

Started on	Tuesday, 22 October 2024, 1:34 PM
State	Finished
Completed on	Tuesday, 22 October 2024, 2:31 PM
Time taken	57 mins 24 secs
Grade	80.00 out of 100.00

Question **1**

Correct

Mark 20.00 out of 20.00

Write a python program to implement merge sort using iterative approach on the given list of float values.

For example:

Test	Input	Result
Merge_Sort(S)	5 10.2 21.3 3.5 7.8 9.8	The Original array is: [10.2, 21.3, 3.5, 7.8, 9.8] Array after sorting is: [3.5, 7.8, 9.8, 10.2, 21.3]
Merge_Sort(S)	6 20.3 41.2 5.3 6.2 8.1 65.2	The Original array is: [20.3, 41.2, 5.3, 6.2, 8.1, 65.2] Array after sorting is: [5.3, 6.2, 8.1, 20.3, 41.2, 65.2]

Answer: (penalty regime: 0 %)

```
1 def Merge_Sort(S):
2     n = len(S)
3     size = 1
4
5     while size < n:
6         left = 0
7         while left < n - 1:
8             mid = min(left + size - 1, n - 1)
9             right = min(left + 2 * size - 1, n - 1)
10
11             merge(S, left, mid, right)
12             left += 2 * size
13
14         size *= 2
15
16 def merge(S, left, mid, right):
17     n1 = mid - left + 1
18     n2 = right - mid
19
20     L = [S[left + i] for i in range(n1)]
21     R = [S[mid + i + 1] for i in range(n2)]
22
```

	Test	Input	Expected	Got
✓	Merge_Sort(S)	5 10.2 21.3 3.5 7.8 9.8	The Original array is: [10.2, 21.3, 3.5, 7.8, 9.8] Array after sorting is: [3.5, 7.8, 9.8, 10.2, 21.3]	The Original array is: [10.2, 21.3, 3.5, 7.8, 9.8] Array after sorting is: [3.5, 7.8, 9.8, 10.2, 21.3]
✓	Merge_Sort(S)	6 20.3 41.2 5.3 6.2 8.1 65.2	The Original array is: [20.3, 41.2, 5.3, 6.2, 8.1, 65.2] Array after sorting is: [5.3, 6.2, 8.1, 20.3, 41.2, 65.2]	The Original array is: [20.3, 41.2, 5.3, 6.2, 8.1, 65.2] Array after sorting is: [5.3, 6.2, 8.1, 20.3, 41.2, 65.2]
✓	Merge_Sort(S)	4 2.3 6.1 4.5 96.5	The Original array is: [2.3, 6.1, 4.5, 96.5] Array after sorting is: [2.3, 4.5, 6.1, 96.5]	The Original array is: [2.3, 6.1, 4.5, 96.5] Array after sorting is: [2.3, 4.5, 6.1, 96.5]

Passed all tests! ✓

Correct

Marks for this submission: 20.00/20.00.

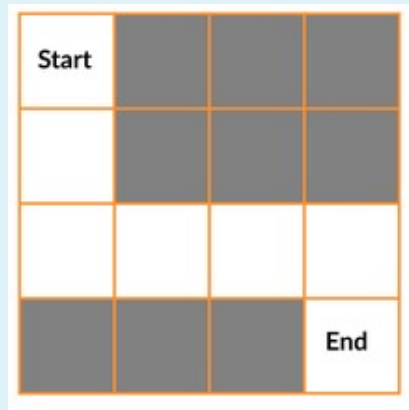
Question **2**

Correct

Mark 20.00 out of 20.00

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 rat sits at the top-left cell and there exists a block of cheese at the bottom-right cell. Both these cells are clear. You need to find if the rat can get the cheese if it can move only in one of the two directions - down or right. It cannot 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 indicating the absence of the path to destination.

