In [2]:

```
#problem 3

from math import *

def my_Triangle(b,h):
    #Area of a triangle; A = (1/2)*b*h, units are m^2
    A = (1/2)*b*h
    return A

print(f"The area of the triangle with base 1 m  and height 1m is {my_Triangle(1,1)} m^2\n")
print(f"The area of the triangle with base 2 m  and height 1m is {my_Triangle(2,1)} m^2\n")
print(f"The area of the triangle with base 25 m  and height 5m is {my_Triangle(12,5)} m^2\n")
```

```
The area of the triangle with base 1 m  and height 1m is 0.5 m^2

The area of the triangle with base 2 m  and height 1m is 1.0 m^2

The area of the triangle with base 25 m  and height 5m is 30.0 m^2
```

In [20]:

```
#problem 5

from math import *
import numpy as np

def my_Cylinder(r,h):
    #Surface Area of cylinder; SA = 2πr^2+2πrh; units: m^2
    S_A = (2*np.pi*pow(r,2)) + (2*np.pi*r*h)
    #Volume of a cylinder; V = π*r^2*h; units: m^3
    V = np.pi*pow(r,2)*h
    Array1 = np.array([S_A, V])
    return Array1


print(np.around(my_Cylinder(1,5), decimals = 3))
print(np.around(my_Cylinder(2,4), decimals = 3))
```

```
[37.699 15.708]
[75.398 50.265]
```

In [38]:

```
#problem 7

from math import *
import numpy as np

def my_twos(m,n):
    #List Comprehension  = [expression + context]
    array1 =[[2 for i in range(n)]  for j in range(m)]
    return array1

print(my_twos(3,2))
print(my_twos(1,4))
```

```
[[2, 2], [2, 2], [2, 2]]
[[2, 2, 2, 2]]
```

In [39]:

```
#problem 8

from math import *
import numpy as np

subt = lambda x,y: x - y
print("The subtraction of x - y is", subt(5,3))
print("The subtraction of x - y is", subt(200,100))
```

```
The subtraction of x - y is 2
The subtraction of x - y is 100
```

In [47]:

```
#problem 9

def add_string(s1,s2):
    return s1 + s2

s1 = "Programming "
s2 = "is fun!"
add_string(s1, s2)
```

Out[47]:

```
'Programming is fun!'
```

In [14]:

```
#problem 12

from math import *
import numpy as np

r1 = np.arange(1,4)
r2 = np.arange(2,7,2)

def my_donut_area(r1,r2):
    #Using Area of a circle A = pi*r^2
    return np.pi*(r2**2 - r1**2)

my_donut_area(r1,r2)
```

Out[14]:

```
array([ 9.42477796, 37.69911184, 84.82300165])
```

In [18]:

```
#problem 13

from math import *
import numpy as np

def my_within_tolerance(A,a,tol):
    Vector = []
    for x in A:
        if abs(x-a)<tol:
            Vector.append(x)
    return Vector

print(my_within_tolerance([0, 1, 2, 3], 1.5, 0.75))
print(my_within_tolerance(np.arange(0, 1.01, 0.01), 0.5, 0.03))
```

```
[1, 2]
[0.47000000000000003, 0.48, 0.49, 0.5, 0.51, 0.52]
```

In [ ]:

```

```