ASSIGNMENT-2

Task 1:

1.1

Write a Python Program to implement your own myreduce() function which works exactly like Python's built-in function reduce()

In [1]:
```
# myreduce function
def myreduce(function,data):

    output = data[0]
     # iteration   on remaining sequence items
    for i in data[1:]:
        output = function(output,i)
    return output    # return output

# two numbers multiplication function
def mult(x,y):
    return x*y

# two numbers addition function
def add(x,y):
    return x+y


data_list = [2,3,6,3,1,9,6,7]

print("Multiplication of all alements of data_list :",myreduce(mult,data_list))
print("Addition of all alements of data_list :",myreduce(add,data_list))
```

```
Multiplication of all alements of data_list : 40824
Addition of all alements of data_list : 37
```

1.2

Write a Python program to implement your own myfilter() function which works exactly like Python's built-in function filter()

In [2]:
```
def myfilter(function,data):
    # An empty list
    out=[]
    for i in data:
        #If  condition applied on item is True  append the item of sequence in  out list.
        if function(i)==True:
            out.append(i)
    #return out as a list
    return out
list_=[2,4,7,9,2,56,23,0,7,-3,-4,12,15]

print("Elements grater than 5 in list :", myfilter(lambda x :x>5,list_))
print("Even numbers in  the list : ", myfilter(lambda x: x%2 !=1, list_))
```

```
Elements grater than 5 in list : [7, 9, 56, 23, 7, 12, 15]
Even numbers in  the list :  [2, 4, 2, 56, 0, -4, 12]
```

```
2.
Implement List comprehensions to produce the following lists.
Write List comprehensions to produce the following Lists
['A', 'C', 'A', 'D', 'G', 'I', ’L’, ‘ D’]
['x', 'xx', 'xxx', 'xxxx', 'y', 'yy', 'yyy', 'yyyy', 'z', 'zz', 'zzz', 'zzzz']
['x', 'y', 'z', 'xx', 'yy', 'zz', 'xx', 'yy', 'zz', 'xxxx', 'yyyy', 'zzzz']
[[2], [3], [4], [3], [4], [5], [4], [5], [6]]
[[2, 3, 4, 5], [3, 4, 5, 6], [4, 5, 6, 7], [5, 6, 7, 8]]
[(1, 1), (2, 1), (3, 1), (1, 2), (2, 2), (3, 2), (1, 3), (2, 3), (3, 3)]
```

In [3]:
```
# List 1
list1 = [i for i in "ACADGILD" ]
list1
```

Out[3]:
```
['A', 'C', 'A', 'D', 'G', 'I', 'L', 'D']
```

In [4]:
```
list2 = [i*j for i in ['x', 'y', 'z'] for j in [1,2,3,4]]
list2
```

Out[4]:
```
['x', 'xx', 'xxx', 'xxxx', 'y', 'yy', 'yyy', 'yyyy', 'z', 'zz', 'zzz', 'zzzz']
```

In [5]:
```
list3=[i*j for i in [1,2,3] for j in ['x','y','z']]
list3
```

Out[5]:
```
['x', 'y', 'z', 'xx', 'yy', 'zz', 'xxx', 'yyy', 'zzz']
```

In [6]:
```
list4=[[i+j] for i in [1,2,3] for j in [1,2,3]]
list4
```

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

In [7]:
```
list5=[[i+j for i in [1,2,3,4]] for j in [1,2,3,4]]
list5
```

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

In [8]:
```
list6=[(i,j) for i in [1,2,3] for j in [1,2,3]]
list6
```

Out[8]:
```
[(1, 1), (1, 2), (1, 3), (2, 1), (2, 2), (2, 3), (3, 1), (3, 2), (3, 3)]
```

Task 2:

1.1

Write a Python Program(with class concepts) to find the area of the triangle using the below formula.
area = (s(*s*-a)(s-b)*(s-c)) * * 0.5

Function to take the length of the sides of triangle from user should be defined in the parent class and function to calculate the area should be defined in subclass.

In [10]:
```
class Polygon():
    def __init__(self,a,b,c):
        self.a=a
        self.b=b
        self.c=c

a=float(input("Enter Side 'a' "))
b=float(input("Enter Side 'b' "))
c=float(input("Enter Side 'c' "))

class Triangle(Polygon):
    def __init__(self,a,b,c):
        Polygon.__init__(self,a,b,c)


    def area(self):
        s = (a+b+c)/2
        return (s*(s-a)*(s-b)*(s-c))** 0.5


t=Triangle(a,b,c)
area=t.area()
print("You entered sides  a = {}, b = {}, c = {}. So  area of this Tringle is = {} sq.unit".format(a,b,c,area) )
```

```
Enter Side 'a' 20
Enter Side 'b' 15
Enter Side 'c' 18
You entered sides  a = 20.0, b = 15.0, c = 18.0. So  area of this Tringle is = 129.75915189303603 sq.unit
```

1.2

Write a function filter_long_words() that takes a list of words and an integer n and returns the list of words that are longer than n.

In [11]:
```
def filter_long_words(words, n):
    return filter(lambda x: len(x) > n, words)


word_list=['python', 'jupyter', 'anaconda','Ipython', 'spyder', 'IDE','pycharm']
list(filter_long_words(word_list,6))
```

Out[11]:
```
['jupyter', 'anaconda', 'Ipython', 'pycharm']
```

2.1

Write a Python program using function concept that maps list of words into a list of integers representing the lengths of the corresponding words .

Hint: If a list [ab,cde,erty] is passed on to the python function output should come as [2,3,4] Here 2,3 and 4 are the lengths of the words in the list

In [12]:
```
def map_word_lenght(words_list):
    word_len=[len(word) for word in words_list]
    return word_len

words=["ab","cde","erty"]
print(' Lenghts of the words in- {} are : {}'.format(words,map_word_lenght(words)))

word_list=["python","jupyter","anaconda","spyder","pycharm"]
print(' Lenghts of the words in- {} are : {}'.format(word_list,map_word_lenght(word_list)))
```

```
 Lenghts of the words in- ['ab', 'cde', 'erty'] are : [2, 3, 4]
 Lenghts of the words in- ['python', 'jupyter', 'anaconda', 'spyder', 'pycharm'] are : [6, 7, 8, 6, 7]
```

2.2

Write a Python function which takes a character (i.e. a string of length 1) and returns True if it is a vowel, False otherwise.

In [13]:
```
def is_vowel(character):
    vowels=["a","e","i","o","u"]
    if character in vowels:
        return True
    else:
        return False

print("'a' is vowel: ", is_vowel('a'))
print("'z' is vowel: ", is_vowel('z'))
print("'i' is vowel: ", is_vowel('i'))
print("'7' is vowel: ", is_vowel('7'))
```

```
'a' is vowel:  True
'z' is vowel:  False
'i' is vowel:  True
'7' is vowel:  False
```