

Name : Avisha Jain

Class: TE IT B 69

UID: 19-ITB69-22

Assignment 5

Code in Python 3

```
1 #lex_auth_01269438594930278448
2
3 def find_pairs_of_numbers (num_list,n):
4     count=0
5     for i in num_list:
6         S=n-i
7         if S in num_list:
8             count+=1
9     return count//2
10
11 num_list=[1, 2, 4, 5, 6]
12 n=6
13 print(find_pairs_of_numbers (num_list,n))
```

Reset Execute Copy Code Verify Submit Last Submission

Execution Result

Output:

2

Functions - Arguments 55m

Variables & Scope 1h 43m

Unit Testing 1h 20m

Assignment Set - 5 2h 55m

Assignment on Function Argumen... 30m

Assignment on Default Argumen... 30m

Assignment on Function Argumen... 30m

Assignment on Local Scope - Le... 45m

Assignment on Global Scope - L... 40m

Exception Handling in Python 1h 45m

Recursion 2h 52m

Assignment Set - 6 3h 45m

Modules & Packages 42m

Contact Us FAQs Terms and Conditions

© 2021, Infosys Ltd.

Code in Python 3

```
9 list1_even(list_of_num)
10 return sum(list1)
11 else:
12     list2=odd(list_of_num)
13     return sum(list2)
14 #Remove pass and write the logic here
15
16 def even(data):
17     l=[]
18     for i in data:
19         if i%2==0:
20             l.append(i)
21     return l
22 #Remove pass and write the logic here
23
24 def odd(data):
25     l=[]
26     for i in data:
27         if i%2!=0:
28             l.append(i)
29     return l
30 #Remove pass and write the logic here
31
32 sample_data = range(1,11)
33
34 print(sum_of_numbers(sample_data,None))
```

Reset Execute Copy Code Verify Submit Last Submission

Execution Result

Output:

55

Functions - Arguments 55m

Variables & Scope 1h 43m

Unit Testing 1h 20m

Assignment Set - 5 2h 55m

Assignment on Function Argumen... 30m

Assignment on Default Argumen... 30m

Assignment on Function Argumen... 30m

Assignment on Local Scope - Le... 45m

Assignment on Global Scope - L... 40m

Exception Handling in Python 1h 45m

Recursion 2h 52m

Assignment Set - 6 3h 45m

Modules & Packages 42m

Contact Us FAQs Terms and Conditions

© 2021, Infosys Ltd.

Assignment on Function Arguments

Code in Python 3

```
1 #lex_auth_01269441810970214471
2
3 def check_double(number):
4     double=2*number
5     s1=""
6     s2=""
7     s3=""
8     s4=""
9     if s3==s4:
10        return True
11    else:
12        return False
13
14 #Provide different values for number and test your program
15 print(check_double(125874))
```

Reset Execute Copy Code Verify Submit Last Submission

Execution Result

Output:

True

Functions - Arguments 55m

Variables & Scope 1h 43m

Unit Testing 1h 20m

Assignment Set - 5 2h 55m

Assignment on Function Argumen... 30m

Assignment on Default Argument... 30m

Assignment on Function Argumen... 30m

Assignment on Local Scope - Le... 45m

Assignment on Global Scope - L... 40m

Exception Handling in Python 1h 45m

Recursion 2h 52m

Assignment Set - 6 3h 45m

Modules & Packages 42m

Contact Us FAQs Terms and Conditions

© 2021, Infosys Ltd.

Assignment on Local Scope - Le...

