

## Untitled note

Notebook: First Notebook

Created: 29-05-2020 11:52

Updated: 29-05-2020 11:59

Author: akram.ahmad2009@gmail.com

### Task 1

#### 1.1

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

```
In [1]: # myreduce function

def myreduce(myfunc, inputList):
    result = inputList[0]
    for member in inputList[1:]:
        result = myfunc(result, member)
    return result

def sum(x,y): return x + y

def multi(x,y): return x * y

def divi(x,y): return x / y

def diff(x,y): return x - y
```

```
In [2]: myreduce(sum, range(1,10))
```

```
Out[2]: 45
```

#### 1.2

```
In [3]: def myfilter(myfunc, inputList1):
        result = []
        for member1 in inputList1:
            if myfunc(member1):
                result.append(member1)
        return result

def ispositive(x):
    if (x <= 0):
        return False
    else:
        return True

def iseven(x):
    if (x % 2 == 0):
        return True
    else:
        return False
```

```
In [4]: print(myfilter(ispositive, [7,2,-9,3,-4,0,1,-1]))
```

```
print(myfilter(iseven, [7,2,-9,3,-4,0,1,-1]))
```

```
[7, 2, 3, 1]
[2, -4, 0]
```

## 2 ¶

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)]
```

```
In [5]: [ number for number in 'ACADGILD' ]
```

```
Out[5]: ['A', 'C', 'A', 'D', 'G', 'I', 'L', 'D']
```

```
In [6]: list1 = ["X", "Y", "Z"]
```

```
[a*i for a in list1 for i in range(1,5)]
```

```
Out[6]: ['x', 'xx', 'xxx', 'xxxx', 'y', 'yy', 'yyy', 'yyyy', 'z', 'zz', 'zzz', 'zzzz']
```

```
In [7]: [a*i for a in (1,2,2,4) for i in list1]
```

```
Out[7]: ['x', 'y', 'z', 'xx', 'yy', 'zz', 'xx', 'yy', 'zz', 'xxxx', 'yyyy', 'zzzz']
```

```
In [8]: list2 = [2,3,4]
```

```
[[a+i for a in list2 for i in range(0,3)]
```

```
Out[8]: [[2], [3], [4], [3], [4], [5], [4], [5], [6]]
```

```
In [9]: list3 = [2, 3, 4, 5]
```

```
[[a+i for a in list3 for i in range(0,4)]
```

```
Out[9]: [[2, 3, 4, 5], [3, 4, 5, 6], [4, 5, 6, 7], [5, 6, 7, 8]]
```

```
In [10]: list4 = [1, 2, 3]
```

```
[[a,b for b in list4 for a in list4]
```

```
Out[10]: [[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.

```
In [11]: def longestWord(x):
         longest = 0
         for i in x:
             if len(i) > longest:
                 longest = len(i)
                 longestW = i
         return longestW
```

```
In [12]: longestWord(["Data", "AI", "Deep Learning", "iNeuron iNtallegence", "Python"])
```

```
Out[12]: 'iNeuron iNtallegence'
```

## Task 2:

1.1

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

$area = (s(s-a)(s-b)(s-c))^{0.5}$

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.

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.

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.

Hint: If a list [ 'ab,cde,erty' ] is passed on to the python function output should come as [2,3,4] Here 2,3 and 4 are the lengths of the words in the list.

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.

## 1.1

```
In [13]: class areaoftriangle:
def __init__(self,side1,side2,side3):
    self.s1=side1
    self.s2=side2
    self.s3=side3

def calc_area(self,a,b,c):
    s=(a+b+c)*0.5
    area = (s*(s-a)*(s-b)*(s-c))**0.5
    return area

def __str__(self):
    return
```

```
In [14]: area=areaoftriangle(1,2,3)
area.calc_area(9,10,11)
```

Out[14]: 42.42640687119285

## 1.2

```
In [15]: class filter_long_words:
def __init__(me,listofword, num):
    me.words = listofword
    me.numbers = num

def filter_words(me, l, n):
    longer = []
    for i in l:
        if len(i) > n:
            longer.append(i)
    return longer
```

```
In [16]: W = filter_long_words(["akram", "akku"], 2)
W.filter_words(["Data", "AI", "Deep Learning", "iNeuron iNtelligence", "Python"], 10)
```

Out[16]: ['Deep Learning', 'iNeuron iNtelligence']

## 2.1

```
In [17]: def str_len(li):
lenth = []
for i in li:
    lenth.append(len(i))
return lenth
```

```
In [18]: str_len(["Data", "AI", "Deep Learning", "iNeuron iNtelligence", "Python"])
```

Out[18]: [4, 2, 13, 20, 6]

## 2.2

```
In [23]: def isVowel(v):
vlist=["a","e","i","o","u","A","E","I","O","U"]
for i in vlist:
    if i==v:
        return True
        break
else:
    return False
```

```
In [24]: isVowel("o")
```

Out[24]: True

