

Sorting Algorithms – Bubble and Selection

Ex.No.:**Date:****EasyGoing****ProblemStatement:**

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

Constraints:

$1 \leq t \leq 10$

$1 \leq n \leq 1000$

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

InputFormat:

FirstlinecontainsanintegerTdenotingthenumberoftestcases. First line of every testcase contains two integerN and M.

Nextlinecontains Nspace separatedintegers denotingtheelements of array

Output:

Foreverytestcaseprintyouranswerinnewline

SampleInput

1

51

12345

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 $14-10=4$.

Program:

```

1  #include<stdio.h>
2  int main()
3  { int t;
4    scanf("%d",&t);
5    while(t-->0)
6    {
7      int n,m,d,min,temp;
8      scanf("%d %d",&n,&m);
9      d=n-m;
10     int arr[n];
11     for(int i=0;i<n;i++)
12       scanf("%d",&arr[i]);
13     for(int j=0;j<n;j++)
14     {
15       min=j;
16       for(int k=j;k<n;k++)
17       {
18         if(arr[k]<arr[min])
19           min=k;
20       }
21       temp=arr[min];
22       arr[min]=arr[j];
23       arr[j]=temp;
24     }
25     int maxsum=0,minsum=0;
26     for(int a=0;a<d;a++)
27       minsum+=arr[a];
28     for(int b=n-1;b>m-1;b--)
29       maxsum+=arr[b];
30     printf("%d\n",maxsum-minsum);
31   }
32 }
33
34
35
36
37
38
39

```

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

Passed all tests! ✓

Ex.No.:**Date:****Sortitout!****ProblemStatement:**

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 "42013"

InputFormat:

The first line of input consists of the size of the array. The next line consists of the array of size m.

OutputFormat:

Output consists of a single line of integers.

Constraints:

$1 \leq m \leq 10^6$

$0 \leq A[i] \leq 10^6$

NOTE: The indexing of the array starts with 0.

Sample Input

5

4 5 3 7 1

Sample Output

4 2 0 1 3

Program:

```

1  #include<stdio.h>
2  int main()
3  {
4      int n;
5      scanf("%d",&n);
6      int arr[n];
7      for(int i=0;i<n;i++)
8          scanf("%d",&arr[i]);
9      int max=arr[0];
10     for(int i=1;i<n;i++)
11     {
12         if(arr[i]>max)
13             max=arr[i];
14     }
15     max++;
16     int min=0;
17     for(int a=0;a<n;a++)
18     {
19         for (int b=0;b<n;b++)
20         {
21             if(arr[b]<arr[min])
22                 min=b;
23         }
24         printf("%d ",min);
25         arr [min]=max;
26     }
27 }
28

```

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

Passed all tests! ✓

Ex.No.:**Date:****SavePatients****ProblemStatement:**

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.

InputFormat

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.

OutputFormat

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

InputConstraint

$1 < N < 10$

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

SampleInput

```
5
123146454542456
100328248689200
```

Sample Output

```
No
```

Program:

```

1  #include<stdio.h>
2  int main()
3  {
4      int n,min1,min2,temp,flag=1;
5      scanf("%d",&n);
6      int vac[n],pat[n];
7      for(int i=0;i<n;i++)
8      {
9          scanf("%d",&vac[i]);
10     }
11     for(int i=0;i<n;i++)
12     scanf("%d",&pat[i]);
13
14     for(int j=0;j<n-1;j++)
15     {
16         min1=j,min2=j;
17         for (int k=j;k<n;k++)
18         {
19             if(vac[k]<vac[min1])
20             min1=k;
21             if(pat[k]<pat[min2])
22             min2=k;
23         }
24         temp=vac[min1];
25         vac[min1]=vac[j];
26         vac[j]=temp;
27         temp=pat[min2];
28         pat[min2]=pat[j];
29         pat[j]=temp;
30     }
31
32     for(int i=0;i<n;i++)
33     {
34         if(vac[i]<=pat[i])
35         {
36             flag=0;
37             break;
38         }
39     }
40
41     }
42     if(flag==1)
43     printf("Yes");
44     else
45     printf("No");
46 }
47

```

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

Passed all tests! ✓

Ex.No.:**Date:****ShubhamandXor****ProblemStatement:**

You are given an array of n integers a_1, a_2, \dots, a_n . Calculate the number of pairs of indices (i, j) such that $1 \leq i < j \leq n$ and $a_i \oplus a_j = 0$.

Inputformat

- First line: n denoting the number of array elements
- Second line: n space separated integers a_1, a_2, \dots, a_n .

Outputformat

Output the required number of pairs.

Constraints

$$1 \leq n \leq 10^6$$

$$1 \leq a_i \leq 10^9$$

SampleInput

5
1 3 1 4 3

Sample Output

2

Explanation

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

Program:

```

1  #include<stdio.h>
2  int main()
3  {
4      int n,count=0;
5      scanf("%d",&n);
6      int arr[n];
7      for(int i=0;i<n;i++)
8          scanf("%d",&arr[i]);
9      for(int i=0;i<n-1;i++)
10     {
11         for(int j=i+1;j<n;j++)
12         {
13             if((arr[i]^arr[j])==0)
14                 count++;
15         }
16     }
17     printf("%d",count);
18 }
19
20

```

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

Passed all tests! ✓