Code in Python 3

```
1 #lex_auth_012693816779112448160
2
3 def calculate(distance,no_of_passengers):
4     ticketno_of_passengers=0
5     mile=(distance)/10
6     fuel=mile*70
7     earning=ticket-fuel
8     if earning>0:
9         return earning
10    else:
11        return -1
12
13 #Remove pass and write your logic here
14
15
16
17
18 #Provide different values for distance, no_of_passenger and test your program
19 distance=20
20 no_of_passengers=50
21 print(calculate(distance,no_of_passengers))
```

Reset Execute Copy Code Verify Submit Last Submission

Execution Result

Output:

3860.0

Functions - Arguments 55m

Variables & Scope 1h 43m

Unit Testing 1h 20m

Assignment Set - 5 2h 55m

Assignment on Function Argumen... 30m

Assignment on Default Argument... 30m

Assignment on Function Argumen... 30m

Assignment on Local Scope - Le... 45m

Assignment on Global Scope - L... 40m

Exception Handling in Python 1h 45m

Recursion 2h 52m

Assignment Set - 6 3h 45m

Modules & Packages 42m

Contact Us FAQs Terms and Conditions

© 2021, Infosys Ltd.

Assignment on Global Scope - L...

```
3 #Global variable
4 list_of_marks=(12,18,25,24,2,5,18,20,20,21)
5
6 def find_more_than_average():
7     sum=0
8     count=0
9     for i in list_of_marks:
10         sum+=i
11     avg=sum/len(list_of_marks)
12     for i in list_of_marks:
13         if i>avg:
14             count+=1
15     per=(count*100)/len(list_of_marks)
16     return per
17
18 def sort_marks():
19     sorted_marks=sorted(list_of_marks)
20     return sorted_marks
21
22 def generate_frequency():
23     l=[]
24     for i in range(0,26):
25         l.append(list_of_marks.count(i))
26     return l
27
28 print(find_more_than_average())
29 print(generate_frequency())
30 print(sort_marks())
```

Reset Execute Copy Code Verify Submit Last Submission

Execution Result

Output:

```
70.0
[0, 0, 1, 0, 0, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 2, 0, 2, 1, 0, 0, 1, 1]
[2, 5, 12, 18, 18, 20, 20, 21, 24, 25]
```

Contact Us FAQs Terms and Conditions

© 2021, Infosys Ltd.

Assignment 6

Assignment on Recursion - Leve...

```
1 #lex_auth_01269442114344550475
2
3 def is_palindrome(word):
4     l=word.lower()
5     r=l[::-1]
6     if(l==r):
7         return True
8     else:
9         return False
10
11 #Provide different values for word and test your program
12 result=is_palindrome("MadMa")
13 if(result):
14     print("The given word is a Palindrome")
15 else:
16     print("The given word is not a Palindrome")
```

Reset Execute Copy Code Verify Submit Last Submission

Execution Result

Output:

```
The given word is not a Palindrome
```

Contact Us FAQs Terms and Conditions

© 2021, Infosys Ltd.

Assignment on Recursion - Level 1

```
32 def is_prime(num, i):
33     if(i==1):
34         return True
35     elif(num%i==0):
36         return False;
37     else:
38         return(is_prime(num,i-1))
39
40 def find_largest_prime_factor(list_of_factors):
41     max=0
42     for i in range(0,len(list_of_factors)):
43         if(max<list_of_factors[i] and is_prime(list_of_factors[i],list_of_factors[i]-1)):
44             max=list_of_factors[i]
45     return max
46
47 def find_f(num):
48     list_of_factors=find_factors(num)
49     return find_largest_prime_factor(list_of_factors)
50
51 def find_g(num):
52     g=find_f(num)+find_f(num+1)+find_f(num+2)+find_f(num+3)+find_f(num+4)+find_f(num+5)+find_f(num+6)+find_f(num+7)+find_f(num+8)
53     return g
54
55 print(find_g(10))
```

Reset Execute Copy Code Verify Submit Last Submission

Execution Result

Output:

```
The number is: 3523014
[23014, 5230, 523, 352]
The number is: 3523014
[23014, 5230, 523, 352]
66
```

Contact Us FAQs Terms and Conditions

© 2021, Infosys Ltd.

