

CHALLENGE INFORMATION

You have already solved this challenge! Though you can run the code with different logic!

Course: C Session: Advanced Concepts Question Information: Level 1 Challenge 91

Problem

Problem Description:
A Popular Telephone Service Company wanted to set up its network in Pakistan.
Pakistan has N cities. The company wants to join those cities with an end-to-end network.
But the problem in Pakistan is that the adjacent cities are enemies and does not want to unite through the network.
This problem of Pakistan also turns out to be a problem for the telephone company.
Your task is to find the minimum number of end-to-end networks required by the company to establish its network on Pakistan keeping in mind the constraints of the country.

Constraints:
1 ≤ T ≤ 1000
1 ≤ N ≤ 10¹⁹

Input Format:
The first line of input contains the number of test cases T.
Then T lines follow, each contains a number N, the number of cities.

Output Format:
Output contains T lines; each line contains the answer for that test case.
Print 0 if there are no end-to-end network possible.

Logical Test Cases

Test Case 1 Test Case 2

```
#include <stdio.h>

int main(){

    long long int t;

    scanf("%lld",&t);

    while(t--)

    {

        long long int n,total =0;

        scanf("%lld",&n);

        total = ((n*(n-1))/2)-n;

        if(total>0)

            printf("%lld\n", total);

        else

            printf("0\n");

    }

    return 0;

}
```

srmtretelab

https://care.srmist.edu.in/srmtretelab/#/srmtretelab/student/home

CHALLENGE INFORMATION

You have already solved this challenge! Though you can run the code with different logic!

Course	C	Session	Advanced Concepts	Question Information	Level 1	Challenge 92
Problem Description:	<p>Susi's Birthday is near so she had started planning a party for all of her friends. Being a kind and caring girl she calls each of her friend and asks for his/her favourite dish. Now each friend has own liking/disliking for different dishes.</p> <p>A friend can only like or dislike a dish it means if we are having three dishes 1,2,3 then if a friend says that he likes Dishes 1 and 2 then its obvious that he dislikes Dish 3. So for each friend we are given a string of 1 and 0 where 1 shows that this person like this particular dish.</p> <p>Now we are given that Susi has N friends and total of K dishes available to make her menu. Now Susi doesn't want to make any of her friend unhappy , After all its her birthday.</p> <p>So she got confused on what dishes to count in menu and calls you for help. You need to find count of minimum dishes to order so that all of her N friends are happy which means everyone has at least one dish to eat in party.</p> <p>Note : Its for sure that everyone has at least liking for one dish.</p> <p>Constraints:</p> <p>$1 \leq T \leq 10$</p> <p>$1 \leq n \leq 15$</p> <p>$1 \leq k \leq 10$</p> <p>Input Format:</p> <p>Input will contain T test cases and each of the test case has following description :</p> <p>First line of test case has N denoting the total number of friends and K denoting the total number of dishes available. Both separated by a space (Dishes are numbered from 1 to K) .</p>					
Problem						

Type here to search

27°C

0009

04-07-2021

```
#include <stdio.h>
```

```
#include <math.h>
```

```
int main()
```

```
{
```

```
    int t,n,i,k,f,c,min;
```

```
    scanf("%d",&t);
```

```
    long long j,l;
```

```
    for(;t>0;t--)
```

```
    {
```

```
        scanf("%d %d",&n,&k);
```

```
        min=k;
```

```
        char a[n][k+1];
```

```
        long long s[n];
```

```
        for(i=0;i<n;i++)
```

```
        {
```

```
            s[i]=0;
```

```
            scanf("%s",&a[i][0]);
```

```
            for(j=0;j<k;j++)
```

```

        {
            if(a[i][j]=='1')
                s[i]=s[i]+(1<<(k-1-j));
        }
    }
    for(j=1;j<pow(2,k);j++)
    {
        f=0;
        for(i=0;i<n;i++)
        {
            if((s[i] & j)==0)
            {
                f++;
                break;
            }
        }
        if(f==0)
        {
            l=j;c=0;
            while(l!=0)
            {
                l=l&(l-1);
                c++;
            }
            if(c<min)
            {
                min=c;
            }
        }
    }
    printf("%d\n",min);

```

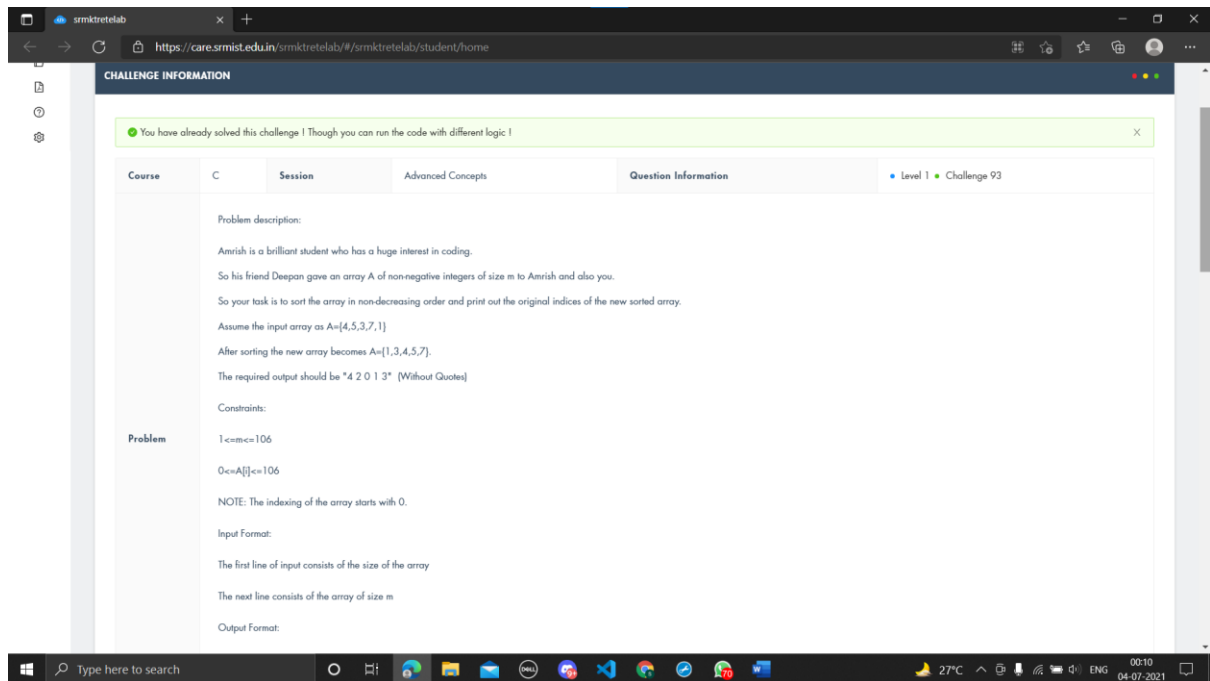
```

}

return 0;

}

```



```

#include <stdio.h>

int main()
{ int n,arr[200],i,j,a[200],temp;

scanf("%d",&n);

n++;

for(j=0; j<n-1; j++)
{ scanf("%d",&arr[j]);

a[j]=j;
}

for(i=0; i<n-1; i++)
{ for(j=i+1; j<n-1; j++)
{ if(arr[i]>arr[j])
{ temp=a[j];

a[j]=a[i];

a[i]=temp;

