In [3]:

```
# exercise 5

B = 10

H = 12

A_t = B*H*(1/2)

print("The area of the triangle is",A_t)
```

```
The area of the triangle is 60.0
```

In [7]:

```
# excersise 6

import math

C_r = 5

H = 3

S_A = (2*(math.pi)*C_r*H) + (2*(math.pi)*C_r**2)

print("The Surface area is", S_A)

V = H*math.pi*C_r**2

print("The volume is", V)
```

```
The Surface area is 251.32741228718345
The volume is 235.61944901923448
```

In [9]:

```
#exercises 7&8

import math

x1 = 3

x2 = 5

y1 = 4

y2 = 9

slope = (y2 - y1)/(x2 - x1)

dist = math.sqrt((x2 - x1)**2 + (y2-y1)**2)

print("the slope of two points are", slope)
print("the distance of two points are", dist)
```

```
the slope of two points are 2.5
the distance of two points are 5.385164807134504
```

In [11]:

```
#exercise 9

import math
import numpy as np

fact = math.factorial(6)

print("The factorial of 6 is", fact)
```

```
The factorial of 6 is 720
```

In [17]:

```
#excercise 10


import numpy as np

Input = np.arange(1500, 2010, 1)

# Find whether it is leap year or not
def Year_Check(year):
    return (((year % 4 == 0) and
             (year % 100 != 0)) or
             (year % 400 == 0))

# Answer Initialization
Ans = 0

for elem in Input:
    if checkYear(elem):
        Ans = Ans + 1

# Printing
print("No of leap years are:", Ans)
```

```
No of leap years are: 124
```

In [26]:

```
#exercise 11
import math
import numpy as np

def PI_APPX(k):
    return ((2*math.sqrt(2))/(9801))*((math.factorial(4*k))*(1103 + 26390*k))/((math.factorial(k))**4*396**(4*k))

true = (1/math.pi)

print("The 1/PI value is for  N = 0 is", PI_APPX(0))
print("The 1/PI value is for  N = 1 is", PI_APPX(1))
print("The true 1/PI value is", true)

```

```
The 1/PI value is for  N = 0 is 0.31830987844047015
The 1/PI value is for  N = 1 is 7.743320483521513e-09
The true 1/PI value is 0.3183098861837907
```

In [29]:

```
#exercise 12
import math
import cmath

def appx_sinh(x):
    return (math.exp(x) + math.exp(-1*x))/(2)

def sinh(x):
   return math.sinh(x)

print("The approximate sinh is", appx_sinh(2))
print("The real sinh is", sinh(2))
```

```
The approximate sinh is 3.7621956910836314
The real sinh is 3.626860407847019
```

In [31]:

```
#exercise 13

import math

def Pyth_Id(x):
    return math.cos(x)**2 + math.sin(x)**2

print("The Pythaorean identity for angle PI is", Pyth_Id(math.pi))
print("The Pythaorean identity for angle PI/2 is", Pyth_Id(math.pi/2))
print("The Pythaorean identity for angle PI/4 is", Pyth_Id(math.pi/4))
print("The Pythaorean identity for angle PI/6 is", Pyth_Id(math.pi/6))
```

```
The Pythaorean identity for angle PI is 1.0
The Pythaorean identity for angle PI/2 is 1.0
The Pythaorean identity for angle PI/4 is 1.0
The Pythaorean identity for angle PI/6 is 1.0
```

In [34]:

```
#exercise 14

import math

print("The sin(87) is ", math.sin(math.degrees(87)))
```

```
The sin(87) is  0.8275467546646788
```

In [1]:

```
#exercise 23

import math
print((math.exp(2)*math.sin(math.pi/6)) + (math.log(3,math.exp(1))*math.cos(math.pi/9)) - (5**3))
```

```
-120.27311408976854
```

In [ ]: