**Experiment No 3.1**

**Student Name: Ayushi Aggarwal UID: 21MCA2806**

**Branch: MCA Section/Group: 8B**

**Semester: 2nd Date of Performance: 06-04-2022**

**Subject Name: Computing Aptitude Subject Code: 21CAP-654**

1. **Aim/Overview of the practical:**

**Question 1: Given a chess board having N\*N cells, you need to place N queens on the board in such a way that no queen attacks any other queen.**

**Input:**

**The only line of input consists of a single integer denoting N.**

**Output:**

**If it is possible to place all the N queens in such a way that no queen attacks another queen, then print N lines having N integers. The integer in ith line and jth column will denote the cell (i, j) of the board and should be 1 if a queen is placed at otherwise 0. If there are more than way of placing queens print any of them. If it is not possible to place all N queens in the desired way, then print "Not possible" (without quotes).**

**Constraints:**

**1<=N<=10.**

**Code:**

#include<stdio.h>

int ar[11][11];

int n;

int issafe(int row,int col)

{

// coloumn check

int count=0;

for(int k=0;k<row;k++)

if(ar[k][col]==1)

return 0;

//upper right diagnoal

int k=row; int l=col;

while(k>=0&&l<n){

if(ar[k][l]==1)

return 0;

k--; l++;

}

//upper left diagnoal

k=row; l=col;

while(k>=0&&l>=0)

{

if(ar[k][l]==1)

return 0;

k--; l--;

}

return 1;

}

int solve(int row)

{

if(row>=n)

return 1;

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

{

if(issafe(row,i))

{ar[row][i]=1;

if(solve(row+1))

return 1;

ar[row][i]=0;

}

}

return 0;

}

void print1()

{

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

{

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

{

printf("%d ",ar[i][j]);

}

printf("\n");

}

}

int main()

{

scanf("%d",&n);

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

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

ar[i][j]=0;

if(solve(0))

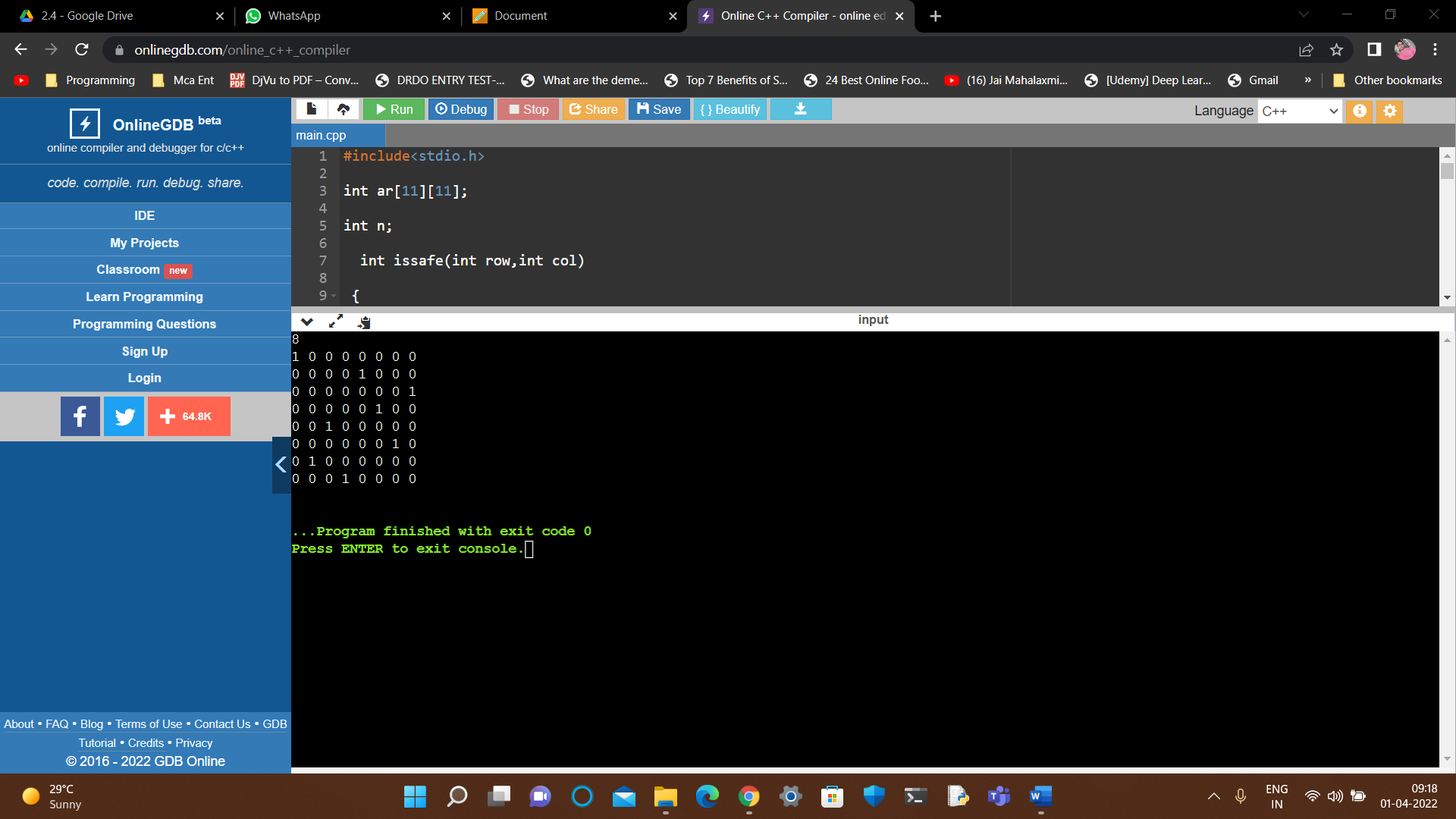
print1();

