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()

```
def myreduce(fun, lst):
    s = lst[0]

    for i in range(1, len(lst)):
        s = summ(s, lst[i])

    return s

def summ(x1, x2):
    return x1 + x2

print(myreduce(summ, [4,6,7,9,5]))
```

```
31
```

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

```
def evenn(x1):
    if x1 % 2 == 0:
        return x1
    else:
        return False

def my_filter(f,l):
    a=[]
    for i in range(0,len(l)):
        s = f(l[i])
        if s!= False:
            a.append(s)
```

print(a)

my_filter(evenn, [4,6,7,9,10,12,85,987,98456])

```
def evenn(x1):
    if x1 % 2 == 0:
        return x1
    else:
        return False

def my_filter(f,l):
    a=[]
    for i in range(0,len(l)):
        s = f(l[i])
        if s!= False:
            a.append(s)
    print(a)

my_filter(evenn, [4,6,7,9,10,12,85,987,98456])
```

```
[4, 6, 10, 12, 98456]
```

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']

a= "ACADGILD"
b=[a[i] for i in range(len(a))]
b

```
a= "ACADGILD"
b=[a[i] for i in range(len(a))]
b
```

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

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

a=['x','y','z']
*b=[a[i]*j for i in range(len(a)) for j in range(1,5)]*
b

```
a=['x','y','z']
b=[a[i]*j for i in range(len(a)) for j in range(1,5)]
b
```

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

- ['x', 'y', 'z', 'xx', 'yy', 'zz', 'xx', 'yy', 'zz', 'xxxx', 'yyyy', 'zzzz']

a=['x','y','z']
*b=[a[j]*i for i in range(1,5) for j in range(len(a))]*
b

```
a=['x','y','z']
b=[a[j]*i for i in range(1,5) for j in range(len(a))]
b
```

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

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

a=[2,3,4]
b=[[a[j]+i] for i in range(0,3) for j in range (0,len(a))]
b

```
a=[2,3,4]
b=[[a[j]+i] for i in range(0,3) for j in range (0,len(a))]
b
```

```
[[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]]

a=[2,3,4,5]
b=[[a[j]+i for i in range(0,4)] for j in range (0,len(a))]
b

```
a=[2,3,4,5]
b=[[a[j]+i for i in range(0,4)] for j in range (0,len(a))]
b
```

```
[[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)]

a=(1,2,3)

b=[(j,i) for i in range(1,len(a)+1) for j in range (1,len(a)+1)]

b

```
a=(1,2,3)
b=[(j,i) for i in range(1,len(a)+1) for j in range (1,len(a)+1)]
b
```

```
[(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.

def longestword(lst):

```
a=0
for i in lst:
    if len(i) > a:
        a=len(i)
        b= i
return b
```

longestword(["Neeraj Varshney","Excercise","Python"])

```
def longestword(lst):
    a=0
    for i in lst:
        if len(i) > a:
            a=len(i)
            b= i
    return b
```

```
longestword(["Neeraj Varshney","Excercise","Python"])
```

```
'Neeraj Varshney'
```

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.

```
class sides:
    def __init__(self):
        self.a= int(input("Enter 1st side: "))
        self.b= int(input("Enter 2nd side: "))
        self.c= int(input("Enter 3rd side: "))


class areaa(sides):
    def __init__(self,*args):
        super(areaa,self).__init__(*args)
    def get_area(self):
        s= (self.a+self.b+self.c)/2
        return(s*(s-self.a)*(s-self.b)*(s-self.c)) ** 0.5


are= areaa()
print("Area of triangle is :",are.get_area())
```

```
class sides:
    def __init__(self):
        self.a= int(input("Enter 1st side: "))
        self.b= int(input("Enter 2nd side: "))
        self.c= int(input("Enter 3rd side: "))


class areaa(sides):
    def __init__(self,*args):
        super(areaa,self).__init__(*args)
    def get_area(self):
        s= (self.a+self.b+self.c)/2
        return(s*(s-self.a)*(s-self.b)*(s-self.c)) ** 0.5


are= areaa()
print("Area of triangle is :" ,are.get_area())
```

```
Enter 1st side: 6
Enter 2nd side: 7
Enter 3rd side: 8
Area of triangle is : 20.33316256758894
```

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.

```
def filter_long_words(lst,a):
    b=[]
    for i in lst:
        if len(i)>a:
            b.append(i)
    print(b)

filter_long_words(["Cat","Dog","Water","Police"],3)
```

```
def filter_long_words(lst,a):
    b=[]
    for i in lst:
        if len(i)>a:
            b.append(i)
    print(b)

filter_long_words(["Cat","Dog","Water","Police"],3)
```

```
['Water', 'Police']
```

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.

```
def countletters(lst):
    b=[]
    for i in lst:
        a=len(i)
        b.append(a)
```

print(b)

countletters(["ab","cde","erty","cat","water"])

```
def countletters(lst):
    b=[]
    for i in lst:
        a=len(i)
        b.append(a)
    print(b)

countletters(["ab","cde","erty","cat","water"])
```

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

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.

def checkvowel(str):
if (str == 'a' or str== 'e'or str=='i' or str== 'o' or str== 'u' or str== 'A' or str =='E' or str=='I' or str== 'O' or str== 'U'):
return True
else:
return False

checkvowel('A')

```
def checkvowel(str):
    if (str == 'a' or str== 'e'or str=='i' or str== 'o' or str== 'u' or str== 'A' or str =='E' or str=='I' or
       str== 'O' or str== 'U'):
        return True
    else:
        return False

checkvowel('A')
```

```
True
```

```
def checkvowel(str):
    if (str == 'a' or str== 'e'or str=='i' or str== 'o' or str== 'u' or str== 'A' or str =='E' or str=='I' or
       str== 'O' or str== 'U'):
        return True
    else:
        return False

checkvowel('Q')
```

```
False
```