Started on	Thursday, 17 April 2025, 3:48 PM
State	Finished
Completed on	Tuesday, 29 April 2025, 11:50 AM
Time taken	11 days 20 hours
Overdue	11 days 18 hours
Grade	**80.00** out of 100.00

Question **1**

Correct

Mark 20.00 out of 20.00

Flag question

Write a python program to print the following pattern based on the given input.

Input:6
Output:

1
2 2
3 3 3
4 4 4 4
5 5 5 5 5

For example:

Input	Result
6	1 2 2 3 3 3 4 4 4 4 5 5 5 5 5
5	1 2 2 3 3 3 4 4 4 4

Answer: (penalty regime: 0 %)

```
n = int(input())
for i in range(1, n):
    print((str(i) + ' ') * i)
```

	Input	Expected	Got	
	6	1 2 2 3 3 3 4 4 4 4 5 5 5 5 5	1 2 2 3 3 3 4 4 4 4 5 5 5 5 5	
	5	1 2 2 3 3 3 4 4 4 4	1 2 2 3 3 3 4 4 4 4	
	8	1 2 2 3 3 3 4 4 4 4 5 5 5 5 5 6 6 6 6 6 6 7 7 7 7 7 7 7	1 2 2 3 3 3 4 4 4 4 5 5 5 5 5 6 6 6 6 6 6 7 7 7 7 7 7 7	

Passed all tests!

Correct

Marks for this submission: 20.00/20.00.

Question **2**

Correct

Mark 20.00 out of 20.00

Flag question

Rat In A Maze Problem

You are given a maze in the form of a matrix of size n * n. Each cell is either clear or blocked denoted by 1 and 0 respectively. A rat sits at the top-left cell and there exists a block of cheese at the bottom-right cell. Both these cells are guaranteed to be clear. You need to find if the rat can get the cheese if it can move only in one of the two directions - down and right. It can't move to blocked cells.

Provide the solution for the above problem(Consider n=4)

The output (Solution matrix) must be 4*4 matrix with value "1" which indicates the path to destination and "0" for the cell indicating the absence of the path to destination.

Answer: (penalty regime: 0 %)

Reset answer

```
    if x == N-1 and y == N-1:
        sol[x][y] = 1
        return True
    if isSafe(maze,x,y) == True:
        sol[x][y] = 1
        if solveMazeUtil(maze,x+1,y,sol) == True:
            return True
        if solveMazeUtil(maze,x,y+1,sol) == True:
            return True
        s[x][y] = 0
        return False


if __name__ == "__main__":
    # Initialising the maze
    maze = [ [1, 0, 0, 0],
             [1, 1, 0, 1],
             [0, 1, 0, 0],
             [1, 1, 1, 1] ]

    solveMaze(maze)

```

	Expected	Got	
	1 0 0 0 1 1 0 0 0 1 0 0 0 1 1 1	1 0 0 0 1 1 0 0 0 1 0 0 0 1 1 1	

Passed all tests!

Correct

Marks for this submission: 20.00/20.00.

Question **3**

Correct

Mark 20.00 out of 20.00

Flag question

You are given an integer **N**. For a given **N** x **N** chessboard, find a way to place **'N'** queens such that no queen can attack any other queen on the chessboard.

A queen can be attacked when it lies in the same row, column, or the same diagonal as any of the other queens. **You have to print one such configuration**.

Note :

Get the input from the user for N . The value of N must be from 1 to 6

If solution exists Print a binary matrix as output that has 1s for the cells where queens are placed

If there is no solution to the problem print "Solution does not exist"

For example:

Input	Result
6	0 0 0 1 0 0 1 0 0 0 0 0 0 0 0 0 1 0 0 1 0 0 0 0 0 0 0 0 0 1 0 0 1 0 0 0

Answer: (penalty regime: 0 %)

Reset answer

```
global N
N = int(input())

def printSolution(board):
    for i in range(N):
        for j in range(N):
            print(board[i][j], end = " ")
        print()

def isSafe(board, row, col):

    # Check this row on left side
    for i in range(col):
        if board[row][i] == 1:
            return False

    # Check upper diagonal on left side
    for i, j in zip(range(row, -1, -1),
                    range(col, -1, -1)):
        if board[i][j] == 1:
            return False

```

	Input	Expected	Got	
	2	Solution does not exist	Solution does not exist	
	3	Solution does not exist	Solution does not exist	
	6	0 0 0 1 0 0 1 0 0 0 0 0 0 0 0 0 1 0 0 1 0 0 0 0 0 0 0 0 0 1 0 0 1 0 0 0	0 0 0 1 0 0 1 0 0 0 0 0 0 0 0 0 1 0 0 1 0 0 0 0 0 0 0 0 0 1 0 0 1 0 0 0	

Passed all tests!

Correct

Marks for this submission: 20.00/20.00.

Question **4**

Correct

Mark 20.00 out of 20.00

Flag question

SUBSET SUM PROBLEM

Given a set of positive integers, and a value sum, determine that the sum of the subset of a given set is equal to the given sum.

Write the program for subset sum problem.

INPUT

1.no of elements

2.Input the given elements

3.Get the target sum

OUTPUT

True , if subset with required sum is found

False , if subset with required sum is not found

For example:

Input	Result
5 4 16 5 23 12 9	4 16 5 23 12 True,subset found

Answer: (penalty regime: 0 %)

Reset answer

```
    if sum>target:
        return False
    if sum == target:
        return True
    return SubsetSum(a,i+1,sum,target,n) or SubsetSum(a,i+1,sum+a[i],target,n)
a=[]
size=int(input())
```

```
for i in range(size):
    x=int(input())
    a.append(x)

target=int(input())
n=len(a)
if(SubsetSum(a,0,0,target,n)==True):
    for i in range(size):
        print(a[i])
    print("True,subset found")
else:
    for i in range(size):
        print(a[i])
    print("False,subset not found")

```

	Input	Expected	Got	
	5 4 16 5 23 12 9	4 16 5 23 12 True,subset found	4 16 5 23 12 True,subset found	
	4 1 2 3 4 11	1 2 3 4 False,subset not found	1 2 3 4 False,subset not found	
	7 10 7 5 18 12 20 15 35	10 7 5 18 12 20 15 True,subset found	10 7 5 18 12 20 15 True,subset found	