In [ ]:

```
#Assigment 2
```

Task 1:

In [ ]:

```
#Question 1.1
```

In [45]:

```
#User defined my reduce function
def myreduce(func,list_val):
    result = list_val[0]
    for i in list_val[1:]:
        result=func(result,i)
    return result;
```

In [57]:

```
#factorial of a number
def fact(n):
    l=list(range(n,0,-1))
    return myreduce(lambda x,y:x*y,l)
```

In [58]:

```
fact(6)
```

Out[58]:

```
720
```

In [63]:

```
myreduce(lambda x,y:x+y,(2,3,4,8))
```

Out[63]:

```
17
```

In [59]:

```
from  functools import reduce
```

In [61]:

```
def fact(n):
    l=list(range(n,0,-1))
    return reduce(lambda x,y:x*y,l)
```

In [64]:

```
fact(6)
```

Out[64]:

```
720
```

In [ ]:

```

```

In [ ]:

```
#Question 1.2
```

In [66]:

```
#User defined my filter function
def myfilter(func,list_val):
    for i in list_val:
        if func(i)==True:
            yield i;
```

In [69]:

```
# list of alphabets
alphabets = ['a', 'b', 'd', 'e', 'i', 'j', 'o']

# list of vowels
vowels = ['a', 'e', 'i', 'o', 'u']

filteredVowels = myfilter(lambda x:x in vowels, alphabets)

print('The filtered vowels are:')
for vowel in filteredVowels:
    print(vowel)
```

```
The filtered vowels are:
a
e
i
o
```

In [70]:

```
filteredVowels = filter(lambda x:x in vowels, alphabets)

print('The filtered vowels are:')
for vowel in filteredVowels:
    print(vowel)
```

```
The filtered vowels are:
a
e
i
o
```

In [ ]:

```
```

In [ ]:

```
#Question 2
```

In [71]:

```
#List Comprehension-1
str1="Acadgild"
list1=[i.upper() for i in str1]
list1
```

Out[71]:

```
['A', 'C', 'A', 'D', 'G', 'I', 'L', 'D']
```

In [72]:

```
#List Comprehension-2
str2='xyz'
list2=[i*n for i in str2 for n in range(1,5)]
list2
```

Out[72]:

```
['x', 'xx', 'xxx', 'xxxx', 'y', 'yy', 'yyy', 'yyyy', 'z', 'zz', 'zzz', 'zzz
z']
```

In [73]:

```
#List Comprehension-3
str3='xyz'
list3=[i*n for n in range(1,5) for i in str3]
list3
```

Out[73]:

```
['x', 'y', 'z', 'xx', 'yy', 'zz', 'xxx', 'yyy', 'zzz', 'xxxx', 'yyyy', 'zzz
z']
```

In [74]:

```
#List Comprehension-4
list4=[[n] for i in range(2,5) for n in range(i,i+3)]
list4
```

Out[74]:

```
[[2], [3], [4], [3], [4], [5], [4], [5], [6]]
```

In [75]:

```
#List Comprehension-5
list5=[[n for n in range(i,i+4)] for i in range(2,6)]
list5
```

Out[75]:

```
[[2, 3, 4, 5], [3, 4, 5, 6], [4, 5, 6, 7], [5, 6, 7, 8]]
```

In [76]:

```
#List Comprehension-6
list6=[(i,n) for n in range(1,4) for i in range(1,4)]
list6
```

Out[76]:

```
[(1, 1), (2, 1), (3, 1), (1, 2), (2, 2), (3, 2), (1, 3), (2, 3), (3, 3)]
```

In [ ]:

```

```

In [77]:

```
#Question-3
def longestWord(word_list):
    word=word_list[0]
    for i in word_list:
        if len(word)<len(i):
            word=i
    return print("The longest word is ",word,"having length of ",len(word))
```

In [78]:

```
words=['abc,cdcvj','fhjduyeg','happiness','dedication','hjfhhfuirfurffhfj','sde']
```

In [79]:

```
longestWord(words)
```

```
The longest word is  hjfhhfuirfurffhfj having length of  17
```

In [ ]:

```

```

Task-2

In [80]:

```
#Question-1.1
class triangle:
    def __init__(self,sideA,sideB,sideC):
        self.a=sideA
        self.b=sideB
        self.c=sideC

class area(triangle):
    def __init__(self,*args):
        super().__init__(*args)

    def calculate(self):
        s=((self.a+self.b+self.c)/2)
        tri_area=area = (s*(s-self.a)*(s-self.b)*(s-self.c)) ** 0.5
        return print("Area of the triangle is: ",round(tri_area,3))
```

In [81]:

```
s=area(3,5,7)
```

In [82]:

```
s.calculate()
```

```
Area of the triangle is:  6.495
```

In [ ]:

```

```

In [13]:

```
#Question-1.2
def filter_long_words():
    words=(input("Enter list of words separated by a comma:")).split(",")
    length=int(input("Enter the minimum length of word desired: "))
    words_result=list(filter(lambda x:len(x)>length,words))
    return print("The length of words greater than the number entered are:",words_result)
```

In [14]:

```
filter_long_words()
```

```
Enter list of words separated by a comma:one, upon, a,time,there,was,lionkin
g,in,the,jungle
Enter the minimum length of word desired: 3
The length of words greater than the number entered are: [' upon', 'time',
'there', 'lionking', 'jungle']
```

In [5]:

```

```

In [20]:

```
#Question-2.1
def word_length(word_list):
    lengths=[]
    for i in word_list:
        lengths.append(len(i))

    #lengths=list(map(lambda x:len(x),word_list))
    return print("The list of word lenngths are: ",lengths)
```

In [83]:

```
word_length(['upon', 'time', 'there', 'lionking', 'jungle'])
```

```
The list of word lenngths are:  [4, 4, 5, 8, 6]
```

In [ ]:

```

```

In [37]:

```
#Question-2.2

# function that identify vowels
def is_vowel(alphabet):
    vowels = ['a', 'e', 'i', 'o', 'u']

    if(alphabet in vowels):
        return True
    else:
        return False
```

In [84]:

```
is_vowel('o')
```

Out[84]:

```
True
```

In [85]:

```
is_vowel('j')
```

Out[85]:

```
False
```

In [ ]:

```

```