Answer: (penalty regime: 0 %)

Reset answer

```
1 N = 4
2 def printSolution( sol ):
3     for i in sol:
4         for j in i:
5             print(str(j) + " ", end = "")
6         print("")
7
8 def isSafe( maze, x, y ):
9     if x >= 0 and x < N and y >= 0 and y < N and maze[x][y] == 1:
10         return True
11     return False
12
13 def solveMaze( maze ):
14
15     # Creating a 4 * 4 2-D list
```

	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.

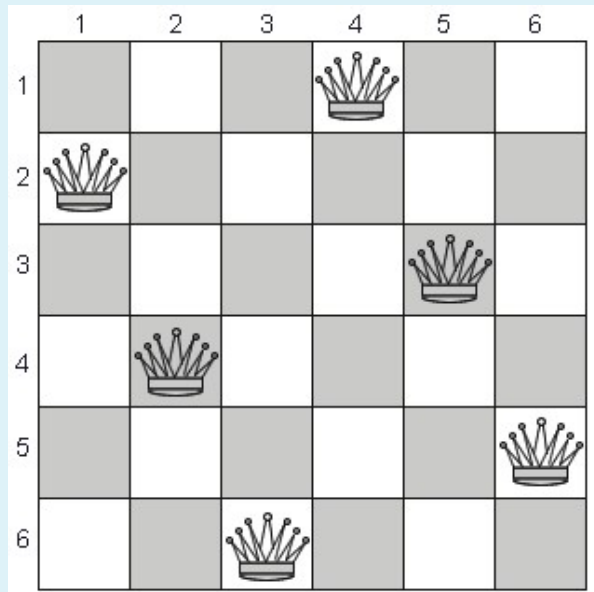
Question 3

Correct

Mark 20.00 out of 20.00

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	<pre> 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 </pre>

Answer: (penalty regime: 0 %)

Reset answer

```

1 global N
2 N = int(input())
3

```

	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

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 value 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
4	16
16	5
5	23
23	12
12	True,subset found
9	

Answer: (penalty regime: 0 %)

Reset answer

```
1
2 def SubsetSum(a,i,sum,target,n):
3     # Write your code here
4     if i == n:
5         return sum == target
6     if SubsetSum(a,i+1,sum+a[i],target,n):
7         return True
8     if SubsetSum(a,i+1,sum,target,n):
9         return True
10    return False
11
12 a=[]
13 size=int(input())
14 for i in range(size):
15     x=int(input())
16     a.append(x)
```


	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	✓

Passed all tests! ✓

Correct

Marks for this submission: 20.00/20.00.

Question 5

Incorrect

Mark 0.00 out of 20.00

Greedy coloring doesn't always use the minimum number of colors possible to color a graph. For a graph greedy coloring will use at most $x+1$ color. Greedy coloring can be arbitrarily bad;

Create a python program to implement graph colouring using Greedy algorithm.

For example:

Test	Result
colorGraph(graph, n)	Color assigned to vertex 0 is BLUE Color assigned to vertex 1 is GREEN Color assigned to vertex 2 is BLUE Color assigned to vertex 3 is RED Color assigned to vertex 4 is RED Color assigned to vertex 5 is GREEN

Answer: (penalty regime: 0 %)

Reset answer

```
1 class Graph:
2     def __init__(self, edges, n):
3         self.adjList = [[] for _ in range(n)]
4
5         # add edges to the undirected graph
6         for (src, dest) in edges:
7             self.adjList[src].append(dest)
8             self.adjList[dest].append(src)
9     def colorGraph(graph, n):
10        ##### Add your code here #####
11    if __name__ == '__main__':
12        colors = ['', 'BLUE', 'GREEN', 'RED', 'YELLOW', 'ORANGE', 'PINK',
13                'BLACK', 'BROWN', 'WHITE', 'PURPLE', 'VOILET']
14        edges = [(0, 1), (0, 4), (0, 5), (4, 5), (1, 4), (1, 3), (2, 3), (2, 4)]
15        n = 6
16        graph = Graph(edges, n)
17        colorGraph(graph, n)
18
```

Syntax Error(s)

Sorry: IndentationError: expected an indented block (__tester__.python3, line 11)

Incorrect

Marks for this submission: 0.00/20.00.