

## GE23131 - Programming Using C

**Ex. No. :****Date :****Save Patients****Problem Statement:**

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

**Program:**

```
#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;
    }
}
```

```
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 == 1)
```

```
printf("Yes");
```

```
else
```

```
printf("No");
```

```
}
```

**Ex. No. :****Date :****Shubham and Xor****Problem Statement:**

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).

**Program:**

```
#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);
}
```

Ex. No. :

Date :

**Add Alternate Elements of 2-Dimensional Array****Problem Statement:**

You are given a two-dimensional 3\*3 array starting from A [0][0]. You should add the alternate elements of the array and print its sum. It should print two different numbers the first being sum of A 0 0, A 0 2, A 1 1, A 2 0, A 2 2 and A 0 1, A 1 0, A 1 2, A 2 1.

**Input Format**

First and only line contains the value of array separated by single space.

A00	A01	A02
4	6	9
A10	A11	A12
2	5	8
A20	A21	A22
1	3	7

**Output Format**

First line should print sum of A 0 0, A 0 2, A 1 1, A 2 0, A 2 2

Second line should print sum of A 0 1, A 1 0, A 1 2, A 2 1

**Sample Input**

1 2 3 4 5 6 7 8 9

**Sample Output**

25

20

**Program:**

```

#include <stdio.h>

int main()
{
    int arr[3][3];
    for(int i=0; i<3; i++)
    {
        for(int j=0; j<3; j++)
        {
            scanf("%d", &arr[i][j]);
        }
    }

    int odd = 0, even = 0;
    for(int i=0; i<3; i++)
    {
        for(int j=0; j<3; j++)
        {
            if((i+j)%2 != 0)
                odd += arr[i][j];
            else
                even += arr[i][j];
        }
    }

    printf("%d\n%d", even, odd);
}

```

Ex. No. :

Date :

**The Wealthy Landlord****Problem Statement:**

Shyam Lal, a wealthy landlord from the state of Rajasthan, being an old fellow and tired of doing hard work, decided to sell all his farmland and to live rest of his life with that money. No other farmer is rich enough to buy all his land so he decided to partition the land into rectangular plots of different sizes with different cost per unit area. So, he sold these plots to the farmers but made a mistake. Being illiterate, he made partitions that could be overlapping. When the farmers came to know about it, they ran to him for compensation of extra money they paid to him. So, he decided to return all the money to the farmers of that land which was overlapping with other farmer's land to settle down the conflict. All the portion of conflicted land will be taken back by the landlord.

To decide the total compensation, he has to calculate the total amount of money to return back to farmers with the same cost they had purchased from him. Suppose, Shyam Lal has a total land area of  $1000 \times 1000$  equal square blocks where each block is equivalent to a unit square area which can be represented on the co-ordinate axis. Now find the total amount of money, he has to return to the farmers. Help Shyam Lal to accomplish this task.

**Input Format:** The first line of the input contains an integer N, denoting the total number of pieces he had distributed. Next N lines contain the 5 space separated integers  $(X_1, Y_1), (X_2, Y_2)$  to represent a rectangular piece of land, and cost per unit area C.

$(X_1, Y_1)$  and  $(X_2, Y_2)$  are the locations of first and last square block on the diagonal of the rectangular region.

**Output Format:**

Print the total amount he has to return to farmers to solve the conflict.

