Started on	Saturday, 3 May 2025, 3:05 PM
State	Finished
Completed on	Saturday, 3 May 2025, 3:52 PM
Time taken	46 mins 30 secs
Grade	**80.00** out of 100.00

Question **1**

Correct

Mark 20.00 out of 20.00

Create a Dynamic Programming python Implementation of Coin Change Problem.

For example:

Test	Input	Result
count(arr, m, n)	3 4 1 2 3	4

Answer: (penalty regime: 0 %)

Reset answer

Ace editor not ready. Perhaps reload page?
Falling back to raw text area.

```
def count(S, m, n):
    table = [[0 for x in range(m)] for x in range(n+1)]
    for i in range(m):
        table[0][i] = 1
    for i in range(1, n+1):
        for j in range(m):
            # Count of solutions including S[j]
            #Start here
            x = table[i - S[j]][j] if i-S[j] >= 0 else 0
            # Count of solutions excluding S[j]
            y = table[i][j-1] if j >= 1 else 0
            # total count
            table[i][j] = x + y
    return table[n][m-1]
    #End here
arr = []
m = int(input())
n = int(input())
```

	Test	Input	Expected	Got	
✔	count(arr, m, n)	3 4 1 2 3	4	4	✔
✔	count(arr, m, n)	3 16 1 2 5	20	20	✔

Passed all tests! ✔

Correct

Marks for this submission: 20.00/20.00.

Question **2**

Correct

Mark 20.00 out of 20.00

Create a python program to find Minimum number of jumps to reach end of the array using naive method(recursion)

For example:

Test	Input	Result
minJumps(arr, 0, n-1)	10 1 3 6 3 2 3 6 8 9 5	Minimum number of jumps to reach end is 4

Answer: (penalty regime: 0 %)

Reset answer

Ace editor not ready. Perhaps reload page?
Falling back to raw text area.

```
def minJumps(arr, l, h):
    ###########   Add your code here ###########
    #Start here
    if (h == l):
        return 0
    if (arr[l] == 0):
        return float('inf')
    min = float('inf')
    for i in range(l + 1, h + 1):
        if (i < l + arr[l] + 1):
            jumps = minJumps(arr, i, h)
            if (jumps != float('inf') and
                        jumps + 1 < min):
                min = jumps + 1

    return min
    #End here
arr = []
```

	Test	Input	Expected	Got	
✔	minJumps(arr, 0, n-1)	10 1 3 6 3 2 3 6 8 9 5	Minimum number of jumps to reach end is 4	Minimum number of jumps to reach end is 4	✔
✔	minJumps(arr, 0, n-1)	7 3 2 5 9 4 1 6	Minimum number of jumps to reach end is 2	Minimum number of jumps to reach end is 2	✔

Passed all tests! ✔

Correct

Marks for this submission: 20.00/20.00.

Question **3**

Correct

Mark 20.00 out of 20.00

Write a python program to print the following pattern

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

For example:

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

Answer: (penalty regime: 0 %)

Ace editor not ready. Perhaps reload page?
Falling back to raw text area.

```
n=int(input())
for i in range(n):
    for j in range(n, i, -1):
        print(j, end=' ')
    print()
```

	Input	Expected	Got	
✔	5	5 4 3 2 1 5 4 3 2 5 4 3 5 4 5	5 4 3 2 1 5 4 3 2 5 4 3 5 4 5	✔
✔	6	6 5 4 3 2 1 6 5 4 3 2 6 5 4 3 6 5 4 6 5 6	6 5 4 3 2 1 6 5 4 3 2 6 5 4 3 6 5 4 6 5 6	✔
✔	4	4 3 2 1 4 3 2 4 3 4	4 3 2 1 4 3 2 4 3 4	✔

Passed all tests! ✔

Correct

Marks for this submission: 20.00/20.00.

Question **4**

Incorrect

Mark 0.00 out of 20.00

Given an integer array `nums`, find the contiguous subarray (containing at least one number) which has the largest sum and return *its sum*.

A **subarray** is a **contiguous** part of an array.

Example 1:

```
Input: nums = [-2,1,-3,4,-1,2,1,-5,4]
Output: 6
Explanation: [4,-1,2,1] has the largest sum = 6.
```

For example:

Test	Input	Result
s.maxSubArray(A)	9 -2 1 -3 4 -1 2 1 -5 4	The sum of contiguous sublist with the largest sum is 6

Answer: (penalty regime: 0 %)

Reset answer

Ace editor not ready. Perhaps reload page?

Falling back to raw text area.

```
class Solution:
    def maxSubArray(self,A):
        ############ Add your Code here


A =[]
n=int(input())
for i in range(n):
    A.append(int(input()))
s=Solution()
print("The sum of contiguous sublist with the largest sum is",s.maxSubArray(A))
```

Syntax Error(s)

```
Sorry: IndentationError: expected an indented block (__tester__.python3, line 6)
```

Incorrect

Marks for this submission: 0.00/20.00.

Question **5**

Correct

Mark 20.00 out of 20.00

Write a Python program to Implement Minimum cost path in a Directed Graph

For example:

Test	Result
`getMinPathSum(graph, visited, necessary, source, dest, 0);`	12

Answer: (penalty regime: 0 %)

Reset answer

Ace editor not ready. Perhaps reload page?
Falling back to raw text area.

```
minSum = 1000000000
def getMinPathSum(graph, visited, necessary,
                  src, dest, currSum):
    #Start here
    global minSum
    if (src == dest):
        flag = True;
        for i in necessary:
            if (not visited[i]):
                flag = False;
                break;
        if (flag):
            minSum = min(minSum, currSum);
        return;

    else:
        visited[src] = True;
        for node in graph[src]:
```

	Test	Expected	Got	
✔	`getMinPathSum(graph, visited, necessary, source, dest, 0);`	12	12	✔

Passed all tests! ✔

Correct

Marks for this submission: 20.00/20.00.