else

printf("Not possible");

}

**Output: -**

****

**Question 2:** You live in a village. The village can be represented as a line that contains n grids. Each grid can be denoted as a house that is marked as H or a blank space that is marked as…

A person lives in each house. A person can move to a grid if it is adjacent to that person. Therefore, the grid must be present on the left and right side of that person.

Now, you are required to put some fences that can be marked as B on some blank spaces so that the village can be divided into several pieces. A person cannot walk past a fence but can walk through a house.

You are required to divide the house based on the following rules:

A person cannot reach a house that does not belong to that specific person.

The number of grids each person can reach is the same and it includes the grid in which the house is situated.

To show that you are enthusiastic and if there are many answers, then you are required to print the one where most fences are placed.

Your task is to decide whether there is a possible solution. Print the possible solution.

**Input format**

First line: An integer n that represents the number of grids (1<=n<=20)

Second line: n characters that indicate the villages that are represented as H or.

**Output format**

The output must be printed in the following format:

First line: If possible, then print YES. Otherwise, print NO.

Second line: If the answer is YES, then print the way to do so.

**Code:**

#include<stdio.h>

int main()

{ int i, n;

char ch[20];

scanf("%d",&n);

scanf("%s",&ch);

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

if(ch[i]=='H' && ch[i+1]=='H') {

printf("NO");

return 0;

}

else if(ch[i]=='.')

ch[i]='B';

}

printf("YES\n");

