Problem Statement

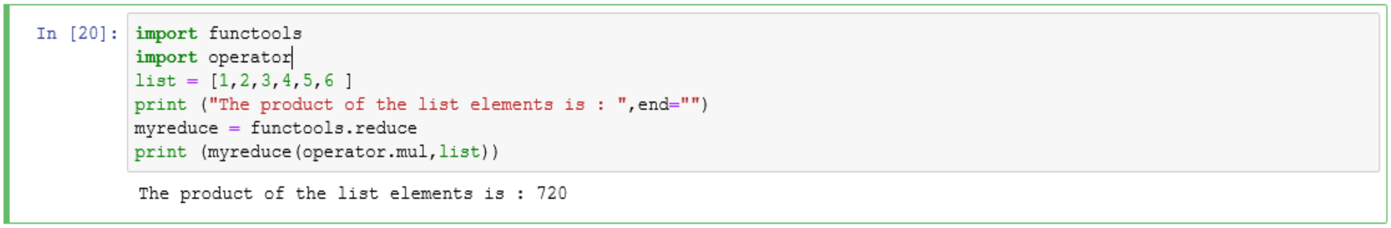
1.1 Write a Python Program to implement your own myreduce() function which works exactly like Python's built-in function reduce()

Solution 1.1

Source Code

```
import functools
import operator
list = [1,2,3,4,5,6 ]
print ("The product of the list elements is : ",end="")
myreduce = functools.reduce
print (myreduce(operator.mul,list))
```

Output Screenshot



```
In [20]: import functools
         import operator
         list = [1,2,3,4,5,6 ]
         print ("The product of the list elements is : ",end="")
         myreduce = functools.reduce
         print (myreduce(operator.mul,list))

         The product of the list elements is : 720
```

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

Solution 1.2

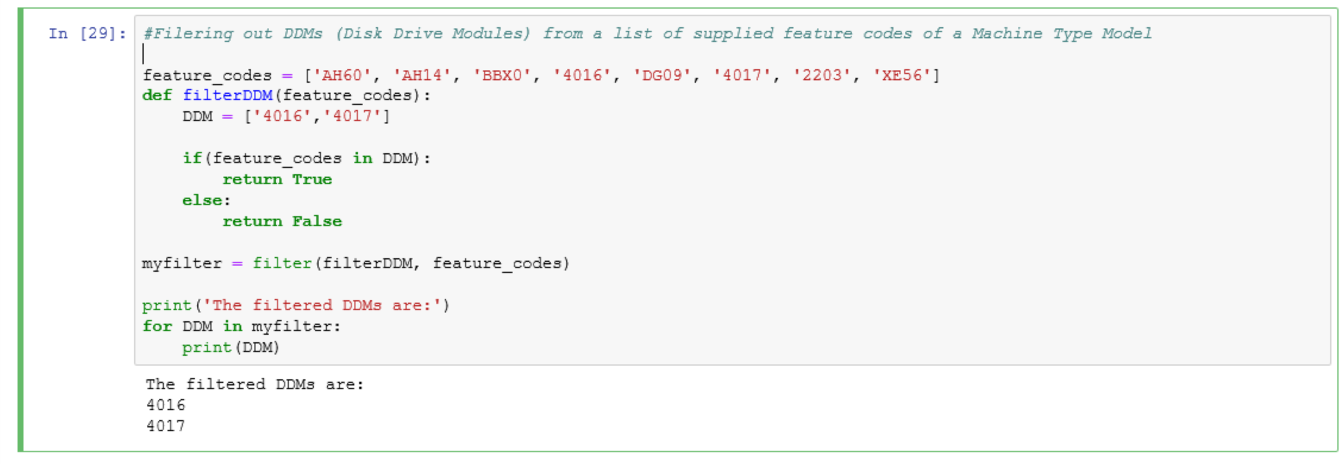
#Filering out DDMs (Disk Drive Modules) from a list of supplied feature codes of a Machine Type Model