**Constraints:**

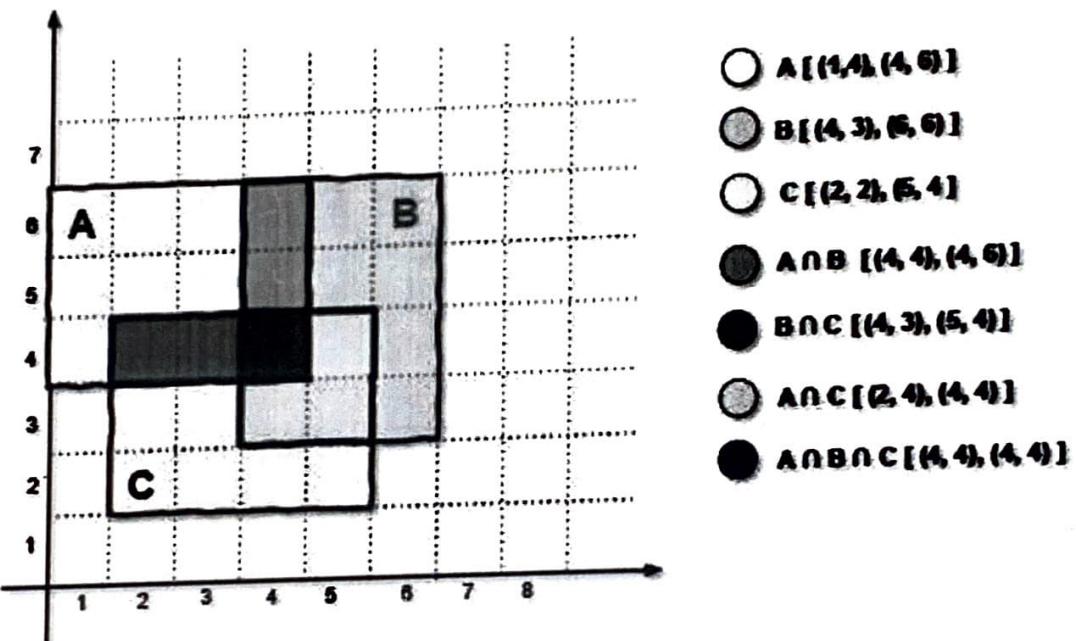
$$\begin{aligned}1 &\leq N \leq 100 \\1 &\leq X_1 \leq X_2 \leq 1000 \\1 &\leq Y_1 \leq Y_2 \leq 1000 \\1 &\leq C \leq 1000\end{aligned}$$

**Sample Input**

```
3
1 4 4 6 1
4 3 6 6 2
2 2 5 4 3
```

**Sample Output**

```
35
```

**Explanation****Simple Illustration of Distribution of Land**

For given sample input (see given graph for reference), compensation money for different farmers is as follows:

Farmer with land area A:  $C_1 = 5 * 1 = 5$

Farmer with land area B:  $C_2 = 6 * 2 = 12$

Farmer with land area C:  $C_3 = 6 * 3 = 18$

Total Compensation Money =  $C_1 + C_2 + C_3 = 5 + 12 + 18 = 35$

**Program:**

```

#include <stdio.h>
int main ()
{
    int i, j, n, x1, x2, y1, y2, t=0;
    long long total = 0;
    int arr[100][100] = {0};
    scanf ("%d", &n);
    while (n--)
    {
        scanf ("%d %d %d %d %d", &x1, &y1, &x2, &y2, &t);
        for (i = x1, i <= x2; i++)
        {
            for (j = y1; j <= y2; j++)
            {
                if (arr[i][j] == 0)
                    arr[i][j] += t;
                else if (arr[i][j] > 0)
                    arr[i][j] = (-1) * (arr[i][j] + t);
                else if (arr[i][j] < 0)
                    arr[i][j] -= t;
            }
        }
    }
}

```

Ex. No. :

Date :

**Priority Interview****Problem Statement:**

Microsoft has come to hire interns from your college. N students got shortlisted out of which few were males and a few females. All the students have been assigned talent levels. Smaller the talent level, lesser is your chance to be selected. Microsoft wants to create the result list where it wants the candidates sorted according to their talent levels, but there is a catch. This time Microsoft wants to hire female candidates first and then male candidates. The task is to create a list where first all-female candidates are sorted in a descending order and then male candidates are sorted in a descending order.

**Input Format**

The first line contains an integer N denoting the number of students. Next, N lines contain two space-separated integers,  $a_i$  and  $b_i$ . The first integer,  $a_i$  will be either 1(for a male candidate) or 0(for female candidate). The second integer,  $b_i$  will be the candidate's talent level.

Constraints:  $1 \leq N \leq 105$ ,  $0 \leq a_i \leq 1$ ,  $1 \leq b_i \leq 109$

**Output Format**

Output space-separated integers, which first contains the talent levels of all female candidates sorted in descending order and then the talent levels of male candidates in descending order.