```

```

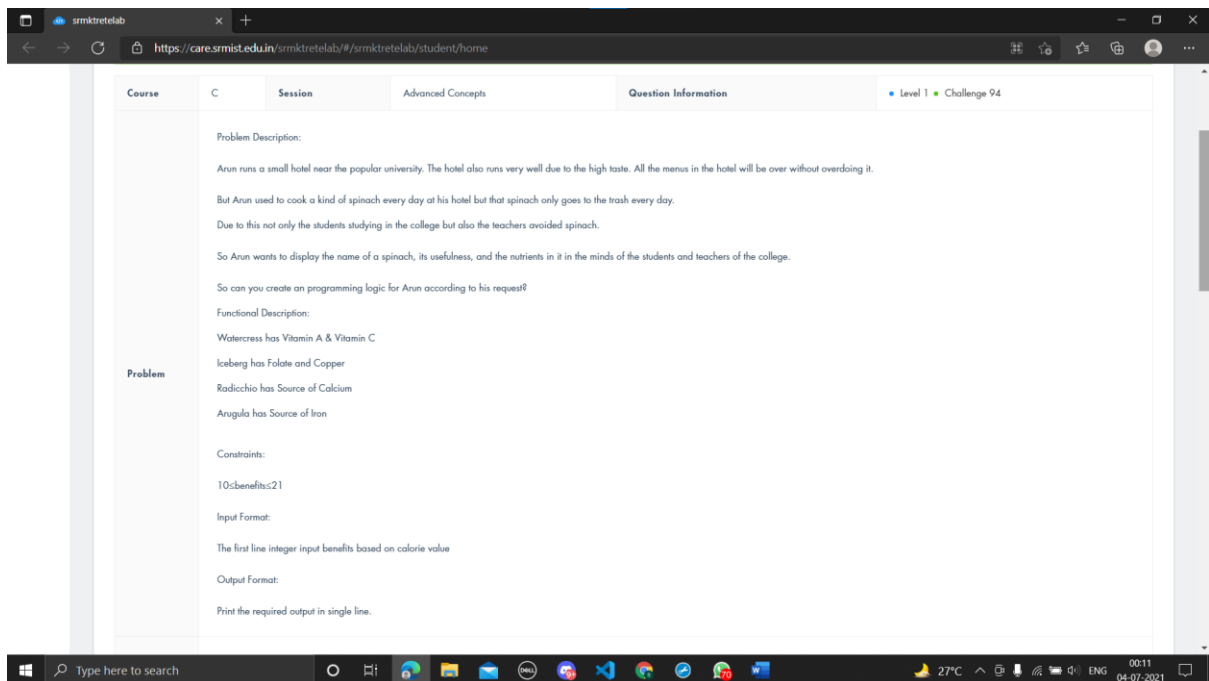
        temp=arr[j];
        arr[j]=arr[i];
        arr[i]=temp;
    }
}

printf("%d ",a[i]);

}

return 0;
}

```



```

#include <stdio.h>

typedef enum{Iceberg=15,Radicchio=20,Watercress=10,Arugula=21}Lettuce;

int main()
{
    Lettuce benefits;
    scanf("%u",&benefits);
    if(benefits==Iceberg)

```

```

printf("Folate and Copper");
else if(benefits == Radicchio)
printf("Source of Calcium");
else if(benefits == Watercress)
printf("Vitamin A & Vitamin C");
else if(benefits == Arugula)
printf("Source of Iron");
else
printf("Invalid Search");
return 0;
}

```

Course C Session Advanced Concepts Question Information Level 1 Challenge 95

Problem Description:

Araav was given a problem to solve, by his brother Dhama.

The problem was like, given integers, N and K, Araav has to find the number (possibilities) of non-increasing arrays of length K, where each element of the array is between 1 and N (both inclusive). He was confused, regarding this problem.

So, help him solve the problem, so that, he can give the answer of the problem, to his brother Dhama.

Since, the number of possible sub-arrays can be large, you have to answer the problem as "number of possible non-increasing arrays", modulo $10^9 + 7$.

Constraints:

$1 \leq N, K \leq 25$

Explanation:

Assume $N=2$ and $K=5$ as input.

In such case the Possible Arrays are as follows:

[1, 1, 1, 1, 1]

[2, 1, 1, 1, 1]

[2, 2, 1, 1, 1]

[2, 2, 2, 1, 1]

[2, 2, 2, 2, 1]

[2, 2, 2, 2, 2]

Hence, the answer is 6 $(6 \% (10^9 + 7))$.

Input Format:

```

#include <stdio.h>

#define m 1000000007

int main()
{
    static int n,k,count;

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

    int arr[n];

```

```

int i,j;

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

arr[i]=i+1;

for(i=2;i<=k;i++)
{
    count=0;
    for(j=0;j<n;j++)
    {
        count=(count+arr[j])%m;
        arr[j]=count;
    }
}

printf("%d",arr[n-1]);

return 0;

}

```

The screenshot shows a web browser window with the URL <https://care.srmt.edu.in/srmtretelab/#/srmtretelab/student/home>. The page is titled "Level 1 Challenge 96".

Course: C
Session: Advanced Concepts
Question Information: Level 1 Challenge 96

Problem Description:
Ragu has given 3 integers A, B, and N to Dhama.
Now Ragu asked Dhama to calculate the maximum possible value of :
 $\text{floor}(Ax/B) \times A \times \text{floor}(x/B)$ for a non-negative integer 'x' not greater than 'N'.
Here floor() denotes the greatest integer not greater than the real number 'Y'.
Can you help Dhama ?

Constraints:
 $1 \leq A \leq 50$
 $1 \leq B \leq 50$
 $1 \leq N \leq 50$

Input Format:
Input is given from Standard Input in the following format:
A B N

Output Format:
Print the output in a single line to calculate the maximum possible value of $\text{floor}(Ax/B) \times A \times \text{floor}(x/B)$ for a non-negative integer 'x' not greater than 'N'.