Source Code

```
feature_codes = ['AH60', 'AH14', 'BBX0', '4016', 'DG09', '4017', '2203', 'XE56']
def filterDDM(feature_codes):
    DDM = ['4016','4017']

    if(feature_codes in DDM):
        return True
    else:
        return False

myfilter = filter(filterDDM, feature_codes)

print('The filtered DDMs are:')
for DDM in myfilter:
```

print(DDM)

Output Screenshot



```
In [29]: #Filering out DDMs (Disk Drive Modules) from a list of supplied feature codes of a Machine Type Model

         feature_codes = ['AH60', 'AH14', 'BBX0', '4016', 'DG09', '4017', '2203', 'XE56']
         def filterDDM(feature_codes):
             DDM = ['4016','4017']

             if(feature_codes in DDM):
                 return True
             else:
                 return False

         myfilter = filter(filterDDM, feature_codes)

         print('The filtered DDMs are:')
         for DDM in myfilter:
             print(DDM)

         The filtered DDMs are:
         4016
         4017
```

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', 'xx', 'yy', 'zz', '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)]

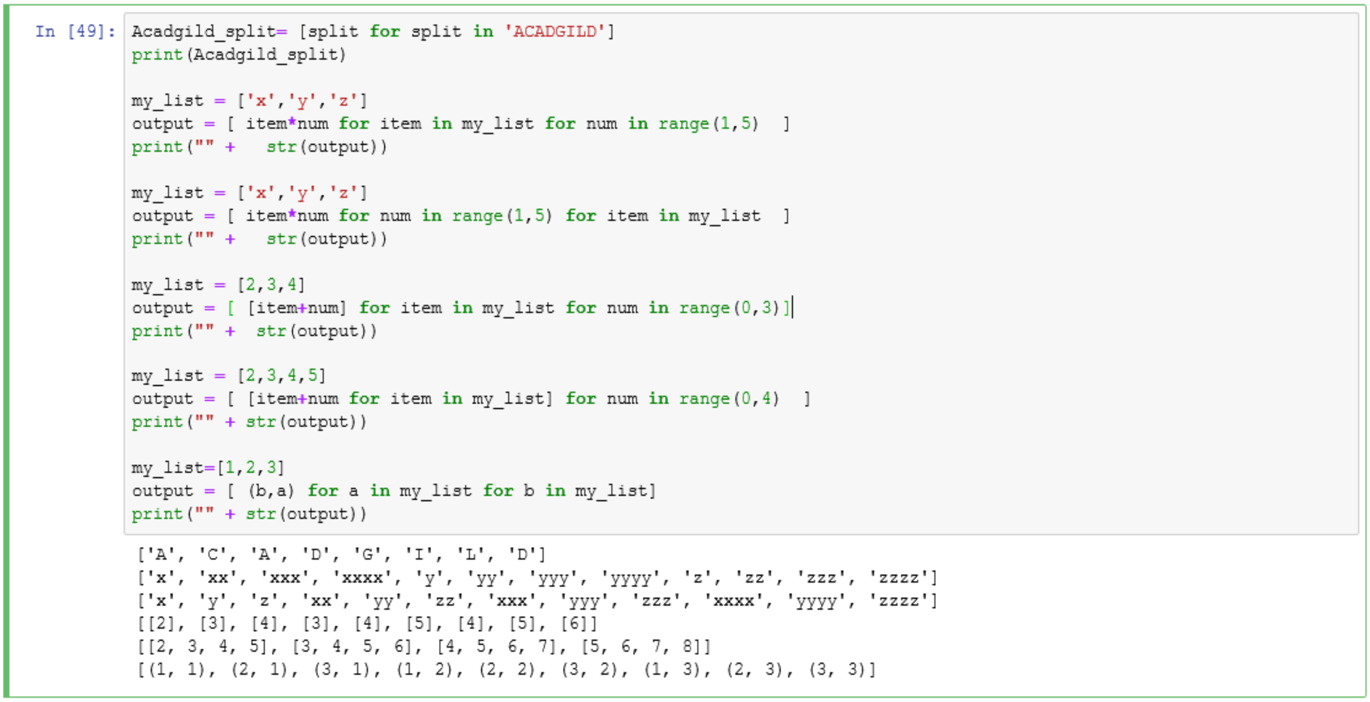
Solution 2

Source Code

```
Acadgild_split= [split for split in 'ACADGILD']

print(Acadgild_split)


my_list = ['x','y','z']

output = [ item*num for item in my_list for num in range(1,5)  ]

print("" +   str(output))


my_list = ['x','y','z']

output = [ item*num for num in range(1,5) for item in my_list  ]
```

```
print("" +   str(output))


my_list = [2,3,4]

output = [ [item+num] for item in my_list for num in range(0,3)]

print("" +  str(output))


my_list = [2,3,4,5]

output = [ [item+num for item in my_list] for num in range(0,4)  ]

print("" + str(output))


my_list=[1,2,3]