**Sample Input**

5  
0 3  
1 6  
0 2  
0 7  
1 15

**Sample Output**

7 3 2 15 6

**Program:**

```
#include <stdio.h>

struct data
{
    int que; int tail;
};

int main()
{
    int n;
    scanf("%d", &n);
```

Ex. No. :

Date :

## Strings

### Problem Statement:

#### Input Format

You are given two strings, a and b, separated by a new line. Each string will consist of lower-case Latin characters ('a'-'z').

#### Output Format

In the first line print two space-separated integers, representing the length of a and b respectively.

In the second line print the string produced by concatenating a and b ( $a + b$ ).

In the third line print two strings separated by a space,  $a'$  and  $b'$ .  $a'$  and  $b'$  are the same as a and b, respectively, except that their first characters are swapped.

#### Sample Input

abcd  
ef

#### Sample Output

4 2  
abcdef  
ebcd af

#### Explanation

a = "abcd"  
b = "ef"  
 $|a| = 4$   
 $|b| = 2$   
 $a + b = "abcdef"$   
 $a' = "ebcd"$   
 $b' = "af"$

**Program:**

```
#include <stdio.h>

int main()
{
    char str1[10], str2[10], t;
    int i = 0, j = 0;
    int count1 = 0, count2 = 0;
    scanf ("%s", str1);
    scanf ("%s", str2);
    while (str1[i] != '\0')
    {
        count1++;
        i++;
    }
    while (str2[j] != '\0')
    {
        count2++;
        j++;
    }
    printf ("%d %d\n", count1, count2);
    printf ("%s %s\n", str1, str2);
    t = str1[0];
    str1[0] = str2[0];
    str2[0] = t;
}
```

```
str2[0] = '+';  
printf ("%s %s", str1, str2);  
return 0;  
}
```

**Ex. No. :****Date :**

### **Printing Tokens**

**Problem Statement:**

Given a sentence, s, print each word of the sentence in a new line.

**Input Format**

The first and only line contains a sentence, s.

**Constraints**

$1 \leq \text{len}(s) \leq 1000$

**Output Format**

Print each word of the sentence in a new line.

**Sample Input**

This is C

**Sample Output**

This  
is  
C

**Explanation**

In the given string, there are three words ["This", "is", "C"]. We have to print each of these words in a new line.

**Hint**

Here, once you have taken the sentence as input, we need to iterate through the input, and keep printing each character one after the other unless you encounter a space. When a space is encountered, you know that a token is complete and space indicates the start of the next token after this. So, whenever there is a space, you need to move to a new line, so that you can start printing the next token.

**Program:**

```
#include <stdio.h>

int main()
{
    char s[1000];
    scanf ("%[^\\n]s", s);
    for (int i = 0; s[i] != '\0'; i++)
    {
        if (s[i] != ' ')
            printf ("%c", s[i]);
        else
            printf ("\n");
    }
    return 0;
}
```

**Ex. No. :****Date :**

### **Digit Frequency**

**Problem Statement:**

Given a string, s, consisting of alphabets and digits, find the frequency of each digit in the given string.

**Input Format**

The first line contains a string, num which is the given number.

**Constraints**

$1 \leq \text{len}(\text{num}) \leq 1000$

All the elements of num are made of English alphabets and digits.

**Output Format**

Print ten space-separated integers in a single line denoting the frequency of each digit from 0 to 9.

**Sample Input 0**

a11472o5t6

**Sample Output 0**

0 2 1 0 1 1 1 0 0

**Explanation 0**

In the given string:

- 1 occurs two times.
- 2, 4, 5, 6 and 7 occur one time each.
- The remaining digits 0, 3, 8 and 9 don't occur at all.

**Hint:**

- Declare an array, freq of size 10 and initialize it with zeros, which will be used to count the frequencies of each of the digit occurring.
- Given a string, s, iterate through each of the character in the string. Check if the current character is a number or not.
- If the current character is a number, increase the frequency of that position in the freq array by 1.
- Once done with the iteration over the string, s, in a new line print all the 10 frequencies starting from 0 to 9, separated by spaces.

