Assignment 2 - > Task 1

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

In [105]:
```
def myreduce(func, seq):
    val = 0;
    indexs = list( range(1, len(seq)) )
    for index in indexs:
      val = func(seq[index-1], seq[index])
    return val

mumber_lst = [10, 20, 23, 30, 99, 89]
myreduce(lambda a, b: ( a if a > b else b ), mumber_lst)
```

Out[105]: 99

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

In [119]:
```
def myfilter(func, seq):
    indexs =list( range(0, len(seq)) )
    for index in indexs:
        if func(seq[index]) == True:
            yield seq[index]

number_lst = list( range(0, 40))
list(myfilter(lambda a: a % 5 == 0, number_lst))
```

Out[119]: [0, 5, 10, 15, 20, 25, 30, 35]

3. Implement 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 [80]:
```
listwithCharslist = [['A', 'C', 'A', 'D'],['G', 'I', 'L', 'D']]
print([char for listOfChars in listwithCharslist for char in listOfChars])
print('\n')

lisfOfChars = ['x', 'y', 'z']
listOfMultiplier = [1,2,3,4]
print([char*multiplier for char in lisfOfChar for multiplier in listOfMultiplier]
print('\n')

print([char*multiplier for multiplier in listOfMultiplier for char in lisfOfChars
print('\n')

listofnumbers = [2, 3, 4, 5, 6]
print([[j] for i in range(0, 3) for j in listofnumbers[i:i+3]])
print('\n')

listofnumbers = [2, 3, 4, 5]
print([ [listofnumbers[i], listofnumbers[i]+1, listofnumbers[i]+2, listofnumbers[

print('\n')
listofnumbers = [1, 2, 3]
[(j, i) for i in listofnumbers for j in listofnumbers]
```

```
['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]]
```

Out[80]:
```
[(1, 1), (2, 1), (3, 1), (1, 2), (2, 2), (3, 2), (1, 3), (2, 3), (3, 3)]
```

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

In [108]:
```
def longestWord(seq):
    lword =''
    indexs =list( range(0, len(seq)) )
    for index in indexs:
        if ( len(seq[index]) > len(lword) ):
            lword = seq[index]
    return lword

words = ['Artificial Intelligence','Deep Learning', 'Neural Networks', 'Data Mini
longestWord(words)
```

Out[108]:
```
'Artificial Intelligence'
```

Assignment 2 - > Task 2

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 [98]:
```
class Triangle:
    def __init__(self, a, b, c):
        self.a = a
        self.b = b
        self.c = c

class AreaOfTriangle(Triangle):
    def __init__(self, *args):
       super(AreaOfTriangle, self).__init__(*args)
    def area(self):
        s = (a + b +c) / 2
        return (s * (s - a) * (s - b) * (s - c)) ** 0.5

a = float( input('Enter first side length: ') )
b = float( input('Enter second side lenght: ') )
c = float( input('Enter thrid side length: ' ) )
print('The area of the triangle is %0.2f' % AreaOfTriangle(a , b ,c).area())
```

```
Enter first side length: 23
Enter second side lenght: 12
Enter thrid side length: 34
The area of the triangle is 66.81
```

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

In [112]:
```
def filter_long_words(seq, n):
    return [word for word in seq if len(word) > n ]

words = ['Artificial Intelligence','Deep Learning', 'Neural Networks', 'Data Mini
filter_long_words(words, 14)
```

Out[112]:
```
['Artificial Intelligence', 'Neural Networks']
```

3. 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 [115]:
```
def map_word_with_length(seq):
    return [len(word) for word in seq]

words = ['Artificial Intelligence','Deep Learning', 'Neural Networks', 'Data Mini
map_word_with_length(words)
```

Out[115]:
```
[23, 13, 15, 11, 13]
```

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

In [122]:
```
def isvowel(char):
    return 'aeiou'.count(char.lower()) != 0

print(isvowel('i'))
```

```
True
```