

## Assignment 03.

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

Code:

```
def myreduce(anyfunc, sequence):  
  
    # Get first item in sequence and assign to result  
    result = sequence[0]  
    # iterate over remaining items in sequence and apply  
    reduction function  
    for item in sequence[1:]:  
        result = anyfunc(result, item)  
  
    return result  
def sum(x,y): return x + y  
  
print ("Sum on list [1,2,3] using custom reduce  
function " + str(myreduce(sum, [1,2,3]))) )
```

Output:

```
1 def myreduce(anyfunc, sequence):  
2  
3     # Get first item in sequence and assign to result  
4     result = sequence[0]  
5     # iterate over remaining items in sequence and apply reduction function  
6     for item in sequence[1:]:  
7         result = anyfunc(result, item)  
8  
9     return result  
10 def sum(x,y): return x + y  
11  
12 print ("Sum on list [1,2,3] using custom reduce function " + str(myreduce(sum, [1,2,3]))) )  
Sum on list [1,2,3] using custom reduce function 6
```

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

Code:

```
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):
```

```

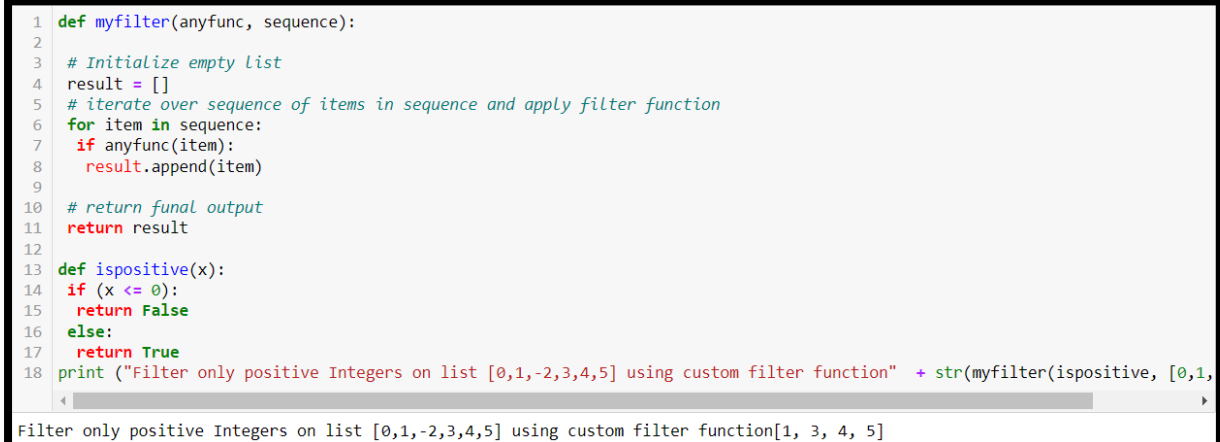
        result.append(item)

# 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 filter function" + str(myfilter(ispositive, [0,1,-
2,3,4,5])))

```

Output:



```

1 def myfilter(anyfunc, sequence):
2
3     # Initialize empty list
4     result = []
5     # iterate over sequence of items in sequence and apply filter function
6     for item in sequence:
7         if anyfunc(item):
8             result.append(item)
9
10    # return funal output
11    return result
12
13 def ispositive(x):
14     if (x <= 0):
15         return False
16     else:
17         return True
18 print ("Filter only positive Integers on list [0,1,-2,3,4,5] using custom filter function" + str(myfilter(ispositive, [0,1,-
2,3,4,5])))

```

Filter only positive Integers on list [0,1,-2,3,4,5] using custom filter function[1, 3, 4, 5]

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']

['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)]

Code:

- word = "ACADGILD"  
alphabet\_list = [ i for i in word ]  
print ( str(alphabet\_list))
- list\_a = "xyz"  
result =[ i\*num for i in list\_a for num in range(1,5)]  
print ( str(result))

- ```
list_b = "xyz"
result = [ i*num for num in range(1,5) for i in list_b ]
print(result)
```

Output:

|                                                                                                              |                                                                                           |
|--------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------|
| <pre>1 word = "ACADGILD" 2 alphabet_list = [ i for i in word ] 3 print ( str(alphabet_list))</pre>           | <pre>['A', 'C', 'A', 'D', 'G', 'I', 'L', 'D']</pre>                                       |
| <pre>1 list_a = "xyz" 2 result = [ i*num for i in list_a for num in range(1,5)] 3 print ( str(result))</pre> | <pre>['x', 'xx', 'xxx', 'xxxx', 'y', 'yy', 'yyy', 'yyyy', 'z', 'zz', 'zzz', 'zzzz']</pre> |
| <pre>1 list_b = "xyz" 2 result = [ i*num for num in range(1,5) for i in list_b ] 3 print(result)</pre>       | <pre>['x', 'y', 'z', 'xx', 'yy', 'zz', 'xxx', 'yyy', 'zzz', 'xxxx', 'yyyy', 'zzzz']</pre> |

- ```
input_lista = [2,3,4]
input_listb = [2,3,4,5]
result = [[item+num] for item in input_lista for num in range(0,3)]
value = [[item+num for item in input_listb] for num in range(0,2)]
print(str(result) ,str(value))
```

- ```
list_c = [2,3,4,5]
result = [ [item+num for item in list_c] for num in range(2,4) ]
print(str(result))
```
- ```
list_d=[1,2,3]
result = [ (b,a) for a in list_d for b in list_d]
print(str(result))
```

Output:

```
1 input_lista = [2,3,4]
2 input_listb = [2,3,4,5]
3 result = [[item+num for item in input_lista for num in range(0,3)]
4 value = [[item+num for item in input_listb for num in range(0,2)]
5 print(str(result) ,str(value))
6
```

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

```
1 list_c = [2,3,4,5]
2 result = [ [item+num for item in list_c for num in range(2,4) ]
3 print(str(result))
```

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

```
1 list_d=[1,2,3]
2 result = [ (b,a) for a in list_d for b in list_d]
3 print(str(result))
```

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