Started on	Saturday, 19 April 2025, 3:30 PM
State	Finished
Completed on	Saturday, 19 April 2025, 5:51 PM
Time taken	2 hours 21 mins
Overdue	21 mins 22 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 Using a recursive function to calculate the sum of a sequence
For example:

Input	Result
20	210
36	666
45	1035

Answer: (penalty regime: 0 %)

```
def seq(n):

    if n<=0:
        return 0

    else:
        return n+seq(n-1)

n = int(input())
print(seq(n))
```

	Input	Expected	Got	
✓	20	210	210	✓
✓	36	666	666	✓
✓	45	1035	1035	✓
✓	58	1711	1711	✓
✓	65	2145	2145	✓

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 for printing Minimum Cost Simple Path between two given nodes in a directed and weighted graph

For example:

Test	Result
minimumCostSimplePath(s, t, visited, graph)	-3

Answer: (penalty regime: 0 %)

Reset answer

```python
import sys
V = 5
INF = sys.maxsize
def minimumCostSimplePath(u, destination,
                          visited, graph):
    ########  Add your code here ##############
    if (u == destination):
        return 0
    visited[u] = 1
    ans = INF
    for i in range(V):
        if (graph[u][i] != INF and not visited[i]):
            curr = minimumCostSimplePath(i, destination,visited, graph)
            if (curr < INF):
                ans = min(ans, graph[u][i] + curr)
    visited[u] = 0
    return ans

if __name__=="__main__":
    graph = [[INF for j in range(V)]
                  for i in range(V)]
    visited = [0 for i in range(V)]
```

	Test	Expected	Got	
✓	minimumCostSimplePath(s, t, visited, graph)	-3	-3	✓

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 find the maximum contiguous subarray on the given float array using kadane's algorithm.

For example:

Test	Input	Result
`s.maxSubArray(A)`	5 -9.6 -3.5 6.3 8.31 9.2	The sum of contiguous sublist with the largest sum is 23.8

Answer: (penalty regime: 0 %)

Reset answer

```python
class Solution:
    def maxSubArray(a,size):
        #####################  Add your Code here #############
        max_sum = A[0]
        current_sum = A[0]
        for i in range(1, len(A)):
            current_sum = max(A[i], current_sum + A[i])
            max_sum = max(max_sum, current_sum)
        return max_sum

A =[]
n=int(input())
for i in range(n):
    A.append(float(input()))
s=Solution()
print("The sum of contiguous sublist with the largest sum is {:.1f}".format(s.maxSubArray(A)))
```

	Test	Input	Expected	Got	
✓	`s.maxSubArray(A)`	5 -9.6 -3.5 6.3 8.31 9.2	The sum of contiguous sublist with the largest sum is 23.8	The sum of contiguous sublist with the largest sum is 23.8	✓
✓	`s.maxSubArray(A)`	7 2.3 6.5 4.6 -7.8 -2.8 -1.6 9.8	The sum of contiguous sublist with the largest sum is 13.4	The sum of contiguous sublist with the largest sum is 13.4	✓

Passed all tests! ✓

Correct

Marks for this submission: 20.00/20.00.

Question **5**

Correct

Mark 20.00 out of 20.00

Create a Python Function to find the total number of distinct ways to get a change of 'target' from an unlimited supply of coins in set 'S'.

For example:

Test	Input	Result
`count(S, len(S) - 1, target)`	3 4 1 2 3	The total number of ways to get the desired change is 4

Answer: (penalty regime: 0 %)

Reset answer

```
def count(S, n, target):


    ###################  Add Your Code Here #################
    if target == 0:
        return 1
    if target < 0 or n < 0:
        return 0
    incl = count(S, n, target - S[n])
    excl = count(S, n - 1, target)
    return incl + excl



if __name__ == '__main__':
    S = []#[1, 2, 3]
    n=int(input())
    target = int(input())
    for i in range(n):
        S.append(int(input()))
    print('The total number of ways to get the desired change is',
        count(S, len(S) - 1, target))
```

	Test	Input	Expected	Got	
✓	`count(S, len(S) - 1, target)`	3 4 1 2 3	The total number of ways to get the desired change is 4	The total number of ways to get the desired change is 4	✓
✓	`count(S, len(S) - 1, target)`	3 11 1 2 5	The total number of ways to get the desired change is 11	The total number of ways to get the desired change is 11	✓

Passed all tests! ✓

Correct

Marks for this submission: 20.00/20.00.