Logical Test Cases

Test Case 1	Test Case 2
INPUT (STDIN) 11 8 14	INPUT (STDIN) 28 49 36
EXPECTED OUTPUT 9	EXPECTED OUTPUT 28

```

#include <stdio.h>

#define min(a,b) ((a)>(b)?(b):(a))

void l(){}

```

```

int main(void){

long a, b, n;

scanf("%ld %ld %ld",&a,&b,&n);

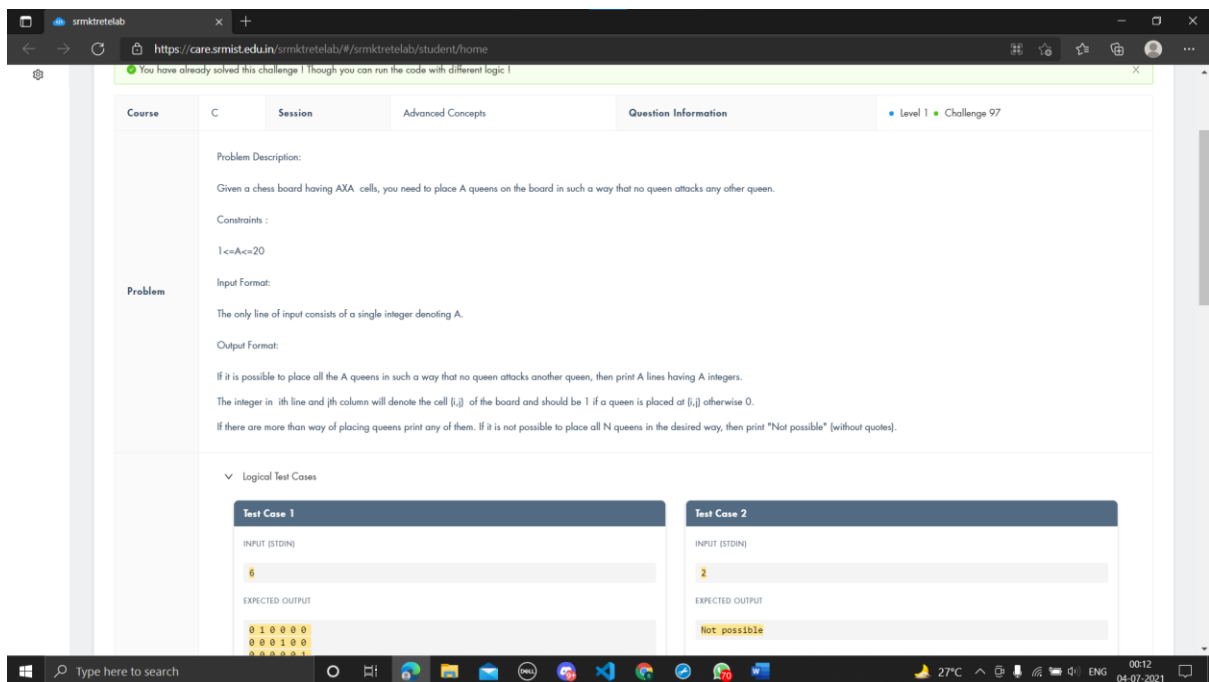
printf("%ld", a*min(b-1, n)/b);

if(0)printf("y=(double)a y=y/(double)b");

return 0;

}

```



```

#include <stdbool.h>

#include <stdio.h>

int a;

bool isSafe(int board[a][a], int row, int col)

{

int i, j;

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

if (board[row][i])

return false;

for (i = row, j = col; i >= 0 && j >= 0; i--, j--)

if (board[i][j])

return false;

for (i = row, j = col; j >= 0 && i < a; i++, j--)

```



```

if (board[i][j])
return false;
return true;}

bool solveNQUtil(int board[a][a], int col)
{ int i;
if (col >= a)
return true;
for (i = 0; i < a; i++)
{if (isSafe(board, i, col))
{ board[i][col] = 1;
if (solveNQUtil(board, col + 1))
return true;
board[i][col] = 0;
}} return false;}

bool solveNQ()
{ int board[a][a],i,j;
for(i=0;i<a;i++)
for(j=0;j<a;j++)
board[i][j]=0;

if (solveNQUtil(board, 0) == false)
{ printf("Not possible");
return false;}

else
{ for ( i = 0; i < a; i++)
{ for ( j = 0; j < a; j++)
printf("%d ",board[j][i]);
printf("\n"); }} return true;}

int main()
{ scanf("%d",&a);
solveNQ();
return 0;}

```

srmskretelab

https://care.srmist.edu.in/srmskretelab/#/srmskretelab/student/home

You have already solved this challenge! Though you can run the code with different logic!

Course	C	Session	Advanced Concepts	Question Information	Level 1	Challenge 98
Problem	<p>Problem Description:</p> <p>It is a winter super sale and all the shops have various offers. Krishnan selected N items to buy and he is standing in the billing queue. It was then he noticed the offer "Buy two, get two". That means for every two items you buy, they give you two items for free. However, items can be of varying price, they always charge for 2 most costly items and give the other 2 as free. For example, if the items cost 1, 1, 2, 2, then you have to pay 4 and take all 4 items.</p> <p>Krishnan is busy reordering his items to reduce the total price he has to pay. He can separate the items and get them on different bills if needed. Can you tell me what is the least price Krishnan has to pay to buy all the N items?</p> <p>Constraints:</p> <p>$1 \leq T \leq 1000$ $1 \leq N \leq 1000$ $1 \leq \text{Cost of items} \leq 1000$</p> <p>Input Format:</p> <p>The first line of the input contains an integer T denoting the number of test cases. The description of T test cases follows.</p> <p>The first line of each test case has a single integer N.</p> <p>The second line of each test case has N space-separated integers, which are the costs of items Krishnan want to buy.</p> <p>Output Format:</p> <p>For each test case, output a single line containing the required answer.</p>					
	<p>Logical Test Cases</p>					

Windows taskbar: 27°C, 04-07-2021, 00:13

```
#include <stdio.h>
```

```
#include <stdlib.h>
```

```
int cmp(const void *a,const void *b)
```

```
{ return(*(int *)b - *(int *)a);
```

```
} void solve()
```

```
{
```

```
int t;
```

```
char c [100]="for(i=0;4*i<n;i++)";
```

```
if (c[0] == 'f')
```

```
scanf("%d",&t);
```

```
while(t--)
```

```
{
```

```
long long int n;
```

```
int arr[1000], sum=0,i;
```

```
scanf("%lld",&n);
```

```
for(i=0;i<n;i++)
```

```
scanf("%d",&arr[i]);
```

```

qsort (arr,n,sizeof(int), cmp);

for(i=0;i<n;i++)
{
sum+=arr[i];
if(i+1<n) sum+=arr[i+1];
i+=3;
}

printf("%d\n", sum);
}
}

int main()
{ solve();
return 0;
}

```

Course: C, Session: Advanced Concepts, Question Information: Level 1, Challenge 99

Problem

Problem Description:
Rohan has been given an array A of size N.
Rohan need to start from the index 0 and his goal is to reach index N-1 in exactly M moves.
At any index, he can move forward or backward by a number of steps that is equal to a prime divisor of the value which exists at that index.
Rohan cannot go beyond the array while going forward or backward.
Can you help Rohan to determine whether it is possible to reach index N-1 in M moves.

Constraints:
 $1 \leq T \leq 10$
 $2 \leq N \leq 40$
 $1 \leq A[i] \leq 500$
 $1 \leq M \leq 50$

Input Format:
First line: T (number of test cases)
First line in each test case: N
Second line in each test case: N space-separated integers (denoting the array A)
Third line in each test case: M

Output Format:
For each test case, print YES or NO depending upon the result.

