Started on	Thursday, 3 October 2024, 1:19 PM
State	Finished
Completed on	Thursday, 3 October 2024, 1:33 PM
Time taken	13 mins 41 secs
Grade	**80.00** out of 100.00

Question **1**

Incorrect

Mark 0.00 out of 20.00

Write a Python program to find an element in a sorted list using tree recursion. If found print the position in the list otherwise print 0 (Hint: Binary search)

For example:

Input	Result
7 88 55 77 22 11 99 33 55	The sorted list is [11, 22, 33, 55, 77, 88, 99] 4

Answer: (penalty regime: 0 %)

```
def strong(num):
    t=num
    rem=0
    sum=0
    while(t>0):
        rem=t%10
        sum+=fact(rem)
        t=t//10
    if(sum==num):
        return True
    else:
        return False
def fact(n):
    p=1
    for i in range(1,n+1):
        p=p*i
    return p
n=int(input())
L=[int(input()) for i in range(n)]
R = list(filter(strong,L))
print(R)
```

	Input	Expected	Got	
✖	7 88 55 77 22 11 99 33 55	The sorted list is [11, 22, 33, 55, 77, 88, 99] 4	[]	✖

Some hidden test cases failed, too.

Your code must pass all tests to earn any marks. Try again.

Show differences

Incorrect

Marks for this submission: 0.00/20.00.

Question **2**

Correct

Mark 20.00 out of 20.00

Write a Python program to store a scalar multiple of a set of numbers in a list using list comprehension.

For example:

Input	Result
3 5 11.5 22 33.23	[11.5, 22.0, 33.23] [57.5, 110.0, 166.14999999999998]

Answer: (penalty regime: 0 %)

```
n=int(input())
scl=int(input())
l=[]
for i in range(n):
    x=float(input())
    l.append(x)
sq_l=[item*scl for item in l]
print(l)
print(sq_l)
```

	Input	Expected	Got	
✔	3 5 11.5 22 33.23	[11.5, 22.0, 33.23] [57.5, 110.0, 166.14999999999998]	[11.5, 22.0, 33.23] [57.5, 110.0, 166.14999999999998]	✔
✔	5 2 2 3.5 6 9 45	[2.0, 3.5, 6.0, 9.0, 45.0] [4.0, 7.0, 12.0, 18.0, 90.0]	[2.0, 3.5, 6.0, 9.0, 45.0] [4.0, 7.0, 12.0, 18.0, 90.0]	✔

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 find whether the given matrix is an identity matrix or not:

if the matrix is an identity matrix ,print True

else print False

For example:

Test	Input	Result
n=int(input()) M=read_matrix(n) print(is_identity(M))	3 1 2 3 4 5 6 7 8 9	False

Answer: (penalty regime: 0 %)

```
def read_matrix(n):
    matrix = [[0]*n for i in range(n)]
    for i in range(n):
        lines = list(map(int,input().split()))
        for j in range(n):
            matrix[i][j]=lines[j]
    return matrix
flag = True
def is_identity(M):
    for i in range(len(M)):
        for j in range(len(M[0])):
            if(i==j and M[i][j]==1):
                flag = True
            if(i!=j and M[i][j]!=0):
                flag = False
    return flag
```

	Test	Input	Expected	Got	
✔	n=int(input()) M=read_matrix(n) print(is_identity(M))	3 1 2 3 4 5 6 7 8 9	False	False	✔
✔	n=int(input()) M=read_matrix(n) print(is_identity(M))	4 1 0 0 0 0 1 0 0 0 0 1 0 0 0 0 1	True	True	✔
✔	n=int(input()) M=read_matrix(n) print(is_identity(M))	2 1 2 3 4	False	False	✔

Passed all tests! ✔

Correct

Marks for this submission: 20.00/20.00.

Question **4**

Correct

Mark 20.00 out of 20.00

Given an array arr[] of size n, its prefix sum array is another array prefixSum[] of the same size,

such that the value of prefixSum[i] is arr[0] + arr[1] + arr[2] ... arr[i]. Write a Python code to generate the prefixSum []

Input : arr[] = {10, 20, 10, 5, 15}
Output : prefixSum[] = {10, 30, 40, 45, 60}

For example:

Test	Input	Result

`n = int(input())`
`arr=createList(n)`
`prefix=fillPrefixSum(arr)`
`print(arr)`
`print(prefix)`
	`3`
	`11`
	`22`
	`33`
		`[11, 22, 33]`
		`[11, 33, 66]`

Answer: (penalty regime: 0 %)

```
def fillPrefixSum(arr):
    prefixSum = [0 for i in range(len(arr))]
    prefixSum[0] = arr[0]
    for i in range(1, len(arr)):
        prefixSum[i] = prefixSum[i - 1] + arr[i]
    return(prefixSum)
def createList(n):
    l=[]
    for i in range(n):
        x=int(input())
        l.append(x)
    return l
```

	Test	Input	Expected	Got	

✔	`n = int(input())`
	`arr=createList(n)`
	`prefix=fillPrefixSum(arr)`
	`print(arr)`
	`print(prefix)`
		`3`
		`11`
		`22`
		`33`
			`[11, 22, 33]`
			`[11, 33, 66]`
				`[11, 22, 33]`
				`[11, 33, 66]`
					✔

✔	`n = int(input())`
	`arr=createList(n)`
	`prefix=fillPrefixSum(arr)`
	`print(arr)`
	`print(prefix)`
		`4`
		`5`
		`8`
		`3`
		`2`
			`[5, 8, 3, 2]`
			`[5, 13, 16, 18]`
				`[5, 8, 3, 2]`
				`[5, 13, 16, 18]`
					✔

Passed all tests! ✔

Correct

Marks for this submission: 20.00/20.00.

Question **5**

Correct

Mark 20.00 out of 20.00

Write a Python Program to extract only the strong numbers from a list using filter

```
Example :145 is a strong number
```

```
Sum of digit factorials  = 1! + 4! + 5!
                         = 1 + 24 + 120
                         = 145
```

For example:

Input	Result
5 2 67 145 40585 60	[2, 145, 40585]

Answer: (penalty regime: 0 %)

```
1
2
3
4  def strong(num):
5      t=num
6      rem=0
7      sum=0
8      while(t>0):
9          rem=t%10
10         sum+=fact(rem)
11         t=t//10
12     if(sum==num):
13         return True
14     else:
15         return False
16 def fact(n):
17     p=1
18     for i in range(1,n+1):
19         p=p*i
20     return p
21 n=int(input())
22 L=[int(input()) for i in range(n)]
```

	Input	Expected	Got	
✔	5 2 67 145 40585 60	[2, 145, 40585]	[2, 145, 40585]	✔

Passed all tests! ✔

Correct

Marks for this submission: 20.00/20.00.