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

```
def myreduce(a, list):
    result = list[0]
    for item in list[1:]:
        result = a(result, item)
    return result
def sum(x,y):
    return x + y
print ("Sum on list [8,9,10,13] using custom myreduce function "    + str(myreduce(sum,
[8,9,10,13] )) )
```

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

```
def myfilter(a, b):
 result = []
 for item in b:
  if a(item):
   result.append(item)
 return result

def ispositive(x):
 if (x <= 0):
  return False
 else:
  return True
print ("Filter only positive Integers on list [-0.3,12,-4,-3,6,7]using custom myfilter function"  +
str(myfilter(ispositive, [-0.3,12,-4,-3,6,7])))
```

2. Implement List comprehensions to produce the following lists.
Write List comprehensions to produce the following Lists

```
list1 = ['x','y','z']
list2 = ['x','y','z']
list3 = [2,3,4]
list4 = [2,3,4,5]
list5=[1,2,3]
print("['x','y','z'] ===> " +   str( [ item*num for item in list1 for num in range(1,5)  ]))
```

```
print("['x','y','z'] ===> " +   str([ item*num for num in range(1,5) for item in list2  ]))
print("[2,3,4] ==========>" +  str([ [item+num] for item in list3 for num in range(0,3)]))
print("[2,3,4,5] ========>" +  str([ [item+num for item in list4] for num in range(0,4)  ]))
print("[1,2,3] ==========>" +  str([ (b,a) for a in list5 for b in list5]))
```