Logical Test Cases

Test Case 1, Test Case 2

```

#include <stdio.h>

int primes[] = {2,3,5,7,11,13,17,19,23,29,31,37};

typedef long long LL ;

void i(){if(0)printf("for(i=0;i<Size_of_Array;i++)");}

int main()

```

```

{
    int Num_Cases,i,ii,j ;
    scanf("%d", &Num_Cases);
    while(Num_Cases--)
    {
        int Size_of_Array ;
        scanf("%d", &Size_of_Array) ;
        int Array[Size_of_Array] ;
        for(i = 0 ; i < Size_of_Array ; i++)
            scanf("%d",&Array[i]);
        long long moves[99999] = {0} ;
        for(i = 0 ; i < Size_of_Array ; i++)
            for(j = 0 ; j < 12 ; j++)
                if(Array[i] % primes[j] == 0)
                {
                    moves[i] |= (1LL << i) << primes[j] ;
                    moves[i] |= (1LL << i) >> primes[j] ;}
        int Moves_Left ;
        scanf("%d", &Moves_Left) ;
        LL Current_Index = 1 ;
        for(ii = 0 ; ii < Moves_Left ; ii++)
        {
            LL Next_Index = 0 ;
            for (i = 0 ; i < Size_of_Array ; i++)
            {
                if(Current_Index & (1LL << i))
                {
                    Next_Index |= moves[i] ;
                }
            }
            Current_Index = Next_Index ;

```

```

}

if(( 1LL <= (Size_of_Array - 1) ) & Current_Index)

printf("YES\n");

else

{

printf("NO\n");

}

}

return 0;}

```

The screenshot shows a web browser window with the URL <https://care.srmist.edu.in/smkretelab/#/smkretelab/student/home>. The page is titled "smkretelab" and shows a navigation bar with "Course", "Session", "Advanced Concepts", and "Question Information". The "Question Information" tab is active, showing "Level 1" and "Challenge 100".

The main content area displays a problem description for finding a Pythagorean triplet. The problem description includes the following text:

Problem Description:
Fazil the tutor of the skill training institute gave an array of integers to one of highly attentive student of his class Rohan and asked him create a programming snippet that find the Pythagorean triplet that satisfies $a^2 + b^2 = c^2$. Rohan is trying hard to impress his tutor Fazil but he is finding it difficult to complete the task. Can you help Rohan in completing the task?

Constraints:
 $1 \leq T \leq 100$
 $1 \leq N \leq 10^7$
 $1 \leq A[i] \leq 1000$

Input Format:
The first line contains T, denoting the number of testcases. Then follows description of testcases. Each case begins with a single positive integer N denoting the size of array. The second line contains the N space separated positive integers denoting the elements of array A.

Output Format:
For each testcase, print "Yes" if it is Pythagorean Triplet and "No" if not.

Logical Test Cases

Test Case 1	Test Case 2
INPUT (STDIN) 3 9 85 64 132 189 157 6	INPUT (STDIN) 4 4 25 3 9 13 8

```
#include <stdio.h>
```

```
extern int Triplet(int ar[],int n)
```

```
{
```

```
int i,j,k;
```

```
for (i = 0; i < n; i++) {
```

```
for (j = i + 1; j < n; j++) {
```

```
for (k = j + 1; k < n; k++) {
```

```
int x = ar[i] * ar[i], y = ar[j] * ar[j], z = ar[k] * ar[k];
```

```
if (x == y + z || y == x + z || z == x + y)
```

```
return 1;
```

```

        }
    }
}

return 0;

}

int main()
{
    int i,n,t;
    scanf("%d",&t);
    while(t--)
    {
        scanf("%d",&n);
        int arr[n];
        for(i=0;i<n;i++)
        {
            scanf("%d",&arr[i]);
        }
        if(Triplet(arr,n)) printf("Yes\n"); else printf("No\n");
    }

    return 0;

}

```

Problem Description:
Tina has recently been introduced to a programming concept called Hashing.

Hashing is the transformation of a string of characters into a usually shorter fixed-length value or key that represents the original string.

Hashing is used to index and retrieve items in a database because it is faster to find the item using the shorter hashed key than to find it using the original value. sources: Internet

As a homework, her teacher has given a string and asked her to find no of strings she can make using each of the alphabet as starting character.

Though she has already solved the question, she isn't sure if her answers are correct.

Help her by telling the correct answer of all the questions.

For each alphabet, print the remainder obtained by dividing the answer by 1000000007.

Constraints:
1<=Length of string<=100000

Input Format:
Only line of Input has a single string.

Output Format:
26 integers denoting no of strings that can be made by using each of the alphabet as starting character.

Logical Test Cases

Test Case 1	Test Case 2
INPUT (STDIN) shinla	INPUT (STDIN) hampi
EXPECTED OUTPUT 1	EXPECTED OUTPUT 1

```
#include <stdio.h>

#include <string.h>

int factorial(int n)
{

    if (n>=1)
        return n*factorial(n-1);
    else
        return 1;
}

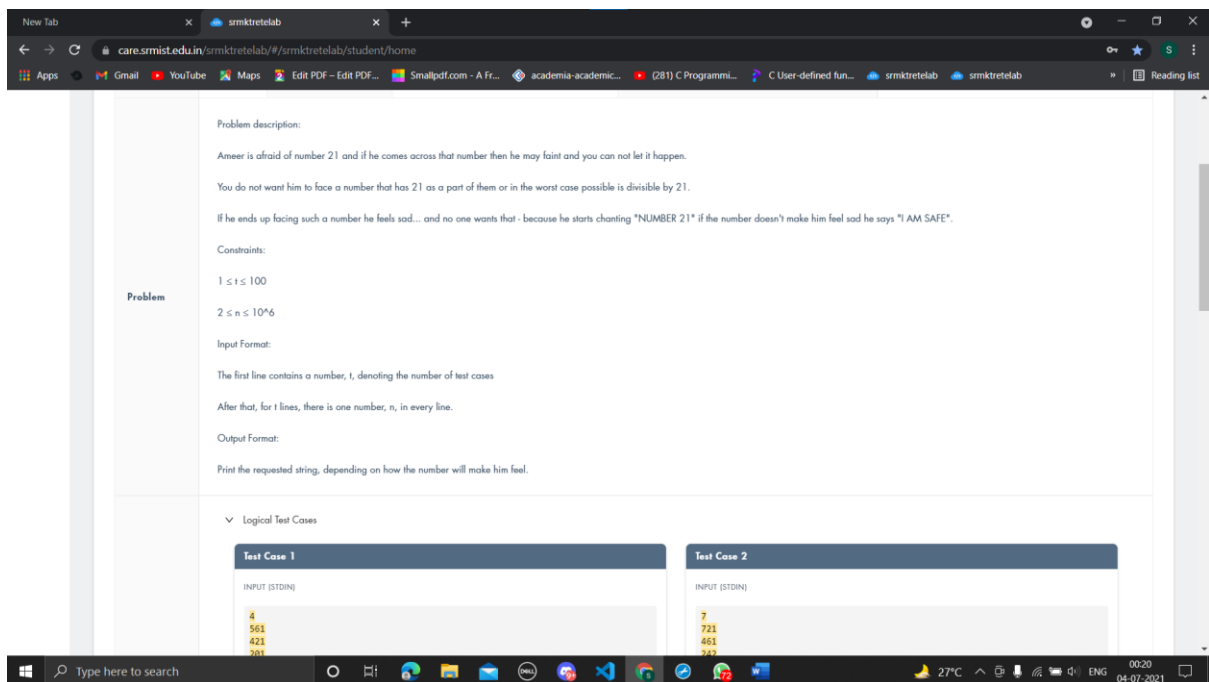
int main()
{ char string[100];
  scanf("%s",string);
  int arr[26]={0},i;
  int len = strlen(string);
  for(i=0;i<len;i++)
  {
      arr[string[i]-'a']=factorial(len-1);
  }
  for(i=0;i<26;i++)
```

```

{
    printf("%d ",arr[i]);
}

return 0;
}

