void *b) {
5     return ((int *)a - (int *)b);
6 }
7
8 int main() {
9     int N;
10    scanf("%d",&N);
11    int vaccines[N], patients[N];
12
13    for (int i=0; i<N; i++) {
14        scanf("%d",&vaccines[i]);
15    }
16
17    for (int i=0; i<N; i++) {
18        scanf("%d",&patients[i]);
19    }
20
21    qsort(vaccines, N, sizeof(int), compare);
22    qsort(patients, N, sizeof(int), compare);
23
24    int canSaveAll = 1;
25
26    for (int i=0; i<N; i++) {
27        if (vaccines[i] >= patients[i]) {
28            canSaveAll = 0;
29            break;
30        }
31    }
32
33
34
35
36
37
38
39
40
41 }
```

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

Passed all tests! ✓

Question 3
Correct
Marked out of 1.00
Flag question

You are given an array of n integer numbers a_1, a_2, \dots, a_n . Calculate the number of pair of indices (i, j) such that $1 \leq i < j \leq n$ and $a_i \text{ xor } a_j = 0$.

Input format

- First line: n denoting the number of array elements

- Second line: n space separated integers a_1, a_2, \dots, a_n

Output format

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 %)

```
#include<stdio.h>
#include<stdlib.h>
3
4 int main() {
5     int n,i,j;
6     scanf("%d",&n);
7
8     int arr[n];
9
10    for (i = 0; i < n; i++) {
11        scanf("%d",&arr[i]);
12    }
13
14    long long count = 0;
15
16    for (i = 0; i < n; i++) {
17        for (j = i+1; j < n; j++) {
18            if (arr[i] == arr[j]) {
19                count++;
20            }
21        }
22    }
23
24    printf("%lld\n",count);
25
26    return 0;
27 }
```

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

Passed all tests! ✓

Question 4
Correct
Marked out of 1.00
Flag question

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:

Output consists of a single line of integers.

CONSTRAINTS:

$1 \leq m \leq 10^6$

$0 \leq A[i] \leq 10^9$

SAMPLE INPUT

5
4 5 3 7 1

SAMPLE OUTPUT

4 2 0 1 3

Answer: (penalty regime: 0 %)

```
#include<stdio.h>
#include<stdlib.h>
3
4 int main() {
5     int m;
6     scanf("%d",&m);
7
8     int *A = (int *)malloc(m * sizeof(int));
9     int *indices = (int *)malloc(m * sizeof(int));
10
11    for (int i = 0; i < m; i++) {
12        scanf("%d",&A[i]);
13        indices[i] = i;
14    }
15
16    for (int i = 0; i < m; i++) {
17        for (int j = i+1; j < m; j++) {
18            if (A[i] > A[j]) {
19                int temp = A[i];
20                A[i] = A[j];
21                A[j] = temp;
22            }
23        }
24    }
25
26    printf("%d", indices[0]);
27
28    for (int i = 1; i < m; i++) {
29        printf(" %d", indices[i]);
30    }
31
32    free(A);
33    free(indices);
34
35    return 0;
36 }
```

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

Passed all tests! ✓

Question 5
Correct
Marked out of 1.00
Flag question

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:

Output consists of a single line of integers.

CONSTRAINTS:

$1 \leq m \leq 10^6$

$0 \leq A[i] \leq 10^9$

SAMPLE INPUT

5
4 5 3 7 1

SAMPLE OUTPUT

4 2 0 1 3

Answer: (penalty regime: 0 %)

```
#include<stdio.h>
#include<stdlib.h>
3
4 int main() {
5     int m;
6     scanf("%d",&m);
7
8     int *A = (int *)malloc(m * sizeof(int));
9     int *indices = (int *)malloc(m * sizeof(int));
10
11    for (int i = 0; i < m; i++) {
12        scanf("%d",&A[i]);
13        indices[i] = i;
14    }
15
16    for (int i = 0; i < m; i++) {
17        for (int j = i+1; j < m; j++) {
18            if (A[i] > A[j]) {
19                int temp = A[i];
20                A[i] = A[j];
21                A[j] = temp;
22            }
23        }
24    }
25
26    printf("%d", indices[0]);
27
28    for (int i = 1; i < m; i++) {
29        printf(" %d", indices[i]);
30    }
31
32    free(A);
33    free(indices);
34
35    return 0;
36 }
```

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

Passed all tests! ✓

Finish review