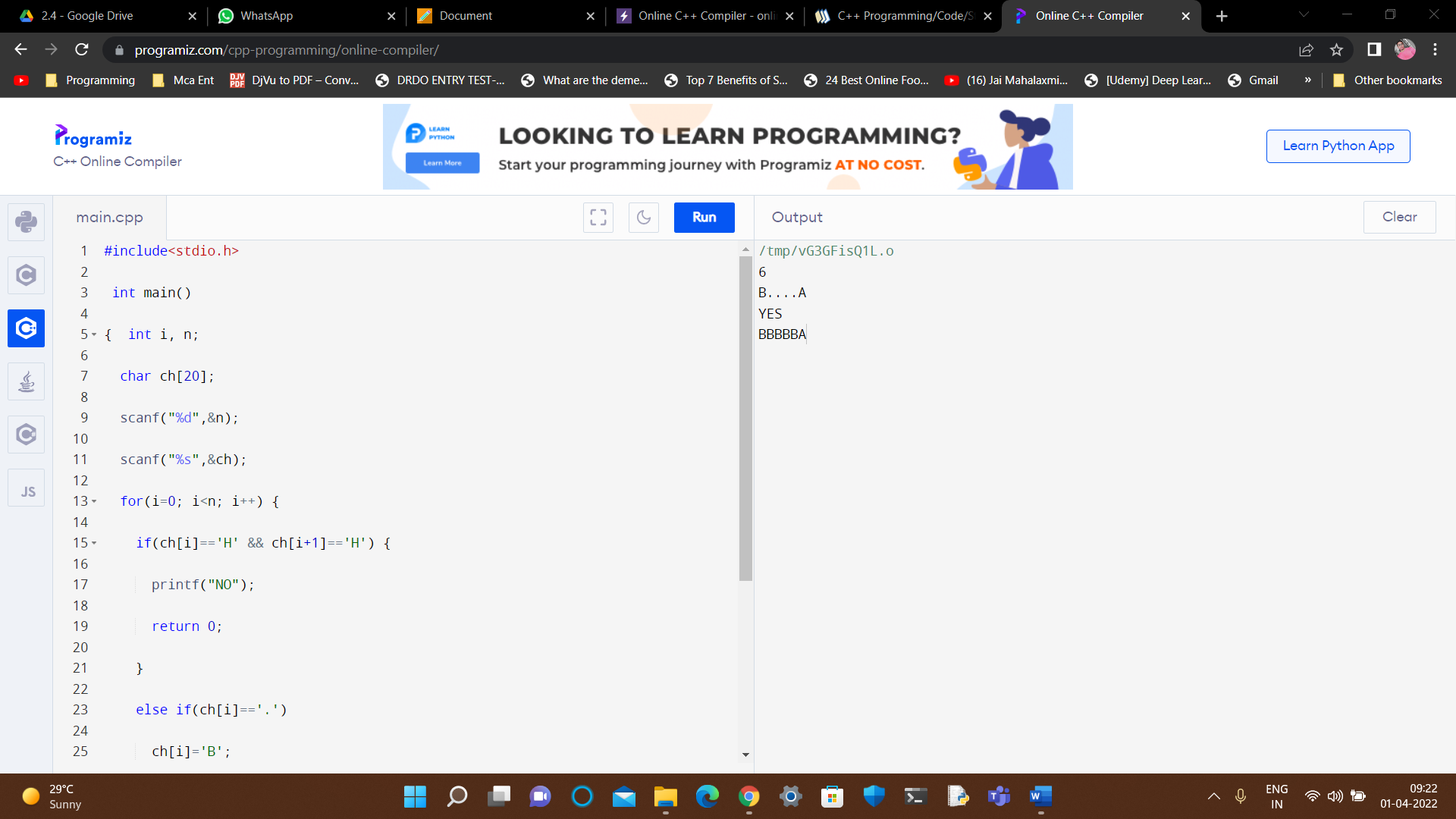
for(i=0; i<n; i++)

printf("%c",ch[i]);

return 0;

}

**Output: -**

****

**Learning outcomes (What I have learnt):**

#### I learned about Learn to solve backtracking problem and the N queens’ problem.

**Evaluation Grid:**

|  |  |  |  |
| --- | --- | --- | --- |
| Sr. No. | Parameters | Marks Obtained | Maximum Marks |
| 1. | Demonstration and Performance |  | 5 |
| 2. | Worksheet |  | 10 |
| 3. | Post Lab Quiz |  | 5 |