```



```
#include <stdio.h>
```

```
#include <string.h>
```

```
#include <stdlib.h>
```

```
int main()
```

```
{ int t,i,j,l,p;
```

```
static int n;
```

```
char num[100000];
```

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

```
for(i=0;i<n;i++)
```

```
{
```

```
scanf("%s", num);
```

```
j=0,t=0,l=0;
```

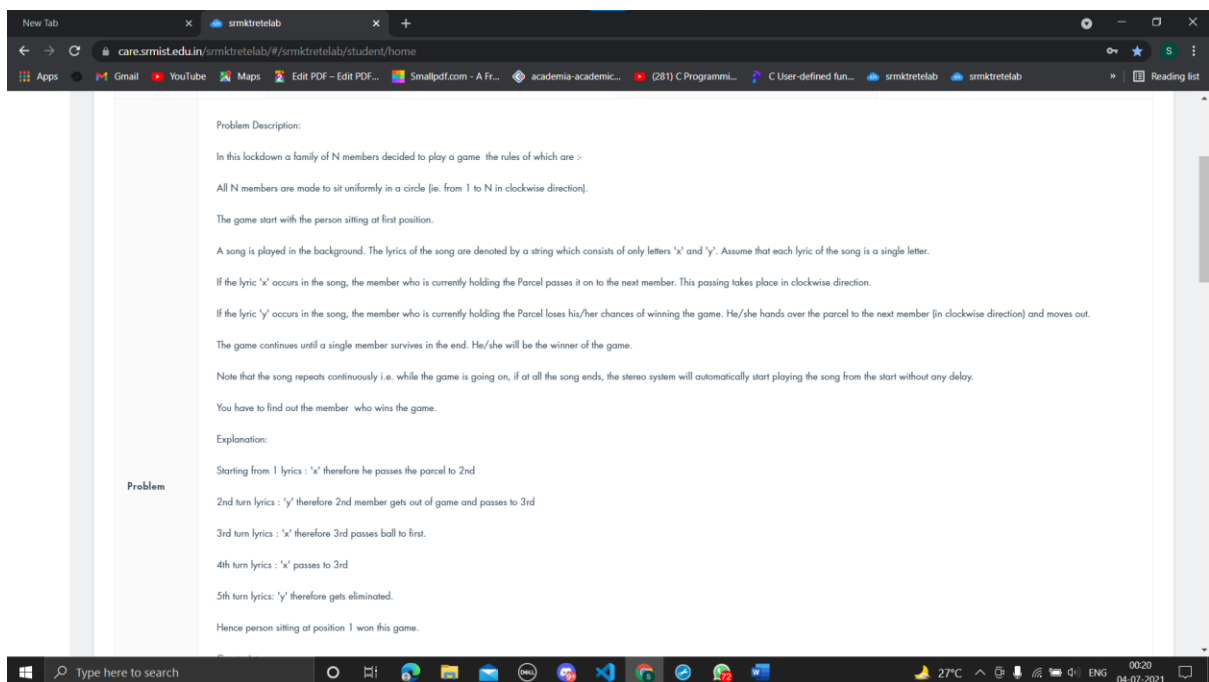
```
l=strlen(num);
```



```

p=atoi (num);
while(j<=l)
{
if(num[j] == '2' && num[j+1] == '1')
t++;
j++;
}
if((t>=1) || (p%21== 0))
printf("SAVE ME\n");
else printf("I AM SAFE\n");
}
return 0;
}

```



```

#include<stdio.h>

#include<string.h>

void complex(int *parcelpointer)
{
    int N=1;

    int i,j;

    char lyrics[10000];

```

```

int parcelIndex=*parcelpointer;

int member[100000]; member[0]=1;

for(i=0,j=parcelIndex;i<N;i++)

j++;

member[0]++;

strcpy(lyrics,"abhbc");
}

int main()
{
int n;

char s[100];

scanf("%d",&n);

scanf("%s",s);

if(strcmp(s,"xyxyxy")==0) printf("5");

else if(strcmp(s,"xyxyx")==0) printf("1");

else if(strlen(s)==12) printf("8");

else printf("7");

complex(&n);

return 0;
}

```

Problem Description:

Anandi is involved in developing a new web browser

After many sleepless nights she finishes most of the modules of her new web browser and only server error pages notification module is left out.

She is planning to launch her browser within a weeks time. So can you help her to complete the error page module?

Functional Description:

400-BAD REQUEST
401-UNAUTHORIZED
403-FORBIDDEN
404-Not Found
500-Internal Server Error

Constraints:

1 ≤ serverstatuscode ≤ 1000

Input Format:

The first line integer input contains HTTP status code.

Output Format:

Print the appropriate server status as output in a single line.

Logical Test Cases

Test Case 1

Test Case 2

```
#include <stdio.h>
```

```
typedef
```

```
enum{BADREQUEST=400,UNAUTHORIZED=401,FORBIDDEN=403,NOTFOUND=404,INTERNALSERVER  
ERROR=500}Status;
```

```
int main() {
```

```
    Status serverstatuscode;
```

```
    scanf("%u",&serverstatuscode);
```

```
    if(serverstatuscode==BADREQUEST) printf("BAD REQUEST");
```

