Assignment 4

Q1 -

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.

In [1]:
```
class Polygon:
    def __init__(self,num_of_sides = 3):
        self.num_of_sides = num_of_sides
    def define_sides(self):
        print('Input dimensions of sides')
        self.sides = [float(input()) for _ in range(self.num_of_sides)]


class Area(Polygon):
    def __init__(self,*args):
        Polygon.__init__(self,*args)
    def area_of_triangle(self):
        a,b,c = self.sides
        s = (a + b + c) / 2
        area = (s*(s-a)*(s-b)*(s-c)) ** 0.5
        print("Area of triangle = ",area)
```

In [2]:
```
area = Area()
area.define_sides()
area.area_of_triangle()
```

```
Input dimensions of sides
8
9
```

```
10.5
Area of triangle =  34.93604648139225
```

In [ ]:

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.

In [3]:
```
def filter_long_words(sentence,length):
    short_word_list = []
    for words in sentence:
        if(len(words)<=length):
            short_word_list.append(words)
    return short_word_list
```

In [4]:
```
docs=['the', 'house', 'had',  'a', 'tiny', 'little', 'mouse', 'the', 'cat',
      'saw', 'the', 'mouse', 'the', 'mouse', 'ran', 'away', 'from', 'the',
      'house', 'the', 'cat', 'finally', 'ate', 'the', 'mouse', 'the', 'end',
      'of', 'the', 'mouse', 'story']
print("Filter Function called --> \n", filter_long_words(docs,4))
```

```
Filter Function called -->
 ['the', 'had', 'a', 'tiny', 'the', 'cat', 'saw', 'the', 'the', 'ran', 'away', 'from', 'the', 'the', 'cat', 'ate', 'the', 'the',
'end', 'of', 'the']
```

In [ ]:

Q2

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.

In [5]:
```
# By defining lambda function
```

```
docs=['the', 'house', 'had',  'a', 'tiny', 'little', 'mouse', 'the', 'cat',
      'saw', 'the', 'mouse', 'the', 'mouse', 'ran', 'away', 'from', 'the',
      'house', 'the', 'cat', 'finally', 'ate', 'the', 'mouse', 'the', 'end',
      'of', 'the', 'mouse', 'story']

docs = list(map(lambda word:len(word) , docs))
print("Mapping done ---> \n",docs)
```

```
Mapping done --->
 [3, 5, 3, 1, 4, 6, 5, 3, 3, 3, 3, 5, 3, 5, 3, 4, 4, 3, 5, 3, 3, 7, 3, 3, 5, 3, 3, 2, 3, 5, 5]
```

OR

In [6]:
```
### can also be done by defining function

def string_to_length(word):
    return (len(word))



docs=['the', 'house', 'had',  'a', 'tiny', 'little', 'mouse', 'the', 'cat',
      'saw', 'the', 'mouse', 'the', 'mouse', 'ran', 'away', 'from', 'the',
      'house', 'the', 'cat', 'finally', 'ate', 'the', 'mouse', 'the', 'end',
      'of', 'the', 'mouse', 'story']
docs = list(map(string_to_length , docs))
print("Mapping done ---> \n",docs)
```

```
Mapping done --->
 [3, 5, 3, 1, 4, 6, 5, 3, 3, 3, 3, 5, 3, 5, 3, 4, 4, 3, 5, 3, 3, 7, 3, 3, 5, 3, 3, 2, 3, 5, 5]
```

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.

In [7]:
```
def check_vowel1(char):
    vowels = ['a','e','i','o','u']
    for ch in vowels:
        if(ch==char):
            return True
    return False
```

OR

In [8]:
```
def check_vowel2(char):
    vowels = ['a','e','i','o','u']
    if(char in vowels):
        return True
    return False
```

In [9]:
```
print("Checking vowel with 1st function ---> ",check_vowel1('a'))
print("Checking vowel with 2nd function ---> ",check_vowel2('a'))
print("Checking vowel with 1st function ---> ",check_vowel1('u'))
print("Checking vowel with 2nd function ---> ",check_vowel2('u'))
print("Checking vowel with 1st function ---> ",check_vowel1('v'))
print("Checking vowel with 2nd function ---> ",check_vowel2('v'))
```

```
Checking vowel with 1st function --->  True
Checking vowel with 2nd function --->  True
Checking vowel with 1st function --->  True
Checking vowel with 2nd function --->  True
Checking vowel with 1st function --->  False
Checking vowel with 2nd function --->  False
```

In [ ]:

In [ ]: