

<b>Started on</b>	Friday, 31 January 2025, 1:22 PM
<b>State</b>	Finished
<b>Completed on</b>	Friday, 31 January 2025, 1:35 PM
<b>Time taken</b>	13 mins 9 secs
<b>Grade</b>	<b>80.00</b> out of 100.00

## Question 1

Correct

Mark 20.00 out of 20.00

Given a 2D matrix **tsp[][]**, where each row has the array of distances from that indexed city to all the other cities and **-1** denotes that there doesn't exist a path between those two indexed cities. The task is to print minimum cost in TSP cycle.

```
tsp[][] = {{-1, 30, 25, 10},
{15, -1, 20, 40},
{10, 20, -1, 25},
{30, 10, 20, -1}};
```

**Answer:** (penalty regime: 0 %)

Reset answer

```
1 from typing import defaultdict
2 INT_MAX = 2147483647
3 def findMinRoute(tsp):
4     sum = 0
5     counter = 0
6     j = 0
7     i = 0
8     min = INT_MAX
9     visitedRouteList = defaultdict(int)
10    visitedRouteList[0] = 1
11    route = [0] * len(tsp)
12    while i < len(tsp) and j < len(tsp[i]):
13        #Write your code here
14        #Start here
15        if counter >= len(tsp[i]) - 1:
16            break
17        if j != i and (visitedRouteList[j] == 0):
18            if tsp[i][j] < min:
19                min = tsp[i][j]
20                route[counter] = j + 1
21            j += 1
22        if j == len(tsp[i]):
```

	Expected	Got	
✓	Minimum Cost is : 50	Minimum Cost is : 50	✓

Passed all tests! ✓

Correct

Marks for this submission: 20.00/20.00.

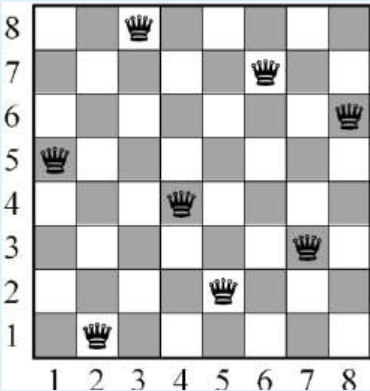
Question 2

Not answered

Mark 0.00 out of 20.00

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 8

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
5	1 0 0 0 0 0 0 0 1 0 0 1 0 0 0 0 0 0 0 1 0 0 1 0 0

Answer: (penalty regime: 0 %)

1

Question 3

Correct

Mark 20.00 out of 20.00

Create a python program using dynamic programming for 0/1 knapsack problem.

For example:

Test	Input	Result
knapSack(W, wt, val, n)	3 3 50 60 100 120 10 20 30	The maximum value that can be put in a knapsack of capacity W is: 220

Answer: (penalty regime: 0 %)

Reset answer

```

1 def knapSack(W, wt, val, n):
2     if W==0 or n==0:
3         return 0
4     if wt[n-1]>W:
5         return knapSack(W,wt,val,n-1)
6     else:
7         inc=val[n-1]+knapSack(W-wt[n-1],wt,val,n-1)
8         exc=knapSack(W,wt,val,n-1)
9         return max(inc,exc)
10
11
12 x=int(input())
13 y=int(input())
14 W=int(input())
15 val=[]
16 wt=[]
17 for i in range(x):
18     val.append(int(input()))
19 for y in range(y):
20     wt.append(int(input()))
21 n = len(val)
22 print('The maximum value that can be put in a knapsack of capacity W is: ',knapSack(W, wt, val, n))

```

	Test	Input	Expected	Got	
✓	knapSack(W, wt, val, n)	3 3 50 60 100 120 10 20 30	The maximum value that can be put in a knapsack of capacity W is: 220	The maximum value that can be put in a knapsack of capacity W is: 220	✓

	Test	Input	Expected	Got	
✓	knapSack(W, wt, val, n)	3 3 40 50 90 110 10 20 30	The maximum value that can be put in a knapsack of capacity W is: 160	The maximum value that can be put in a knapsack of capacity W is: 160	✓

Passed all tests! ✓

Correct

Marks for this submission: 20.00/20.00.

## Question 4

Correct

Mark 20.00 out of 20.00

Create a python program for the following problem statement.

You are given an  $n \times n$  grid representing a field of cherries, each cell is one of three possible integers.

- 0 means the cell is empty, so you can pass through,
- 1 means the cell contains a cherry that you can pick up and pass through, or
- -1 means the cell contains a thorn that blocks your way.

Return the maximum number of cherries you can collect by following the rules below.

- Starting at the position (0, 0) and reaching ( $n - 1$ ,  $n - 1$ ) by moving right or down through valid path cells (cells with value 0 or 1).
- After reaching ( $n - 1$ ,  $n - 1$ ), returning to (0, 0) by moving left or up through valid path cells.
- When passing through a path cell containing a cherry, you pick it up, and the cell becomes an empty cell 0.
- If there is no valid path between (0, 0) and ( $n - 1$ ,  $n - 1$ ), then no cherries can be collected.

For example:

Test	Result
obj.cherryPickup(grid)	5

Answer: (penalty regime: 0 %)

Reset answer

```

1 class Solution:
2     def cherryPickup(self, grid):
3         n = len(grid)
4         rows=len(grid)
5         cols=len(grid[0])
6         memo={}
7         def dp(r,c1,c2):
8             if r==rows or c1<0 or c1==cols or c2<0 or c2==cols:
9                 return 0
10            if (r,c1,c2) in memo:
11                return memo[(r,c1,c2)]
12            cherries=grid[r][c1]+(grid[r][c2] if c1!=c2 else 0)
13            maxcherries=0
14            for dc1 in [-1,0,1]:
15                for dc2 in [-1,0,1]:
16                    maxcherries=max(maxcherries,dp(r+1,c1+dc1,c2+dc2))
17            result=cherries+maxcherries
18            memo[(r,c1,c2)]=result
19            return result
20
21            #####      Add your code here      #####
22

```

	Test	Expected	Got	
✓	obj.cherryPickup(grid)	5	5	✓

Passed all tests! ✓



Marks for this submission: 20.00/20.00.

## Question 5

Correct

Mark 20.00 out of 20.00

Create a python program using brute force method of searching for the given substring in the main string.

For example:

Test	Input	Result
match(str1,str2)	AABAACAADAABAABA AABA	Found at index 0 Found at index 9 Found at index 12

Answer: (penalty regime: 0 %)

Reset answer

```

1 def match(string,sub):
2     l = len(string)
3     ls = len(sub)
4     start = sub[0]
5
6     ##### Add your code here #####
7     for i in range(l-ls+1):
8         j=0
9         while j<ls and string[i+j]==sub[j]:
10            j+=1
11        if j==ls:
12            print('Found at index',i)
13        return -1
14
15 str1=input()
16 str2=input()

```

	Test	Input	Expected	Got	
✓	match(str1,str2)	AABAACAADAABAABA AABA	Found at index 0 Found at index 9 Found at index 12	Found at index 0 Found at index 9 Found at index 12	✓
✓	match(str1,str2)	saveetha savee	Found at index 0	Found at index 0	✓

Passed all tests! ✓

Correct

Marks for this submission: 20.00/20.00.