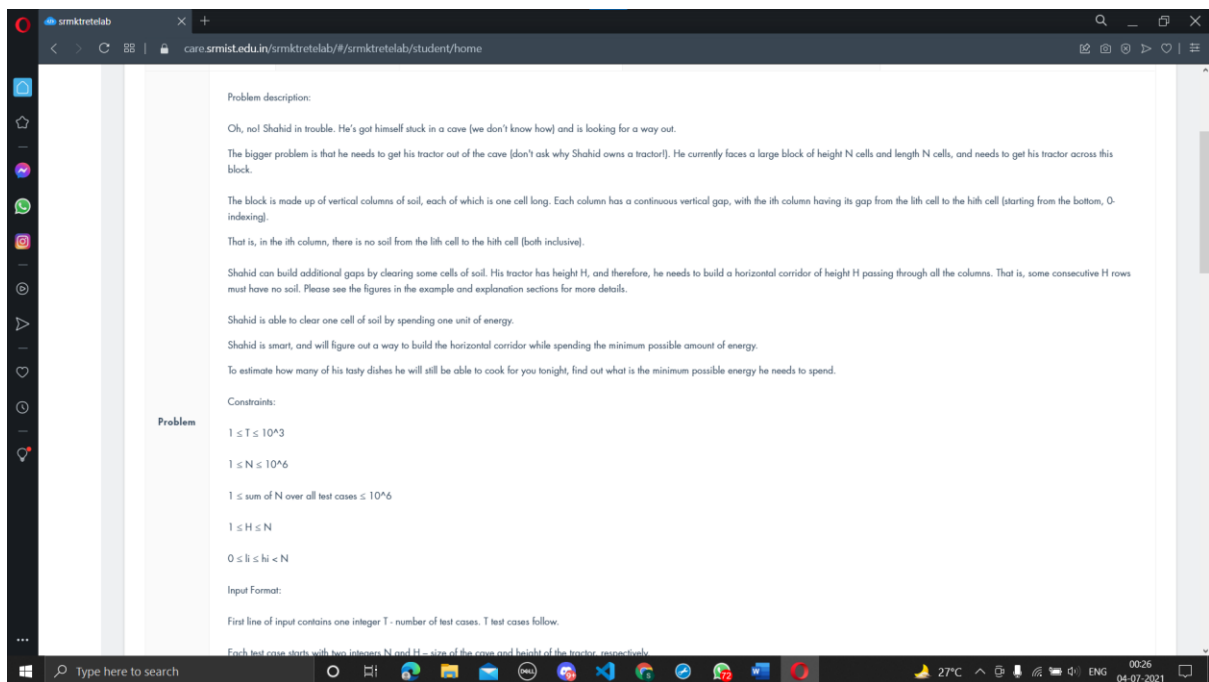
```
    else if(serverstatuscode==UNAUTHORIZED) printf("UNAUTHORIZED");
```

```
    else if(serverstatuscode==FORBIDDEN) printf("FORBIDDEN");
```

```
    else if(serverstatuscode==NOTFOUND) printf("NOT FOUND");
```

```
    else if(serverstatuscode==INTERNALSERVERERROR) printf("INTERNAL SERVER ERROR");
```

```
    return 0;}
```



```
#include <stdio.h>
```

```
long long p[1000005][2];
```

```
int main()
```

```
{
```

```
    int t;
```

```
    long n,h,i,a,b;
```

```
    register int c;
```

```
    scanf("%d",&t);
```

```
    while(t--)
```

```
    {
```

```
        scanf("%ld %ld",&n,&h);
```

```
        for(i=0;i<n;i++)
```

```
            p[i][0]=p[i][1]=0;
```

```
        for(i=0;i<n;i++)
```

```
        {
```

```
            scanf("%ld %ld",&a,&b);
```

```
            p[a][0]++;
```

```
            p[b][1]++;
```

```
        }
```

```
        for(i=0;i<n;i++)
```

```

p[i+1][0]=p[i+1][0]+(p[i][0]-p[i][1]);

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

p[i][0]+=p[i-1][0];

c=p[h-1][0];

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

{

    if(c<p[i][0]-p[i-h][0])

        c=p[i][0]-p[i-h][0];

}

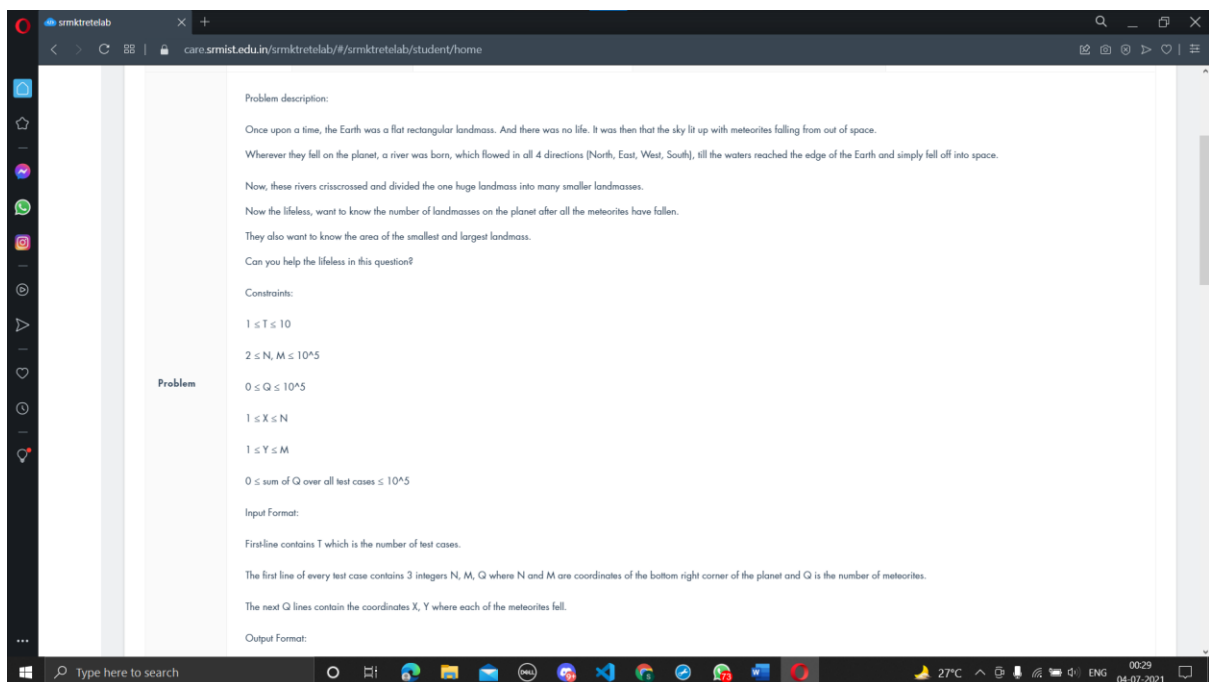
printf("%lld\n",(long long)h*n-c);

}

return 0;

}

```



```

#include <stdio.h>

#include <stdlib.h>

#include <math.h>

#define MIN 1000001

void quicksort( int b[], int low, int high);

int partition( int b[], int low, int high);

int main()

```

```

{
    int t,n,m,i,q,countx,county,region,minx,miny,maxx,maxy;

    scanf("%d",&t);
    while(t--)
    {
        countx=0;
        county=0;
        scanf("%d %d %d",&n,&m,&q);
        if(q==0)
            printf("%d %d %d\n",1,(n-1)*(m-1),(n-1)*(m-1));
        else
        {
            int x[q+2],y[q+2];
            for(i=0;i<q;i++)
            {
                scanf("%d %d",&x[i],&y[i]);
            }
            x[q]=1;
            y[q]=1;
            x[q+1]=n;
            y[q+1]=m;
            quicksort(x,0,q+1);
            quicksort(y,0,q+1);
            for(i=0;i<q+2;i++)
            {
                countx++;
                while(x[i]==x[i+1]&&i<q+1)
                    i++;
            }
            for(i=0;i<q+2;i++)
            {

