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

**Answer:**   
def myreduce(func, sequence):

# Get the first item in the sequence and assign it to result

result = sequence[0]

# Iterate over the remaining items in the sequence

for item in sequence[1:]:

# Apply the function to the current result and the current item

result = func(result, item)

return result

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

**Answer:**

def myfilter(func, sequence):

# Create an empty list to store the filtered items

result = []

# Iterate over each item in the sequence

for item in sequence:

# Apply the function to the item and check if it returns True

if func(item):

result.append(item)

return result

# Define a simple function to use with myfilter

def is\_even(n):

return n % 2 == 0

# Test the custom myfilter function

sequence = [1, 2, 3, 4, 5, 6]

print(myfilter(is\_even, sequence))

# Output: [2, 4, 6]

**Q.2. Implement List comprehensions to produce the following lists.**

**Answers :**

**python Code**

**a. List comprehension to produce ['A', 'C', 'A', 'D', 'G', 'I', 'L', 'D']**

input\_str = "ACADGILD"

output\_list = [char for char in input\_str]

print(output\_list)

# Output: ['A', 'C', 'A', 'D', 'G', 'I', 'L', 'D']

**b. List comprehension to produce ['x', 'xx', 'xxx', 'xxxx', 'y', 'yy', 'yyy', 'yyyy', 'z', 'zz', 'zzz', 'zzzz']**

**python Code**

input\_list = ['x', 'y', 'z']

output\_list = [char \* i for char in input\_list for i in range(1, 5)]

print(output\_list)

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

**c. List comprehension to produce ['x', 'y', 'z', 'xx', 'yy', 'zz', 'xxx', 'yyy', 'zzz', 'xxxx', 'yyyy', 'zzzz']**

**python Code**

input\_list = ['x', 'y', 'z']

output\_list = [char \* i for i in range(1, 5) for char in input\_list]

print(output\_list)

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

**d. List comprehension to produce [[2], [3], [4], [3], [4], [5], [4], [5], [6]]**

**python Code**

input\_list = [2, 3, 4]

output\_list = [[i + j] for i in input\_list for j in range(3)]

print(output\_list)

# Output: [[2], [3], [4], [3], [4], [5], [4], [5], [6]]

**e. List comprehension to produce [[2, 3, 4, 5], [3, 4, 5, 6], [4, 5, 6, 7], [5, 6, 7, 8]]**

**python Code**

input\_list = [2, 3, 4, 5]

output\_list = [[i + j for i in input\_list] for j in range(4)]

print(output\_list)

# Output: [[2, 3, 4, 5], [3, 4, 5, 6], [4, 5, 6, 7], [5, 6, 7, 8]]

**f. List comprehension to produce [(1, 1), (2, 1), (3, 1), (1, 2), (2, 2), (3, 2), (1, 3), (2, 3), (3, 3)]**

**python Code**

output\_list = [(j, i) for i in range(1, 4) for j in range(1, 4)]

print(output\_list)

# Output: [(1, 1), (2, 1), (3, 1), (1, 2), (2, 2), (3, 2), (1, 3), (2, 3), (3, 3)]