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.

First Approach

In [70]:

```
class triangle:
    def __init__(self):
        pass
    def take_dtls(self,s,a,b,c):
        self.a = a
        self.b = b
        self.c = c
        self.s = s
```

In [86]:

```
class calarea(triangle):
    def __init__(self):
        pass

    def area(self,s,a,b,c):
        triangle.take_dtls(self,s,a,b,c)
        print(a)

        ar = (s*(s-a)*(s-b)*(s-c)) ** 0.5
        print(ar)
```

In [87]:

```
x = calarea()
x.area(10,3,3,2)
```

```
3
62.609903369994115
```

Second Approach

In [145]:

```
class triangle:
    def __init__(self):
        pass

    def take_dtls(self):
        self.a = input("Enter a : ")
        self.b = input("Enter b : ")
        self.c = input("Enter c : ")
        self.s = input("Enter s : ")
```

In [173]:

```
class calarea(triangle):
    def __init__(self):
        pass

    def area(self):
        triangle.take_dtls(self)
        ar = (int(self.s)*(int(self.s)-int(self.a))*(int(self.s)-int(self.b))*(int(self.s)-
        print("Triangle Area: {} ".format(ar))
```

In [174]:

```
x = calarea()
```

In [175]:

```
x.area()
```

```
Enter a : 1
Enter b : 1
Enter c : 1
Enter s : 1
Triangle Area: 0.0
```

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 [176]:

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

In [179]:

```
filter_long_words(['a','aa','aaa','aaaa'],2)
```

Out[179]:

```
['aaa', 'aaaa']
```

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.

In [184]:

```
def wordlenmap(word):
    wordlen = []
    for i in word:
        wordlen.append(len(i))
    return wordlen
```

In [185]:

```
wordlenmap(['a','aa','aaa'])
```

Out[185]:

```
[1, 2, 3]
```

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 [193]:

```
def idenvowel(c):
    vowel = ['a','e','i','o','u']
    for i in vowel:
        if i == c:
            return True

    return False
```

In [195]:

```
idenvowel('a')
```

Out[195]:

```
True
```