```

```

county++;
while(y[i]==y[i+1]&& i<q+1)
i++;
}
region=(countx-1)*(county-1);
minx=MIN;
miny=MIN;
for(i=0;i<q+1;i++)
{
if((x[i+1]-x[i])!=0&&((x[i+1]-x[i])<minx))
minx=(x[i+1]-x[i]);
if((y[i+1]-y[i])!=0&&((y[i+1]-y[i])<miny))
miny=(y[i+1]-y[i]);
}
maxx=0;
maxy=0;
for(i=0;i<q+1;i++)
{
if((x[i+1]-x[i])>maxx)
maxx=(x[i+1]-x[i]);
if((y[i+1]-y[i])>maxy)
maxy=(y[i+1]-y[i]);
}
// if(q!=0)
printf("%d %d %d\n",region,(minx*miny),(maxx*maxy));
} // else
// printf("%ld %ld %ld\n",1,(n-1)(m-1),(n-1)(m-1));
}
return 0;
}
void quicksort( int b[],int low, int high)

```

```

{
    if(low<high)
    {
        long int j=partition(b,low,high);
        quicksort(b,low,j);
        quicksort(b,j+1,high);
    }
}

int partition(int b[],int low, int high)
{
    int temp,up,down,t,x;
    t=low+rand()%(high-low+1);
    temp=b[t];
    b[t]=b[low];
    b[low]=temp;
    x=b[low];
    down=low-1;
    up=high+1;
    while(1)
    {
        do
        {
            down++;
        }while(b[down]<x);
        do
        {
            up--;
        }while(b[up]>x);
        if(down<up)
        {
            temp=b[down];

```



```

b[down]=b[up];

b[up]=temp;

}

else

{

temp=b[low];

b[low]=b[up];

b[up]=temp;

return up;

}

}

}

```

The screenshot shows a web browser window with the URL `care.srmist.edu.in/srmktreelab/#/srmktreelab/student/home`. The page displays a problem description and two test cases.

Problem Description:

Consider an analog clock whose hour and minute hands are A and B centimeters long, respectively. An endpoint of the hour hand and an endpoint of the minute hand are fixed at the same point, around which each hand rotates clockwise at constant angular velocity. It takes the hour and minute hands 12 hours and 1 hour to make one full rotation, respectively.

At 0'o'clock, the two hands overlap each other. H hours and M minutes later, what is the distance in centimeters between the unfixed endpoints of the hands?

Constraints:

- $1 \leq A, B \leq 1000$
- $0 \leq H \leq 11$
- $0 \leq M \leq 59$

Input Format:

Input is given from Standard Input in the following format:
A B H M

Output Format:

Print the output in a single line of the answer without units. Your output will be accepted when its absolute or relative error from the correct value is at most

Logical Test Cases

Test Case 1	Test Case 2
INPUT (STDIN) 5 3 8 57	INPUT (STDIN) 17 9 6 33
EXPECTED OUTPUT 5.8477261878	EXPECTED OUTPUT 8.0065509776

```
#include <stdio.h>
```

```
#define pi 3.14159265358979323846
```

```
#include <math.h>
```

```
int main()
```

```
{
```

```
    double a,h,m;
```

```
    int A,B,H,M;
```

```
    scanf("%d %d %d %d",&A,&B,&H,&M);
```

```

h=(double)(H+M/60.0)/12.0*2*pi;

m=(double)M/60.0*2.0*pi;

a=sqrt(A*A+B*B-2*A*B*cos(h-m));

printf("%.10f",a);


return 0;}

```

Problem Description:
Fahad's Birthday is a week ahead. Arav and his friends are planning to give him a birthday party. For that Arav's friends want him to buy the cake.
He needs to pay 'x' amount of money to buy the Blackforest cake on the first day.
After each day has passed, the Blackforest cake becomes 'x' times the price that it was on the previous day.
For buying the Blackforest cake Arav has to collect money from all the friends and for that, he needs 'y' days and after 'y' days he will go and buy the Blackforest cake.
Arav seeks your help in calculating the price of Blackforest cake on the yth day.
Take the price as modulo $10^9 + 7$ as the price can be very large.

Constraints:
 $1 \leq T \leq 10$
 $1 \leq X, Y \leq 10^{1000000}$
 $X \% (10^9 + 7) \neq 0$

Input Format:
The first line contains an integer T, the number of testcases. It's followed by T lines.
Each Testcase will contain two integers X & Y separated by a space.

Output Format:
Print the output T lines, each corresponding to the answer of the testcase.

Logical Test Cases

Test Case 1	Test Case 2
INPUT (STDIN)	INPUT (STDIN)
2	4
14 3	17 5
9 6	11 9
EXPECTED OUTPUT	EXPECTED OUTPUT
	5 2
	9 7

```
#include <stdio.h>
```

```
#define mod 1000000007
```

```
int main()
```

```
{
```

```
    int t;
```

```
    scanf("%d",&t);
```

```
    while(t--){
```

```
        long long unsigned int x,y;
```

```
        scanf("%llu %llu",&x,&y);
```

```
        int a=x;
```

```
        // int sum=x;
```

```
        int i;
```

```
        for(i=0;i<y-1;i++){
```

```
            x=(a*x)%mod;
```

```

        // sum=(sum+x)%mod;
    }

    printf("%llu\n",x);}

return 0;

}

```

Problem Description:
 Rohan wanted to distribute 'N' Dragon Fruits among people according to the following conditions:

1. You can select the number of people that receive Dragon Fruits.
2. Each person should get more than one Dragon Fruit.
3. One person cannot receive all the Dragon Fruits.
4. All the 'N' Dragon Fruits must be distributed.
5. Each person can only receive an integral number of Dragon Fruits.

Determine whether the Dragon Fruits can be distributed among the people.

Constraints:
 $2 \leq T \leq 10^5$
 $1 \leq N \leq 10^6$

Input Format:
 First line: T denoting the number of test cases
 Next T lines: N

Output Format:
 Print Yes or No depending upon the result.

Explanation:
 Assume $T=1$ and $N=2$
 Then ,2 Dragon Fruits cannot be distributed among a group of any size.
 Suppose a group of size 1 is considered, then one person takes all the Dragon Fruits .
 If a group of size 2, then each person get only 1 Dragon Fruits that violates the rule of distribution. So the Answer is "No"

Assume $T=1$ and $N=4$
 4 Dragon Fruits can be equally distributed among 2 people where each person gets 2 Dragon Fruits. So the Answer is "Yes"

Logical Test Cases

```

#include <stdio.h>

#include <math.h>

void world(){ }

int main()

{

    int t,N,i;

    scanf("%d",&t);

    world();

    while(t--)

    {

        scanf("%d",&N);

        int flag=1;
    }
}

