

# Problem 01

You are given a string S of small letters, now sort the string on the basis of their count in descending order , It is guaranteed that all the characters count are distinct.

## Sample Input 1:

bbbbaccddd

## Sample Output 1:

bbbdddcca

## Sample Input 2:

aabccc

## Sample Output 2:

cccaab

Explanation of sample input 1 –

Total count of b = 4 , Total count of d=3, Total count of c=2 , Total count of a=1.

We can see that count of b is greatest among all the characters that's why firstly b will come then 2<sup>nd</sup> greatest count is d , 3<sup>rd</sup> greatest count c and lastly a will come.

Hints – Nested loop and check the 31<sup>st</sup> October conceptual session

```

1 #include <stdio.h>
2 int main()
3 {
4     char s[100];
5     scanf("%s", s);
6
7     int i=0, LE= 26, len=strlen(s), total=0;
8     int freq[LE+1];
9
10    for(i=0; i<=LE; i++)
11    {
12        freq[i] = 0;
13    }
14
15    // char to ascii convert +freq array increment(how many b?)
16    for(i=0; i<len; i++)
17    {
18        freq[(s[i] - 'a') + 1]++;
19    }
20
21    for(i=0; i<=LE; i++)
22    {
23        if(freq[i] >= 1)
24            total++;
25    }
26
27    for(int next=0; next!=total; next++)
28    {
29        int max=-1, index=-1;
30        char ch;
31        for(i=1; i<=LE; i++)
32        {
33            if(freq[i] > max)
34            {
35                max = freq[i];
36                ch = i + 96;
37                index = i;
38            }
39        }
40        for(i=1; i<=max; i++)
41        {
42            printf("%c", ch);
43        }
44        freq[index] = 0;
45    }
46    return 0;
47 }

```

## Problem 02

You are given a string S of small letters, now check whether the given string is palindrome or not. If the string is palindrome print YES otherwise print NO.

Note - A palindrome is a string that reads the same backward as forward

**Sample Input 1:**

aabbaa

**Sample Output 1:**

YES

**Sample Input 2:**

abcd

**Sample Output 2:**

NO

Hints – Reverse the string and compare.

```
#include <stdio.h>

int main() {
    int i, len=0, flag=0;
    char w[100], rev[100];
    scanf("%s", w);

    for(i=0; w[i] != '\0'; i++)
        len++;

    for(i=len; i>=0; i--)
    {
        if (w[i] != '\0')
            rev[len-i] = w[i];
    }

    for(i=1; i<=len; i++)
    {
        if(w[i-1] != rev[i])
            flag=1;
        break;
    }

    if(flag==1)
        printf("NO");
    else
        printf("YES");

    return 0;
}
```

## Problem 03

You are given a string *S* of small letters , Now count the number of vowels and consonant from the given string.

### Sample Input 1:

aeibcou

### Sample Output 1:

Vowel – 5

Consonant -2

### Sample Input 2:

trpplqw

### Sample Output 2:

Vowel – 0

Consonant - 7

```
1  #include <stdio.h>
2
3  int main()
4  {
5      int i=0, vc=0, cc=0;
6      char s[100];
7      printf("Enter a word : ");
8      scanf("%s", s);
9
10     for(i=0; s[i]!='\0'; i++)
11     {
12         if(s[i] == 'a' || s[i] == 'e' || s[i] == 'i' || s[i] == 'o' || s[i] == 'u')
13             vc++;
14         else
15             cc++;
16     }
17     printf("Vowel - %d\nConsonant - %d", vc, cc);
18     return 0;
19 }
```

## Problem 04

You are given two matrix of size NxM where N is the row number and M is the column number. Now first line of the input will contain the value of N and M , and the next two line will contain the two matrix, Now perform matrix addition operation.

**Sample Input:**

```
2 3
1 2 3
4 5 6
7 8 9
10 11 12
```

**Sample Output:**

```
8 10 12
14 16 18
```

```
1 #include <stdio.h>
2
3 int main() {
4     int N, M, i, j;
5     printf("Row = ");
6     scanf("%d", &N);
7     printf("Col = ");
8     scanf("%d", &M);
9     int a[N][M], b[N][M];
10
11     printf("Enter %dx%d matrix A \n", N, M);
12     for(i=0; i<N; i++) {
13         for(j=0; j<M; j++) {
14             scanf("%d", &a[i][j]);
15         }
16     }
17     printf("Enter %dx%d matrix B \n", N, M);
18     for(i=0; i<N; i++) {
19         for(j=0; j<M; j++) {
20             scanf("%d", &b[i][j]);
21         }
22     }
23
24     printf("Output : \n");
25     for(i=0; i<N; i++) {
26         for(j=0; j<M; j++) {
27             printf("%d ", a[i][j] + b[i][j]);
28         }
29         printf("\n");
30     }
31     return 0;
32 }
```

## Problem 05

There's a chessboard of size 3×3 (It is a special chessboard for Phitron's student) . R rooks are placed on it and all others cells are empty. Now you need to tell the total empty cells and their position (row,column) .