Assignment on Exception Handling

```
1 #lex_auth_01269442760027340879
2
3 def find_smallest_number(num):
4     for value in range(1,1000):
5         list=[]
6         for n in range(1,value+1):
7             if value%n ==0:
8                 list.append(n)
9
10        if len(list) == num:
11            return value
12
13 num=16
14 print("The number of divisors :",num)
15 result=find_smallest_number(num)
16 print("The smallest number having",num," divisors:",result)
```

Reset Execute Copy Code Verify Submit Last Submission

Execution Result

Output:

```
The number of divisors : 16
The smallest number having 16 divisors: 120
```

Contact Us FAQs Terms and Conditions

© 2021, Infosys Ltd.

Assignment on Exception Handling

```
1 #lex_auth_0126944254504222726
2
3 def find_ten_substring(num_str):
4     num_str=list(map(int,num_str))
5     list_a=[]
6     for x in range(0,a):
7         st="" #To put some empty strings
8         z=[] #Temporary list to check sum
9         for y in range(x,a):
10             st+=str(num_str[y])
11             z.append(num_str[y])
12             if sum(z)==10:
13                 list_a.append(st)
14         return list_a
15
16 #Remove pass and write your logic here
17
18 num_str="2825302"
19 print("The number is:",num_str)
20 result_list=find_ten_substring(num_str)
21 print(result_list)
```

Reset Execute Copy Code Verify Submit Last Submission

Execution Result

Output:

```
The number is: 2825302
[28, '82', '253', '2530', '5302']
```

Contact Us FAQs Terms and Conditions

© 2021, Infosys Ltd.

Assignment on Recursion - Level 1

```
1 #lex_auth_01269443477535129681
2
3 def find_duplicates(list_of_numbers):
4     #start writing your code here
5     a=[]
6     for i in range(len(list_of_numbers)):
7         for j in range(i+1,len(list_of_numbers)):
8             if(list_of_numbers[i]==list_of_numbers[j]):
9                 if(list_of_numbers[j] in a):
10                     continue
11                 else:
12                     a.append(list_of_numbers[j])
13                     break
14     return a
15
16 list_of_numbers=[1,2,2,3,3,3,4,4,4,4]
17 list_of_duplicates=find_duplicates(list_of_numbers)
18 print(list_of_duplicates)
```

Reset Execute Copy Code Verify Submit Last Submission

Execution Result

Output:

```
[2, 3, 4]
```

Contact Us FAQs Terms and Conditions

© 2021, Infosys Ltd.

Assignment 7

Assignment on String APIs - Level 1

Code in Python 3

```
1 #lex_auth_01269446319507046499
2
3 from collections import OrderedDict
4 def remove_duplicates(value):
5     #start writing your code here
6     return ''.join(OrderedDict.fromkeys(value))
7 print(remove_duplicates("123456566666ababzz@001234*"))
```

Reset

Execute

Copy Code

Verify

Submit

Last Submission

Execution Result

Output:
123456abz@#*

Contents

Details

Exception Handling in Python

1h 45m

✓

Recursion

2h 52m

...

Assignment Set - 6

3h 45m

✓

Modules & Packages

42m

...

File Handling in Python

30m

...

Assignment Set - 7

5h 15m

...

Assignment on String APIs - Le...

40m

...

Assignment on String APIs - Le...

40m

...

Assignment on List - Level 1 (...

30m

...

Assignment on String APIs - Le...

45m

...

Assignment on Integers - Level...

40m

...

Assignment on List Slice - Lev...

30m

...

Assignment on String APIs - Le...

30m

...

Assignment on List APIs - Leve...

1h

...

Playground

5m

...

inlyTQ: A x Assignm... x (12) Wha... x inlyTQ: A x inlyTQ: A x inlyTQ: A x inlyTQ: A x inlyTQ: A x inlyTQ: A x inlyTQ: A x day-1day x +

infytq.onwingspan.com/en/viewer/hands-on/lex_auth_01269446533799116898_shared?collectionId=lex_auth_012734003600908288382_shared&collectionType=Course

Code in Python 3

```
1 #lex_auth_01269446533799116898
2
3 def check_perfect_number(number):
4     t=number
5     s=0
6     for i in range(1,t):
7         if (t%i)==0:
8             s+=i
9     if s==t:
10        return True
11    else:
12        return False#start writing your code here
13
14 def check_perfectno_from_list(no_list):
15     l=[]
16     for i in range(0,len(no_list)):
17         if no_list[i]!=0 and check_perfect_number(no_list[i])==True:
18             l.append(no_list[i])
19     else:
20         continue
21     return l#start writing your code here
22
23 perfectno_list=check_perfectno_from_list([18,5,4,2,1,28,0])
24 print(perfectno_list)
```

Reset

Execute

Copy Code

Verify

Submit

Last Submission

Execution Result

Output:

[6, 28]

recursion

Assignment Set - 63h 45m ✓

Modules & Packages42m ...

