TT DS PYTHON MODULE-24

Started on	Tuesday, 20 May 2025, 3:28 PM
State	Finished
Completed on	Tuesday, 20 May 2025, 3:51 PM
Time taken	23 mins 3 secs
Grade	**80.00** out of 100.00

Question **1**

Correct

Mark 20.00 out of 20.00

Flag question

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

```
def knapSack(W, wt, val, n):
    dp=[[-1]*(W+1) for _ in range(len(val)+1)]
    for i in range(W+1):
        dp[0][i]=0
    for i in range(n+1):
        dp[i][0]=0
    for i in range(1,n+1):
        for j in range(1,W+1):
            if j<wt[i-1]:
                dp[i][j]=dp[i-1][j]
            else:
                dp[i][j]=max(dp[i-1][j],dp[i-1][j-wt[i-1]]+val[i-1])
    return dp[n][W]

x=int(input())
y=int(input())
W=int(input())
val=[]
wt=[]
for i in range(x):
    val.append(int(input()))
for y in range(y):
```

	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
	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

Passed all tests!

Correct

Marks for this submission: 20.00/20.00.

Question **2**

Not answered

Mark 0.00 out of 20.00

LONGEST PALINDROMIC SUBSEQUENCE

Given a sequence, find the length of the longest palindromic subsequence in it.

Flag question

For example:

Input	Result
ABBDCACB	The length of the LPS is 5

Answer: (penalty regime: 0 %)

```
1
```

Question **3**

Correct

Mark 20.00 out of 20.00

Flag question

Create a python program to for the following problem statement.

You are given an `n x 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

```python
class Solution:
    def cherryPickup(self, grid):
        n = len(grid)
        dp = [[[-1] * n for _ in range(n)] for _ in range(n)]

        def f(x1, y1, x2):
            y2 = x1 + y1 - x2
            if x1 < 0 or y1 < 0 or x2 < 0 or y2 < 0 or grid[x1][y1] == -1 or grid[x2][y2] == -1:
                return float('-inf')
            if x1 == 0 and y1 == 0 and x2 == 0 and y2 == 0:
                return grid[0][0]
            if dp[x1][y1][x2] != -1:
                return dp[x1][y1][x2]
            cherries = grid[x1][y1]
            if x1 != x2 or y1 != y2:
                cherries += grid[x2][y2]
            cherries += max(
                f(x1 - 1, y1, x2 - 1),
                f(x1, y1 - 1, x2 - 1),
                f(x1 - 1, y1, x2),
                f(x1, y1 - 1, x2))

```

	Test	Expected	Got	
	obj.cherryPickup(grid)	5	5	

Passed all tests!

Correct

Marks for this submission: 20.00/20.00.

Question **4**

Correct

Mark 20.00 out of 20.00

Flag question

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

```
from typing import DefaultDict


INT_MAX = 2147483647


def findMinRoute(tsp):
    sum = 0
    counter = 0
    j = 0
    i = 0
    min = INT_MAX
    visitedRouteList = DefaultDict(int)


    visitedRouteList[0] = 1
    route = [0] * len(tsp)


    while i < len(tsp) and j < len(tsp[i]):
        if counter >= len(tsp[i]) - 1:
            break
```

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

Passed all tests!

Correct

Marks for this submission: 20.00/20.00.

Question **5**

Correct

Mark 20.00 out of 20.00

Flag question

Create a python program to find the maximum value in linear search.

For example:

Test	Input	Result
find_maximum(test_scores)	10 88 93 75 100 80 67 71 92 90 83	Maximum value is 100

Answer: (penalty regime: 0 %)

Reset answer

```
def find_maximum(lst):
    if len(lst)==0:
```

```
    if len(lst)==0:
        return 0
    max_=lst[0]
    for i in lst:
        if i>max_:
            max_=i
    return max_

test_scores = []
n=int(input())
for i in range(n):
    test_scores.append(int(input()))
print("Maximum value is ",find_maximum(test_scores))
```

	Test	Input	Expected	Got	
	find_maximum(test_scores)	10 88 93 75 100 80 67 71 92 90 83	Maximum value is 100	Maximum value is 100	
	find_maximum(test_scores)	5 45 86 95 76 28	Maximum value is 95	Maximum value is 95	

Passed all tests!

Correct

Marks for this submission: 20.00/20.00.

Finish r