The first line of the input will contain a single positive integer R, The next line will contain the position of the R rooks in this (x,y) format where x is the row number of the rook and y is the column number of the rook.

Note – Here row and column starts from 1.

**Sample Input 1:**

```
3
2 1
3 3
1 3
```

**Sample Output 1:**

```
Total empty cells - 6

1 1
1 2
2 2
2 3
3 1
3 2
```

```
1  #include <stdio.h>
2
3  int main() {
4      int R, x, y, i, j;
5      scanf("%d", &R);
6      int a[3][3], b[3][3];
7
8      for (i=1; i<=3; i++) {
9          for (j=1; j<=3; j++) {
10             a[i][j] = 0;
11         }
12     }
13
14     for (j=1; j<=R; j++) {
15         scanf("%d %d", &x, &y);
16         a[x][y] = 1;
17     }
18
19     printf("Total empty cells - %d\n", 9-R);
20     for (i=1; i<=R; i++) {
21         for (j=1; j<=R; j++) {
22             if (a[i][j] != 1) {
23                 printf("%d %d", i, j);
24                 printf("\n");
25             }
26         }
27     }
28     return 0;
29 }
```



## Problem 06

You are given a matrix of size  $N \times M$  where  $N$  is the row number and  $M$  is the column number. Now first line of the input will contain the value of  $N$  and  $M$ , and the next line will contain the matrix, Now print the transpose version of this matrix.

Note - The transpose of a matrix is simply a flipped version of the original matrix. We can transpose a matrix by switching its rows with its columns.

Sample Input:

```
3 3
1 2 3
4 5 6
7 8 9
```

Sample Output:

```
1 4 7
2 5 8
3 6 9
```

```
1  #include <stdio.h>
2
3  int main()
4  { int N, M, i, j;
5    scanf("%d %d", &N, &M);
6    int a[N][M];
7
8    for(i=0; i<N; i++){
9      for(j=0; j<M; j++){
10         scanf("%d", &a[i][j]);
11      }
12    }
13
14    printf("Transpose matrix: \n");
15    for(i=0; i<N; i++){
16      for(j=0; j<M; j++){
17         printf("%d ", a[j][i]);
18      }
19      printf("\n");
20    }
21    return 0;
22 }
```

## Codeforces Problem Link –

1) <https://codeforces.com/edu/course/2/lesson/9/1/practice/contest/307092/problem/B>

2) <https://codeforces.com/contest/1511/problem/A>

3) <https://codeforces.com/problemset/problem/1433/C>

```
#include<stdio.h>
// Dominant piranha
int main(){
    int test_case, indexes, max;
    scanf("%d", &test_case);

    for(int i=0; i<test_case; i++){
        scanf("%d", &indexes);
        int a[indexes];

        for(int j=0; j<indexes; j++){
            scanf("%d", &a[j]);
            if(j==0)
                max=a[0];
            else{
                if(a[j]>max)
                    max=a[j];
            }
        }

        int flag=0, l;
        for (int j=0; j<indexes; j++){
            if(j==0 && a[j]==max && a[j]>a[j+1]){
                flag=1;
                l=1;
                break;
            }
            else if(j==indexes-1 && a[indexes-1]==max && a[indexes-1]>a[indexes-2]){
                flag=1;
                l=indexes;
                break;
            }
            else if(j!=0 && j!=indexes-1){
                if(a[j]==max && (a[j]>a[j+1] || a[j]>a[j-1])){
                    flag=1;
                    l=j+1;
                    break;
                }
            }
        }

        if(flag==1)
            printf("%d\n", l);
        else if(flag==0)
            printf("-1\n");
    }

    return 0;
}
```

```
1 #include<stdio.h>
2 // Review site
int main()
{
    int test_case, reviewer, vote, i, review;
    scanf("%d", &test_case);

    while(test_case--){
        vote = 0;
        scanf("%d", &reviewer);
        for(i=0; i<reviewer; i++){
            scanf("%d", &review);
            if(review == 1 || review == 3)
                vote++;
        }

        printf("%d\n", vote);
    }
}
```