File Handling in Python30m ...

Assignment Set - 75h 15m ...

Assignment on String APIs - Le...40m ...

Assignment on String APIs - Le...40m ...

Assignment on List - Level 1 (...)**30m ...**

Assignment on String APIs - Le...45m ...

Assignment on Integers - Level...40m ...

Assignment on List Slice - Lev...30m ...

Assignment on String APIs - Le...30m ...

Assignment on List APIs - Leve...1h ...

Playground5m ...

Contact Us

FAQs

Terms and Conditions

© 2021, Infosys Ltd.

Assignment on String APIs - Level 1

```
34 flight_set=set(flight_list)
35 for flight in flight_set:
36     count=flight_list.count(flight)
37     passenger_per_flight.append(flight+":"+str(count))
38     return passenger_per_flight
39
40 def sort_passenger_list():
41     #Write the logic to sort the list returned from find_passengers_per_flight() function in the descending order of number of passengers
42     passenger_per_flight=find_passengers_per_flight()
43     for i in range(0,len(passenger_per_flight)):
44         max_indx=i
45         for j in range(i+1, len(passenger_per_flight)):
46             cnt_cnt=int(passenger_per_flight[max_indx][6:])
47             nxt_cnt=int(passenger_per_flight[j][6:])
48             if nxt_cnt>cnt_cnt:
49                 max_indx=j
50         passenger_per_flight[max_indx],passenger_per_flight[i]=passenger_per_flight[i],passenger_per_flight[max_indx]
51     return passenger_per_flight
52
53
54 #Provide different values for airline_name and destination and test your program.
55 print(find_passengers_per_flight("AI"))
56 print(find_passengers_destination("LOW"))
57 print(sort_passenger_list())
58
```

Reset Execute Copy Code Verify Submit Last Submission

Execution Result

Output:

```
6
4
['AI077:4','SI267:3','AI567:2','BA896:1']
```

Contact Us FAQs Terms and Conditions

© 2021, Infosys Ltd.

Assignment on Integers - Level 1

```
1 #lex_auth_01269443664174284882
2 def nearest_palindrome(number):
3     #Start writing your code here
4     number=number-1
5     for i in range(number,number+1000):
6         sum=0
7         temp=i
8         while(i>0):
9             rem=i%10
10            sum=sum*10+rem
11            i=i//10
12        if sum==temp:
13            return temp
14        break
15
16 number=111
17 print(nearest_palindrome(number))
```

Reset Execute Copy Code Verify Submit Last Submission

Execution Result

Output:

```
121
```

Contact Us FAQs Terms and Conditions

© 2021, Infosys Ltd.

Code in Python 3

```
6
7 def rotations(num):
8     f=[num]
9     s=str(num)
10    digit = len(s)
11    powTen = pow(10, digit - 1)
12    for i in range(digit - 1):
13        firstDigit = num // powTen
14        left = (num * 10 - firstDigit *
15              (firstDigit * powTen * 10))
16        f.append(left)
17        num = left
18    return f
19
20 #Remove pass and write your logic here. It should return the list of different combinations of digits of the given number.
21 #Rotation should be done in clockwise direction. For example, if the given number is 197, the list returned should be [197, 971, 719]
22
23 def get_circular_prime_count(limit):
24     c=0
25     for i in range(1,limit):
26         if all (check_prime(i) for i in rotations(i)):
27             c+=1
28     return c #Remove pass and write your logic here.It should return the count of circular prime numbers below the given limit.
29
30 #Provide different values for limit and test your program
31 print(get_circular_prime_count(1000))
```

Reset Execute Copy Code Verify Submit Last Submission

Execution Result

Output:

25

Contact Us FAQs Terms and Conditions

© 2021, Infosys Ltd.

