Started on	Thursday, 1 May 2025, 11:13 AM
State	Finished
Completed on	Thursday, 1 May 2025, 11:52 AM
Time taken	39 mins 12 secs
Grade	**100.00** out of 100.00

Question **1**

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) using float values

For example:

Test	Input	Result
minJumps(arr, 0, n-1)	6 2.3 7.4 6.3 1.5 8.2 0.1	Minimum number of jumps to reach end is 2

Answer: (penalty regime: 0 %)

Reset answer

```
def minJumps(arr, l, h):
    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
arr = []
n = int(input())
for i in range(n):
    arr.append(float(input()))
print('Minimum number of jumps to reach','end is', minJumps(arr, 0, n-1))
```

	Test	Input	Expected	Got	
✔	minJumps(arr, 0, n-1)	6 2.3 7.4 6.3 1.5 8.2 0.1	Minimum number of jumps to reach end is 2	Minimum number of jumps to reach end is 2	✔

	Test	Input	Expected	Got	

✔	`minJumps(arr, 0, n-1)`	`10`
		`3.2`
		`3.2`
		`5`
		`6.2`
		`4.9`
		`1.2`
		`5.0`
		`7.3`
		`4.6`
		`6.2`
			`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 **2**

Correct

Mark 20.00 out of 20.00

Write a python program to Implement Minimum cost path using Dynamic Programming.

For example:

Input	Result
3 3	8

Answer: (penalty regime: 0 %)

```
R = int(input())
C = int(input())
def minCost(cost, m, n):

    tc = [[0 for x in range(C)] for x in range(R)]
    tc[0][0] = cost[0][0]
    for i in range(1, m+1):
        tc[i][0] = tc[i-1][0] + cost[i][0]
    for j in range(1, n+1):
        tc[0][j] = tc[0][j-1] + cost[0][j]
    for i in range(1, m+1):
        for j in range(1, n+1):
            tc[i][j] = min(tc[i-1][j-1], tc[i-1][j], tc[i][j-1]) + cost[i][j]
    return tc[m][n]

cost = [[1, 2, 3],
        [4, 8, 2],
        [1, 5, 3]]
print(minCost(cost,2,2))
```

	Input	Expected	Got	
✔	3 3	8	8	✔

Passed all tests! ✔

Correct

Marks for this submission: 20.00/20.00.

Question **3**

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

```
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):

            x = table[i - S[j]][j] if i-S[j] >= 0 else 0
            y = table[i][j-1] if j >= 1 else 0
            table[i][j] = x + y

    return table[n][m-1]


arr = []
m = int(input())
n = int(input())
for i in range(m):
    arr.append(int(input()))
print(count(arr, m, n))
```

	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 **4**

Correct

Mark 20.00 out of 20.00

Write a python program to calculate the length of the given string using recursion

For example:

Test	Input	Result
length(str)	saveetha	length of saveetha is 8
length(str)	engineering	length of engineering is 11

Answer: (penalty regime: 0 %)

```
def length(str):
    if str=="":
        return 0
    return 1+length(str[1:])
str = input()
leng = length(str)
print("length of",str,"is",leng)
```

	Test	Input	Expected	Got	
✔	length(str)	saveetha	length of saveetha is 8	length of saveetha is 8	✔
✔	length(str)	engineering	length of engineering is 11	length of engineering is 11	✔
✔	length(str)	Welcome	length of Welcome is 7	length of Welcome is 7	✔

Passed all tests! ✔

Correct

Marks for this submission: 20.00/20.00.

Question **5**

Incorrect

Mark 20.00 out of 20.00

Write a python program to find the maximum contiguous subarray.

For example:

Test	Input	Result
maxSubArraySum(a,n)	8 -2 -3 4 -1 -2 1 5 -3	Maximum contiguous sum is 7

Answer: (penalty regime: 0 %)

Reset answer

```
def maxSubArraySum(a,size):
    max_till_now = a[0]
    max_ending = 0

    for i in range(0, size):
        if max_ending < 1:
            max_ending = 0


        elif (max_till_now < max_ending):
            max_till_now = max_ending

    return max_till_now
n=int(input())
a =[]
for i in range(n):
    a.append(int(input()))

print("Maximum contiguous sum is", maxSubArraySum(a,n))
```

Syntax Error(s)

```
Sorry: IndentationError: unindent does not match any outer indentation level (__tester__.python3, line 10)
```

Incorrect

Marks for this submission: 0.00/20.00.