output = [ (b,a) for a in my_list for b in my_list]

print("" + str(output))
```

Output Screenshot



```
In [49]: Acadgild_split= [split for split in 'ACADGILD']
         print(Acadgild_split)

         my_list = ['x','y','z']
         output = [ item*num for item in my_list for num in range(1,5)  ]
         print("" +   str(output))

         my_list = ['x','y','z']
         output = [ item*num for num in range(1,5) for item in my_list  ]
         print("" +   str(output))

         my_list = [2,3,4]
         output = [ [item+num] for item in my_list for num in range(0,3)]
         print("" +  str(output))

         my_list = [2,3,4,5]
         output = [ [item+num for item in my_list] for num in range(0,4)  ]
         print("" + str(output))

         my_list=[1,2,3]
         output = [ (b,a) for a in my_list for b in my_list]
         print("" + str(output))

          ['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)]
```

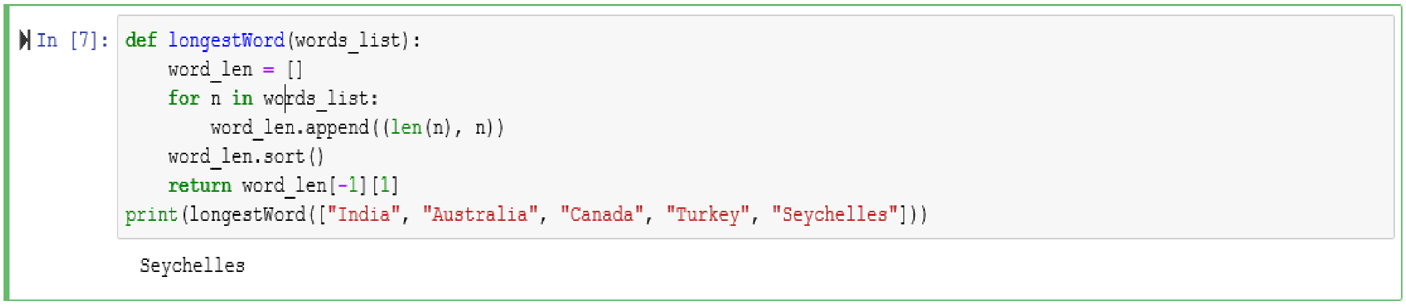
3. Implement a function longestWord() that takes a list of words and returns the longest one.
NOTE: The solution shared through Github should contain the source code used and the screenshot of the output.

Solution 3

Source Code

```
def longestWord(words_list):
    word_len = []
    for n in words_list:
        word_len.append((len(n), n))
    word_len.sort()
    return word_len[-1][1]
print(longestWord(["India", "Australia", "Canada", "Turkey", "Seychelles"]))
```

Output Screenshot



```
In [7]: def longestWord(words_list):
            word_len = []
            for n in words_list:
                word_len.append((len(n), n))
            word_len.sort()
            return word_len[-1][1]
        print(longestWord(["India", "Australia", "Canada", "Turkey", "Seychelles"]))

         Seychelles
```