Assignment 2 test results:

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

In [18]:
```
def myreduce(func,param):

    #set intial variable
    comp=0 if func=="add"else 1

    for i in param:
        if func=="add":
            comp= addition(i,comp)
        elif func=="mult":
            comp = multiply(i,comp)
        else:
            return ("Function does not exists in the library")
    return comp

def addition(i,comp):
        return i+comp

def multiply(i,comp):
        return i*comp
```

In [19]:
```
#Testing outputs for add
addition=myreduce("add", [1,2,3,4])
print ('Addition of list numbers is: ', addition)
```

```
Addition of list numbers is:  10
```

In [20]:
```
#Testing multplication
product=myreduce("mult", range(1, 4))
print ('Product of numbers in the given range is: ', product)
```

```
Product of numbers in the given range is:  6
```

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

In [21]:
```
#Custome myfiter function
def myfilter(func,param):

    #set intial variable
    comp=[]

    for i in param:
        if func=="even":
            if isEvenOdd(i):
                comp.append(i)
        elif func=="odd":
            if isEvenOdd(i)==False:
                comp.append(i)
        else:
            return ("Function does not exists in the library")

    return comp

def isEvenOdd(i):
        if i%2==0:
            return True
        else:
            return False
```

In [22]:
```
#Get odd numbers from 1-10
odd=myfilter("odd", range(1, 10))
print ('Odd numbers between 1-10:', odd)
```

```
Odd numbers between 1-10: [1, 3, 5, 7, 9]
```

In [23]:
```
#Get even numbers from the list
even = myfilter("even", [1,2,4,6,8,10,15,20])
print ('Even numbers from the given list:', even)
```

```
Even numbers from the given list: [2, 4, 6, 8, 10, 20]
```

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', 'xxx', 'yyy', 'zzz', '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 [24]:
```
#['A', 'C', 'A', 'D', 'G', 'I', 'L', ' D']
#Input string for list compression
istring="ACADGILD"

#Build List
ilist= [i for i in (istring)]

#Display results
print(ilist)
```

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

In [25]:
```
#['x', 'xx', 'xxx', 'xxxx', 'y', 'yy', 'yyy', 'yyyy', 'z', 'zz', 'zzz', 'zzzz']
#Input string for list compression
istring="xyz"

#Build List
ilist= [chr*count for chr in (istring) for count in range(1,5)]

#Display results
print(ilist)
```

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

In [26]:
```
#['x', 'y', 'z', 'xx', 'yy', 'zz', 'xxx', 'yyy', 'zzz', 'xxxx', 'yyyy', 'zzzz']
#Input string for list compression
istring="xyz"

#Build List
ilist= [chr*count for count in range(1,5) for chr in (istring)]

#Display results
print(ilist)
```

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

In [27]:
```
#[[2], [3], [4], [3], [4], [5], [4], [5], [6]]
#Input list
ilist=[2,3,4]

#Build List
ilist=[[num+count] for count in range(3) for num in ilist]

#Display results
print(ilist)
```

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

In [28]:
```
#[[2, 3, 4, 5], [3, 4, 5, 6], [4, 5, 6, 7], [5, 6, 7, 8]]
#Input list
ilist=[2,3,4,5]

#Build List
ilist=[[num+count for count in range(4)] for num in ilist]

#Display results
print(ilist)
```

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

In [29]:
```
#[(1, 1), (2, 1), (3, 1), (1, 2), (2, 2), (3, 2), (1, 3), (2, 3), (3, 3)]
#Input list
ilist=[1,2,3]

#Build List
ilist=[(num2,num1) for num1 in ilist for num2 in ilist]

#Display results
print(ilist)
```

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

3. Implement a function longestWord() that takes a list of words and returns the longest one.

In [31]:
```
def longestWord(word_list):
    word_new_list=[]
    for item in word_list:
        word_new_list.append((len(item),item))

    word_new_list.sort()

    return word_new_list[-1][1]

word_list=["Python","Panda","metplotlib"]

print("Longest word from the sample list is: ",longestWord(word_list))
```

```
Longest word from the sample list is:  metplotlib
```

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 [33]:
```
#Triangle area class - assumption: User will be entering numberic value
class areaTriangle:
    def __init__(self,a,b,c):
        self.a= a
        self.b= b
        self.c= c
        #self.s= s

    def areacal(self):
        #Calculate perimeter
        self.s= (self.a+self.b+self.c)/2
        return round(((self.s*(self.s-self.a)*(self.s-self.b)*(self.s-self.c)) ** 0.5),2)


#Take user inputs
side1 = int(input("Please enter 1st side of a triangle: "))
side2 = int(input("Please enter 2nd side of a triangle: "))
side3 = int(input("Please enter 3rd side of a triangle: "))

#Create class object
triangleArea=areaTriangle(side1,side2,side3)
area =triangleArea.areacal()
print(f"\nArea of triangle {area} square units")
```

```
Please enter 1st side of a triangle: 5
Please enter 2nd side of a triangle: 6
Please enter 3rd side of a triangle: 7

Area of triangle 14.7 square units
```

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 [34]:
```
#Define function to retuen list of selected words
def filter_long_words(list,n):
    slist =[]
    for item in list:
        if len(item)>n:
            slist.append(item)

    return slist

#Sample input list
list=["HP","Oracle","Intel","Microsoft","Google"]

#Display result for words longer than 5 characters
print(filter_long_words(list,5))
```

```
['Oracle', 'Microsoft', 'Google']
```

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 [35]:
```
#Define function to return length of list of given words
def word_length(list):
    slist =[]
    for item in list:
        slist.append(len(item))

    return slist

#Sample input list
list=["HP","Oracle","Intel","Microsoft"]

#Display result for length of words
print(word_length(list))
```

```
[2, 6, 5, 9]
```

In [ ]:
```
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 [37]:
```
#Function that determines if char is vowel or not
def checkVowel(chr):
    vlist=["a","e","i","o","u"]
    if (chr).lower() in vlist:
        return True
    else:
        return False

chr=input("Please enter character you would like to check if vowel or not:")

while len(chr)>0:

    isVowel = checkVowel(chr)

    print(f"\n Is character '{chr}' that you entered is vowel?: {isVowel}")

    choice = input("\n Do you want to check for another character: Y(es) or N(o):")
    if choice.upper() == "Y":
       chr = input("\n Please enter character you would like to check if vowel or not:")
    else:
        chr=""
```

```
Please enter character you would like to check if vowel or not:w

 Is character 'w' that you entered is vowel?: False

 Do you want to check for another character: Y(es) or N(o):y

 Please enter character you would like to check if vowel or not:u

 Is character 'u' that you entered is vowel?: True

 Do you want to check for another character: Y(es) or N(o):y

 Please enter character you would like to check if vowel or not:e

 Is character 'e' that you entered is vowel?: True

 Do you want to check for another character: Y(es) or N(o):n
```