

## Assignment-2 Solutions

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

Solution:

```
[17]: #Python Program to implement your own myreduce() func
def myreduce(func,iterable):
    return func(iterable)

[18]: sample1=[15,45,65,23,56,63,23.4]

[19]: myreduce(max,sample1)

[19]: 65

[20]: myreduce(sum,sample1)

[20]: 290.4

[21]: sample2='hello'

[22]: myreduce(len,sample2)

[22]: 5
```

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

Solution:

```
Task 1.2

[23]: #Write a Python program to implement your own myfi
def myfilter(func_,iterable_):
    result = []
    for i in iterable_:
        if func_(i):
            result.append(i)
    return result

[24]: #fuction for filter the vowels from the iterables
def fun_(variable):
    letters = ['a', 'e', 'i', 'o', 'u']
    if (variable.lower() in letters):
        return True
    else:
        return False

[25]: str1=['I','r','o','n','m','a','n']

[26]: myfilter(fun_,str1)

[26]: ['I', 'o', 'a']
```

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:

Task 1 (2)

```
[27]: lst1 = [x for x in 'ACADGILD']
```

```
[28]: lst1
```

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

```
[29]: lst2 = [i*j for i in 'xyz' for j in range(1,5)]
```

```
[30]: lst2
```

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

```
[31]: lst3 = [i*j for i in range(1,5) for j in 'xyz']
```

```
[32]: lst3
```

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

```
[33]: lst4 = [[x+y for x in range(2,5) for y in range(3)]
```

```
[34]: lst4
```

```
[34]: [[2], [3], [4], [3], [4], [5], [4], [5], [6]]
```

```
[35]: lst5 = [[x+y for x in range(2,6)] for y in range(4)]
```

```
[36]: lst5
```

```
[36]: [[2, 3, 4, 5], [3, 4, 5, 6], [4, 5, 6, 7], [5, 6, 7, 8]]
```

```
[37]: lst6 = [(y,x) for x in range(1,4) for y in range(1,4)]
```

```
[38]: lst6
```

```
[38]: [(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.

Solution:

```
Task 1 (3)

[39]: def longestWord(words):
        result=words[0]
        for i in words[1:]:
            if len(i)>len(result):
                result = i
        return result

[40]: words = ['Implement', 'a', 'function', 'longest', 'Word', 'that', 'takes', 'a', 'list', 'of', 'words', 'and', 'returns', 'the', 'longest', 'one']

[41]: print("Longest word is:",longestWord(words))

Longest word is: Implement
```

2 1.1 Write a Python Program(with class concepts) to find the area of the triangle using the below formula.

$$\text{area} = (s*(s-a)*(s-b)*(s-c)) ** 0.5$$

Function to take the length of the sides of triangle from user should be defined in the parent class and function to calculate the area should be defined in subclass.

Solution:

```
Task 2 (1.1)

[42]: class sides(object):
        def __init__(self):
            self.a, self.b, self.c = map(int,input("Enter the sides of the triangle in cms seperated by comma").split(','))

        class area(sides):
            def __init__(self):
                sides.__init__(self)
                s= (self.a+self.b+self.c)/2
                tri_area = (s*(s-self.a)*(s-self.b)*(s-self.c))**0.5
                print("Area of the triangle is: {}sq.cm".format(round(tri_area,2)))

triangle = area()

Enter the sides of the triangle in cms seperated by comma 5,7,8
Area of the triangle is: 17.32sq.cm
```

2 1.2 Write a function `filter_long_words()` that takes a list of words and an integer n and returns the list of words that are longer than n.

Solution:

```
Task 2 (1.2)

[43]: def filter_long_words(words,n):
        long=[]
        for i in words:
            if len(i)>n:
                long.append(i)
        return long

[44]: lst_word = ['Write', 'a', 'function', 'filter_long_words()', 'that', 'takes', 'a', 'list', 'of', 'words', 'and', 'an', 'integer', 'and', 'returns', 'the', 'list', 'of']

[45]: filter_long_words(lst_word,4)

[45]: ['Write',
        'function',
        'filter_long_words()',
        'takes',
        'words',
        'integer',
        'returns',
        'words',
        'longer']
```

2.2.1 Write a Python program using function concept that maps list of words into a list of integers representing the lengths of the corresponding words.

Solution:

```
Task 2 (2.1)

[46]: def word_to_num(wrds):
      return len(wrds)

[47]: new = list(map(word_to_num, ['ab', 'cde', 'erty']))

[48]: print(new)

[2, 3, 4]
```

2.2 Write a Python function which takes a character (i.e. a string of length 1) and returns True if it is a vowel, False otherwise.

Solution

```
Task 2 (2.2)

[49]: def is_vowel(strn):
      if strn.lower() in ['a', 'e', 'i', 'o', 'u']:
          result = True
      else:
          result = False
      return result

[50]: is_vowel('f')

[50]: False

[51]: is_vowel('A')

[51]: True
```