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 [6]:
```python
class Triangle:
    def __init__(self , a,b,c):
        self.a = a
        self.b = b
        self.c = c
class area_of_triangle(Triangle):
    def area(self):
        s = ((self.a+self.b+self.c)/2)
        return ((s*(s-self.a)*(s-self.b)*(s-self.c))**0.5)



a,b,c = map(int ,input("Enter a ,b ,c by using space:-").split(" "))
area_calculation = area_of_triangle(a,b,c)
print(area_calculation.area())
```

```
Enter a ,b ,c by using space:-10 25 30
117.09371246996997
```

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.

```python
In [13]: def filter_long_words(a , n):
             b=[]
             for i in a:
                 length = len(i)
                 if length>n:
                     b.append(i)
                 else:
                     pass
             return b

         a=["lemon","mangoes","apple","orange","papaya","watermelon"]
         n=5
         #a=list(a for a in input("").split(" "))
         #n=int(input(""))
         print(filter_long_words(a , n))
```

```
['mangoes', 'orange', 'papaya', 'watermelon']
```

```python
In [ ]:
```

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.

```python
In [16]: def fun1(n):
             return len(n)


         a=["ab","cde","erty"]
         b=list(map(fun1, a))
         print(b)
```

```
[2, 3, 4]
```

```python
In [ ]:
```

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 [18]:
```python
def charr(a):
    if a=="a" or a=="e" or a=="i" or a=="o" or a=="u" or a=="A" or a=="E" or a==
        return True
    else:
        return False

s=""
for i in s:
    print(charr(i))
```

```
True
False
False
False
False
True
False
False
False
True
False
False
False
True
True
```

In [ ]: