**MOCK TEST CSED**

Q1:Unique cells in a binary matrix

Given a matrix of size n x m consisting of 0’s and 1’s.

write a program to find number of unique cells with value 1 such that corresponding entire row and entire column does not have another 1

**Input Format:**

First Input: row size

Second Input: column size

Third input: matrix values

**Input matrix**

3

4

0 **1** 0 0

0 0 **1** 0

1 0 0 1

**Output:**

2

**Explanation of the output:**

The two 1s that are unique in their rows and columns are highlighted.

**CODE:**

#include<stdio.h>

int main()

{

int m,n,i,j,flag,c2=0,b,c,k,l;

scanf("%d%d",&m,&n);

int a[m][n];

for (i=0;i<m;i++)

{

for (j=0;j<n;j++)

scanf("%d",&a[i][j]);

}

for (i=0;i<m;i++)

{

flag=0;

for (j=0;j<n;j++)

{

if (a[i][j]==1)

{

b=i;

c=j;

break;

}

}

for (k=0;k<m;k++)

{

if (a[k][c]==1&&k!=i)

flag=1;

}

for (l=0;l<n;l++)

{

if (a[b][l]==1&&l!=j)

flag=1;

}

if (flag==0)

c2++;

}

printf("%d",c2);

}

Q2: Given a string, the task is to encrypt the string using ! and @ symbols, alternatively. While encrypting the message the encrypted format must repeat the symbol as many times as the letter position in Alphabetical order.

Examples:

**Input**

Ab

**Output**

!@@

Explanation:

Position of 'A' in alphabetical order is 1 and in String is odd position, so encrypted message will have 1 '!'

Position of 'b' in alphabetical order is 2 and in String is even position, so encrypted message will have 2 '@'

Therefore, the output "!@@"

**Input:**

CDE

**Output:**

!!!@@@@!!!!!

**CODE:**

#include<stdio.h>

#include<string.h>

int main()

{

int i,r,len,j;

char s[50];

gets(s);

len=strlen(s);

for (i=0;i<len;i++)

{

if (s[i]>='a'&&s[i]<='z')

{

if (i%2==0)

{

r=s[i]-96;

for (j=0;j<r;j++)

printf("!");

}

else

{

r=s[i]-96;

for (j=0;j<r;j++)

printf("@");

}

}

if (s[i]>='A'&&s[i]<='Z')

{

if (i%2==0)

{

r=s[i]-64;

for (j=0;j<r;j++)

printf("!");

}

else

{

r=s[i]-64;

for (j=0;j<r;j++)

printf("@");

}

}

}

}