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

In [4]:
```
def myreduce(add,iterable):# We want to get the sum of all the elements present i
    first =iterable[0]
    for i in iterable[1:]:
        first = add(first,i)
    return first

def add(a,b):
    return a+b
#Example 1
lst1 = [1,2,3,4,5]
sum_of_all_nos = myreduce(add,lst1)
print("The sum of all the elements present in the list is :- {}".format(sum_of_al
```

```
The sum of all the elements present in the list is :- 15
```

In [ ]:

In [ ]:

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

In [25]:
```
#Using our own fliter method we will convert float , int, string into integer val
def myfilter(lst):
    new_list =[]
    for i in lst:
        new = int(i)
        new_list.append(new)
    return new_list
#Example 1
l1= [1,11.5,2.6,9.2,3.5]
res_1 = myfilter(l1)
print("Example 1 :-{a}".format(a = res_1))

#Example 2
l2= ['15','85','36','74']
res_2 = myfilter(l2)
print("\nExample 2 :-{a}".format(a = res_2))
```

```
Example 1 :-[1, 11, 2, 9, 3]

Example 2 :-[15, 85, 36, 74]
```

In [ ]:
```

```

2. Implement List comprehensions to produce the following lists.

Write List comprehensions to produce the following Lists

q1) ['A', 'C', 'A', 'D', 'G', 'I', ’L’, ‘ D’]

In [27]:
```
string = "ACADGILD"
LIST=[m for m in string]
print(LIST)
```

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

In [ ]:
```

```

q2) ['x', 'xx', 'xxx', 'xxxx', 'y', 'yy', 'yyy', 'yyyy', 'z', 'zz', 'zzz', 'zzzz']

In [30]:
```
string = "xyz"
LIST=[m*i for m in string for i in range(1,5) ]
print(LIST)
```

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

In [ ]:

q3)['x', 'y', 'z', 'xx', 'yy', 'zz', 'xxx', 'yyy', 'zzz', 'xxxx', 'yyyy', 'zzzz']

In [34]:
```
string = "xyz"
LIST=[m*i for i in range(1,5)  for m in string]
print(LIST)
```

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

In [ ]:

q4) [[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]]

In [29]:
```
lst = [int(x) for x in range(1,9)]
new_lst1=[]
new_lst2=[]
ed=1
for i in range(3):
    ed+=1
    for l in range(ed,ed+3):
        sd =[]
        sd.append(l)
        new_lst1.append(sd)

m=1
for j in range(4):#2nd option
    n = []
    m+=1
    for k in range(m , m+4):
        n.append(k)
    new_lst2.append(n)
print(new_lst1,new_lst2)
```

```
[[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]]
```

In [ ]:

q5) [(1, 1), (2, 1), (3, 1), (1, 2), (2, 2), (3, 2), (1, 3), (2, 3), (3, 3)]

In [7]:
```
lst = [1,2,3]
new_list = []
for i in lst:
    for j in lst:
        a= [j,i]
        a = tuple(a)
        new_list.append(a)

print(new_list)
```

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

In [ ]:
```

```