Started on	Friday, 8 August 2025, 9:26 AM
State	Finished
Completed on	Friday, 8 August 2025, 10:09 AM
Time taken	42 mins 34 secs
Grade	**80.00** out of 100.00

Question **1**

Correct

Mark 20.00 out of 20.00

Create a python function to compute the fewest number of coins that we need to make up the amount given.

For example:

Test	Input	Result
`ob1.coinChange(s,amt)`	3 11 1 2 5	3

Answer: (penalty regime: 0 %)

Reset answer

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

```
class Solution(object):
   def coinChange(self, coins, amount):
       dp = [float('inf')] * (amount + 1)
       dp[0]=0
       for coin in coins:
           for i in range(coin, amount + 1):
               dp[i] = min(dp[i], dp[i - coin] + 1)
       return dp[amount] if dp[amount]!=float('inf') else -1

ob1 = Solution()
n=int(input())
s=[]
amt=int(input())
for i in range(n):
    s.append(int(input()))


print(ob1.coinChange(s,amt))
```

	Test	Input	Expected	Got	
✔	ob1.coinChange(s,amt)	3 11 1 2 5	3	3	✔
✔	ob1.coinChange(s,amt)	3 12 1 2 5	3	3	✔

	Test	Input	Expected	Got	
✔	ob1.coinChange(s,amt)	3 22 1 2 5	5	5	✔

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 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

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

```
class Solution:
    def maxSubArray(a,size):
        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	✔

	Test	Input	Expected	Got	
✔	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 **3**

Correct

Mark 20.00 out of 20.00

Write a Python program using A Naive recursive implementation of Minimum Cost Path Problem.

For example:

Input	Result
3 3	8

Answer: (penalty regime: 0 %)

Reset answer

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

```
R = int(input())
C = int(input())
import sys
def minCost(cost, m, n):
    ###########  Add your Code Here ##################
    if (n < 0 or m < 0):
        return sys.maxsize
    elif (m == 0 and n == 0):
        return cost[m][n]
    else:
        return cost[m][n] + min( minCost(cost, m-1, n-1),
                                 minCost(cost, m-1, n),
                                 minCost(cost, m, n-1) )
def min(x, y, z):
    if (x < y):
        return x if (x < z) else z
    else:
        return y if (y < z) else z
```

	Input	Expected	Got	
✔	3 3	8	8	✔

Passed all tests! ✔

Correct

Marks for this submission: 20.00/20.00.

Question **4**

Not answered

Mark 0.00 out of 20.00

Write a python program for the implementation of merge sort on the given list of float values.

For example:

Input	Result
5 6.3 2.3 1.5 8.9 4.5	Given array is 6.3 2.3 1.5 8.9 4.5 Sorted array is 1.5 2.3 4.5 6.3 8.9
6 2.3 6.5 4.9 8.7 6.2 2.1	Given array is 2.3 6.5 4.9 8.7 6.2 2.1 Sorted array is 2.1 2.3 4.9 6.2 6.5 8.7

Answer: (penalty regime: 0 %)

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

Question **5**

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):
    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 = []#[1, 3, 6, 3, 2, 3, 6, 8, 9, 5]
n = int(input()) #len(arr)
for i in range(n):
    arr.append(int(input()))
```

	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.