**Program:**

```
#include <stdio.h>
int main()
{
    char str[1000];
    scanf("%s", str);
    int hash[10] = {0,0,0,0,0,0,0,0,0,0};
    int temp;
    for(int i=0; str[i] != '\0'; i++)
    {
        temp = str[i] - '0';
        if(temp <= 9 & temp >= 0)
        {
            hash[temp]++;
        }
    }
    for(int i=0; i <= 9; i++)
    {
        printf("%d", hash[i]);
    }
    return 0;
}
```

Ex. No. :

Date :

**Monk Takes a Walk****Problem Statement:**

Today, Monk went for a walk in a garden. There are many trees in the garden and each tree has an English alphabet on it. While Monk was walking, he noticed that all trees with vowels on it are not in good state. He decided to take care of them. So, he asked you to tell him the count of such trees in the garden.

**Note:** The following letters are vowels: 'A', 'E', 'I', 'O', 'U', 'a', 'e', 'i', 'o' and 'u'.

**Input Format:**

The first line consists of an integer T denoting the number of test cases.

Each test case consists of only one string, each character of string denoting the alphabet (may be lowercase or uppercase) on a tree in the garden.

**Output Format:**

For each test case, print the count in a new line.

**Constraints:**

$1 \leq T \leq 10$

$1 \leq \text{length of string} \leq 105$

**Sample Input**

```
2
nBBZLaosnm
JHkIsnZtTL
```

**Sample Output**

```
2
1
```

**Explanation**

In test case 1, a and o are the only vowels. So, count=2

**Brief Description:** Given a string S you have to count number of vowels in the string.

**Solution 1:**

For each vowel, count how many times it is appearing in the string S. Final answer will be the sum of frequencies of all the vowels.

**Solution 2:**

Iterate over all the characters in the string S and use a counter (variable) to keep track of number of vowels in the string S. While iterating over the characters, if we encounter a vowel, we will increase the counter by 1.

**Time Complexity:**  $O(N)$  where N is the length of the string S. **Space Complexity:**  $O(1)$

**Program:**

```
#include <stdio.h>

int main()
{
    int t;
    scanf("%d", &t);
    while(t--)
    {
        char str[100000];
        int count = 0;
        scanf("%s", str);
        for(int i=0; str[i] != '\0'; i++)
        {
            char c = str[i];
            if((c == 'a') || (c == 'e') || (c == 'i') || (c == 'o')
                || (c == 'u') || (c == 'A') || (c == 'E') || (c == 'I')
                || (c == 'O') || (c == 'U'))
                count++;
        }
        printf("%d\n", count);
    }
    return 0;
}
```

**Ex. No. :****Date :****What is your mobile number?****Problem Statement:**

These days Bechan Chacha is depressed because his crush gave him list of mobile number some of them are valid and some of them are invalid. Bechan Chacha has special power that he can pick his crush number only if he has valid set of mobile numbers. Help him to determine the valid numbers.

You are given a string "S" and you have to determine whether it is Valid mobile number or not. Mobile number is valid only if it is of length 10 , consists of numeric values and it shouldn't have prefix zeroes.

**Input Format:**

First line of input is T representing total number of test cases.

Next T line each representing "S" as described in in problem statement.

**Output Format:**

Print "YES" if it is valid mobile number else print "NO".

Note: Quotes are for clarity.

**Constraints:**

$1 \leq T \leq 103$

sum of string length  $\leq 105$

**Sample Input**

3

1234567890

0123456789

0123456.87

**Sample Output**

YES

NO

NO

**Program:**

```

#include <stdio.h>
#include <string.h>
int main()
{
    int t;
    scanf("%d", &t);
    while (t--) {
        int flag = 1;
        char s[100000];
        scanf("%s", s);
        int k = strlen(s);
        if (k == 10)
        {
            for (int i = 0; i < 10; i++)
            {
                if (s[i] == '0')
                {
                    flag = 0;
                    break;
                }
                if (s[i] < '0' || s[i] > '9')
                {
                    flag = 0;
                    break;
                }
            }
        }
    }
}

```

```
else
    flag = 0;
if (flag == 1)
    printf("YES\n");
else
    printf("NO\n");
}
return 0;
```

3