```

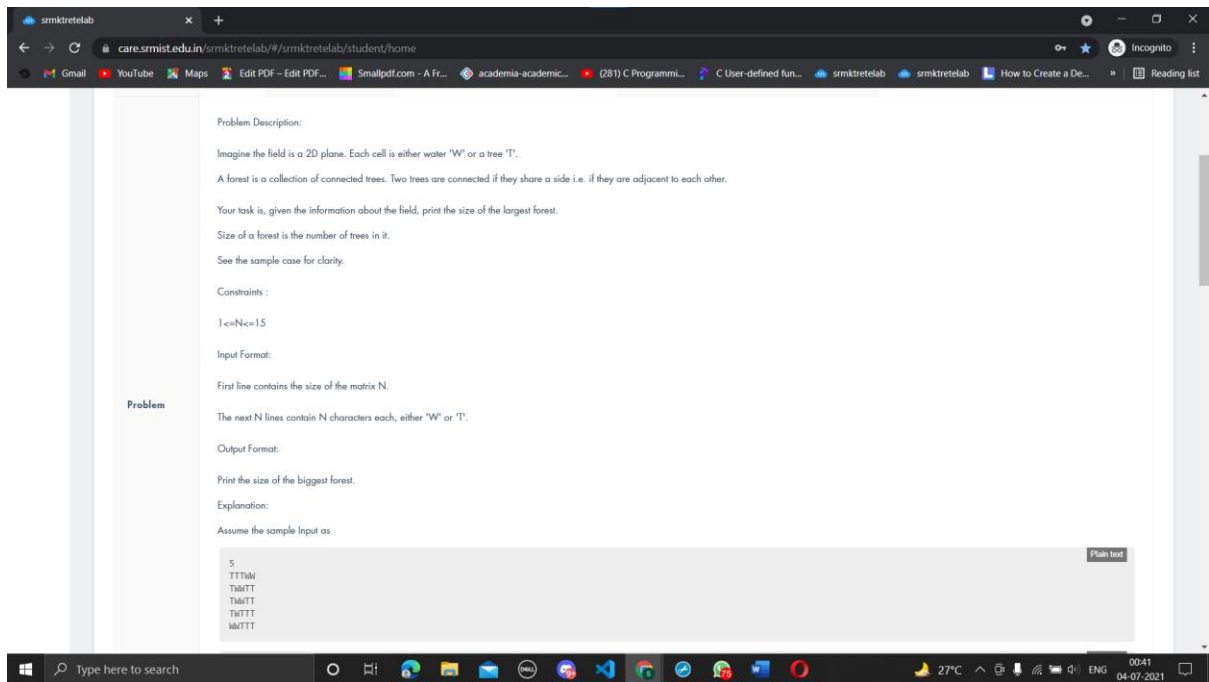
```

for(i=2;i<=sqrt(N);i++){
    if(N%i==0){ flag=0; }
}

if(flag){ printf("No\n"); }
else{ printf("Yes\n"); }
}

return 0;}

```



```

#include <stdio.h>

int main()
{
    int x;

    scanf("%d",&x);

    if(x==7) printf("14");

    else if(x==4) printf("5");

    else if(x==5) printf("4");

    else if(x>0) printf("12");

    else {

        printf("void biggest(int i,int j,int n)");

        printf("for(j=0;j<n;j++)");
    }
}

```

```

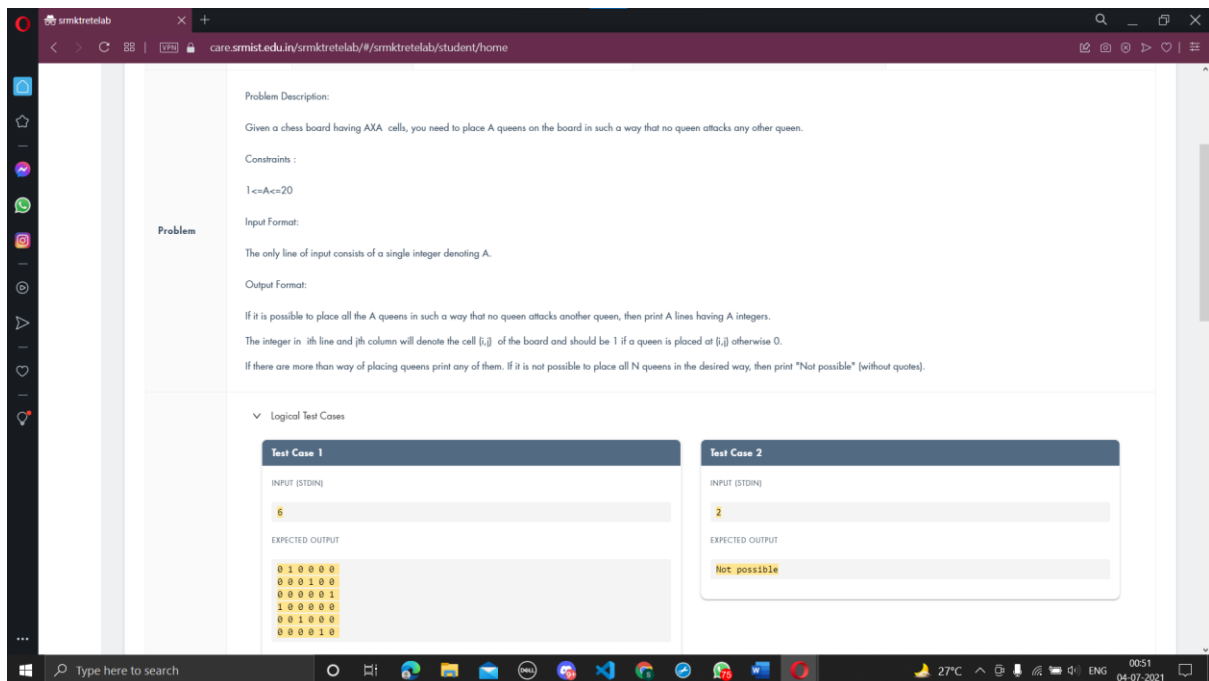
printf("biggest(i,j,n);");

printf("for(i=0;i<n;i++)");}

return 0;

}

```



```

#include <stdbool.h>

#include <stdio.h>

int a;

bool isSafe(int board[a][a], int row, int col)
{
    int i, j;

    for (i = 0; i < col; i++)
        if (board[row][i])
            return false;

    for (i = row, j = col; i >= 0 && j >= 0; i--, j--)
        if (board[i][j])
            return false;

    for (i = row, j = col; j >= 0 && i < a; i++, j--)
        if (board[i][j])
            return false;
}

```

```

return true;}

bool solveNQUtil(int board[a][a], int col)
{
    int i;
    if (col >= a)
        return true;
    for (i = 0; i < a; i++)
        if (isSafe(board, i, col))
        {
            board[i][col] = 1;
            if (solveNQUtil(board, col + 1))
                return true;
            board[i][col] = 0;
        }
    return false;}

bool solveNQ()
{
    int board[a][a], i, j;
    for(i=0; i<a; i++)
        for(j=0; j<a; j++)
            board[i][j]=0;

    if (solveNQUtil(board, 0) == false)
    {
        printf("Not possible");
        return false;}

    else
    {
        for ( i = 0; i < a; i++)
            for ( j = 0; j < a; j++)
                printf("%d ",board[j][i]);
        printf("\n");
    }
    return true;}

int main()
{
    scanf("%d",&a);
    solveNQ();
    return 0;}

```

