In [ ]:

```
# 1.1 Write a Python Program to implement your own myreduce() function which works exactly
```

In [6]:

```
def myreduce(lst):#reduce function will take list and imagine elements in list as x and y
    result = lst[0]
    for yo in lst [1:]:
        result = (result+yo)# here we have given plus operation, so it will add x + y and t
    return result
```

In [7]:

```
#this is how reduce function work by giving single direct result.
lst = [1,2,3,5,6]
myreduce(lst)
```

Out[7]:

```
17
```

In [ ]:

```
#1.2 Write a Python program to implement your own myfilter() function which works exactly l
```

In [8]:

```
def own_filter(num):
    x=0
    for x in num:
        if x%2==0:
            print (x)
```

In [9]:

```
num= (9,5,2,4)
own_filter(num)
```

```
2
4
```

In [ ]:

```
#Implement List comprehensions to produce the following lists.
# Write List comprehensions to produce the following Lists

#['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 [1]:

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

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

In [2]:

```
a = ['x','y','z']
result = [ b*num for num in range(1,5) for b in a ]
print("['x','y','z'] => " +   str(result))
```

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

In [3]:

```
a = [2,3,4]
result = [[b + c] for b in range(0,3) for c in a]
print(str(result))
```

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

In [4]:

```
a = [1,2,3,4]


result = [[b + x for b in a ]for x in range(1,5) ]
print(str(result))
```

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

In [5]:

```
yo=[1,2,3]
result = [ (b,a) for a in yo for b in yo]
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)]
```

In [ ]:

In [ ]:

In [ ]: