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

In [1]:
```
def myreduce(func,seq):

    result=seq[0]
    for i in seq[1:]:
        result=func(result,i)
    return result
def sum(x,y):    return x+y
x=myreduce(sum,[2,2,6])
print(x)
```

```
10
```

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

In [2]:
```
# Custom filter function
def myfilter(anyfunc, sequence):

 # Initialize empty list
 result = []
 # iterate over sequence of items in sequence and apply filter function
 for item in sequence:
    if anyfunc(item):
        result.append(item)

 # return funal output
 return result

def ispositive(x):
    if (x <= 0):
        return False
    else:
        return True

print ("Filter only positive Integers on list [0,-1,-2,3,4,5] using custom fil
ter function is\t" \
       + str(myfilter(ispositive, [0,-1,-2,3,4,5])))
```

```
Filter only positive Integers on list [0,-1,-2,3,4,5] using custom filter fun
ction is        [3, 4, 5]
```

Write List comprehensions to produce the following Lists

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

In [3]:
```
word = "ACADGILD"
alphabet_list = [ alphabet for alphabet in word ]
print ("ACADGILD => " + str(alphabet_list))
```

```
ACADGILD => ['A', 'C', 'A', 'D', 'G', 'I', 'L', 'D']
```

Write List comprehensions to produce the following List

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

In [10]:
```
input_list = ['x','y','z']
result = [ item*num for item in input_list for num in range(1,5)  ]
print( str(result))
```

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

Write List comprehensions to produce the following List

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

In [11]:
```
input_list = ['x','y','z']
result = [ item*num for num in range(1,5) for item in input_list  ]
print( str(result))
```

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

Write List comprehensions to produce the following List

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

In [6]:
```
input_list = [2,3,4]
result = [ [item+num] for item in input_list for num in range(0,3)]
print("[2,3,4] =>" + str(result))
```

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

Write List comprehensions to produce the following List

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

In [7]:
```
input_list = [2,3,4,5]
result = [ [item+num for item in input_list] for num in range(0,4)  ]
print(result)
```

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

Write List comprehensions to produce the following List

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

In [8]:
```
input_list=[1,2,3]
result = [ (b,a) for a in input_list for b in input_list]
print("[1,2,3] =>" + str(result))
```

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

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

In [9]:
```
from functools import reduce
list_words = ["This","is","a","beautiful","morning","supernatural"]

# Function to compare and reduce list to the result
def longestWord(list_words):
    return reduce( (lambda x,y:y if len(y) > len(x) else x), list_words )

print ('Longest word in array ["This","is","a","beautiful","morning","supernat
ural"] => ' + longestWord(list_words) )
```

```
Longest word in array ["This","is","a","beautiful","morning","supernatural"]
=> supernatural
```

In [ ]:
```

```