Type here to search

Assignment on String APIs - Le... 40m

Assignment on String APIs - Le... 40m

Assignment on List - Level 1 ... 30m

Assignment on String APIs - Le... 45m

Assignment on Integers - Level... 40m

Assignment on List Slice - Lev... 30m

Assignment on String APIs - Le... 30m

Assignment on List APIs - Leve... 1h

Playground 5m

Appendix 35m

Practice Quizzes 45m

Practice Problems 16h 10m

Assessment - Programming Funda... 46m

Code in Python 3

```
1 #lex_auth_01269445968039936095
2
3 def validate_credit_card_number(card_number):
4     n=str(card_number)
5     l=len(n)
6     for i in range(1,l):
7         l.append(int(i))
8     for i in range(0,len(n),2):
9         l[i]=l[i]*2
10        if l[i]>10:
11            l[i]=l[i]-9
12        if sum(l)%10==0:
13            return True
14        else:
15            return False #Start writing your code here
16
17 card_number= 1456734512345698 #45398696508133101 #1456734512345698 # #5239512608615007
18 result=validate_credit_card_number(card_number)
19 if(result):
20     print("credit card number is valid")
21 else:
22     print("credit card number is invalid")
```

Reset Execute Copy Code Verify Submit Last Submission

Execution Result

Output:

credit card number is invalid

Contact Us FAQs Terms and Conditions

© 2021, Infosys Ltd.

Type here to search

Assignment on String APIs - Le... 40m

Assignment on String APIs - Le... 40m

Assignment on List - Level 1 ... 30m

Assignment on String APIs - Le... 45m

Assignment on Integers - Level... 40m

Assignment on List Slice - Lev... 30m

Assignment on String APIs - Le... 30m

Assignment on List APIs - Leve... 1h

Playground 5m

Appendix 35m

Practice Quizzes 45m

Practice Problems 16h 10m

Assessment - Programming Funda... 46m

A screenshot of a web browser displaying a Python program on the left and a course navigation menu on the right.

The Python code defines a function `validate_all` that checks if a name, phone number, and email ID are valid. It uses flags to track which fields have been validated and prints messages accordingly. The code includes comments explaining its purpose and usage.

```
def validate_all(name,phone_no,email_id):  
    #Start writing your code here  
    # Use the below given print statements to display appropriate messages  
    # Also, do not modify them for verification to work  
    flag=0  
    if validate_name(name):  
        flag=1  
    else:  
        print("Invalid Name")  
        return  
    if validate_phone_no(phone_no):  
        flag=2  
    else:  
        print("Invalid phone number")  
        return  
    if validate_email_id(email_id):  
        flag=3  
    else:  
        print("Invalid email id")  
        return  
    if flag==3:  
        print("All the details are valid")  
  
#Provide different values for name, phone_no and email_id and test your program  
validate_all("Tina","9994599998","tina@yahoo.com"]
```

Below the code are buttons for "Reset", "Execute", "Copy Code", "Verify", "Submit", and "Last Submission".

The "Execution Result" section shows the output: "All the details are valid".

The right sidebar contains a list of course topics with their durations and completion status:

- Assignment on String APIs - Le... (40m) ✓
- Assignment on String APIs - Le... (40m) ✓
- Assignment on List - Level 1 L... (30m) ✓
- Assignment on String APIs - Le... (45m) ✓
- Assignment on Integers - Level... (40m) ✓
- Assignment on List Slice - Lev... (30m) ✓
- Assignment on String APIs - Le... (30m) ...
- Assignment on List APIs - Leve... (1h) ✓
- Playground (5m) ...
- Appendix (35m) ...
- Practice Quizzes (45m) ...
- Practice Problems (16h 10m) ...
- Assessment - Programming Funda... (46m) ...