Assignment 1: Programming in Python CST 362

Name: **Alby Thekkedan** **Roll No 8** **Class CS6B**

KTU-ID: **MDL20CS129**

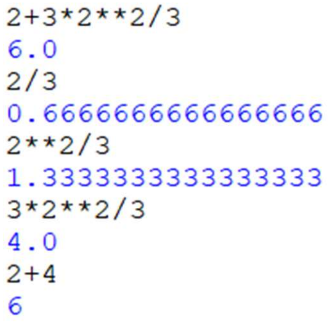
Date of submission: 27-Feb-2023

Learning outcome: Python basics, operators, modules/packages, control statements

1. Python is developed by **Guido van Rossum**
2. How Python got its name - **The name, Python comes from an old BBC television comedy sketch series called Monty Python's Flying Circus**
3. Write the output of the statement 2**3**2. **Output - 512**
4. What is the difference between / and // operators. Write one example

   ‘ / ’ is Normal division the output will be a real value or floating value. eg: 15/4=3.75

   ‘ // ’ is Integer division or Floor division divides and returns integer value eg: 15//4=3
5. How the expression 2+3*2**2/3 is evaluated show the order of evaluation and final output



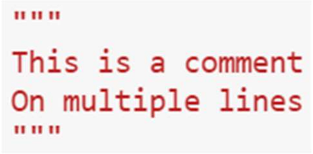
```
2+3*2**2/3
6.0
2/3
0.6666666666666666
2**2/3
1.3333333333333333
3*2**2/3
4.0
2+4
6
```

Ans : 6

6. How single line and multiline comments are added in Python script

   To make single-line comments in Python, add # to the beginning of each line. #a=7

   To make multi-line comments in Python, Use docstrings



```
"""
This is a comment
On multiple lines
"""
```

7. Write the logical operators

   They are **and, or, not**

   The **and** operator checks whether **two conditions are both True simultaneously.** It returns True if both conditions are True. And it returns False if either the condition is False.

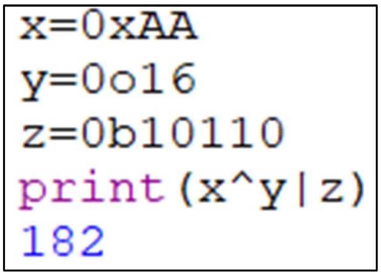
   The **or** operator returns True when **either or both individual conditions are True.** The or operator returns **False** only when **both conditions are False**.

   The **not** operator **If the condition is True, the not operator returns False and vice versa.**

8. x=0xAA, y=0o16, z=0b10110 , find x^y|z

   x is $(170)_{10}$, y is $(14)_{10}$, z is $(22)_{10}$

   Ans 182



```
x=0xAA
y=0o16
z=0b10110
print(x^y|z)
182
```

9. What are bit wise operators

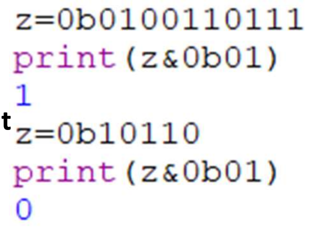
OPERATOR	NAME	DESCRIPTION	SYNTAX
&	Bitwise AND	Result bit 1, if both operand bits are 1; otherwise results bit 0.	x & y
|	Bitwise OR	Result bit 1, if any of the operand bit is 1; otherwise results bit 0.	x | y
~	Bitwise NOT	Inverts individual bits	~x
^	Bitwise XOR	Results bit 1, if any of the operand bit is 1 but not both, otherwise results bit 0.	x ^ y
>>	Bitwise right shift	The left operand's value is moved toward right by the number of bits specified by the right operand.	x>>
<<	Bitwise left shift	The left operand's value is moved toward left by the number of bits specified by the right operand.	x<<

10. a. How to get the last bit of a number. Write the bitwise operation, e/g: x=2 o/p:0, x=3 o/p:1

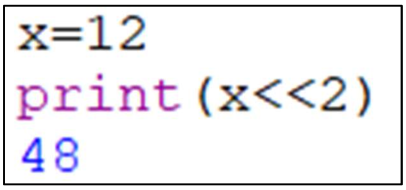
b. x=12, what is the output of x<<2 justify your answer

BITWISE AND operation with binary 01

12 is 1100, left shift by 2 position on left most bit gives 110000, which is $(48)_{10}$



```
z=0b0100110111
print(z&0b01)
1
z=0b10110
print(z&0b01)
0
```



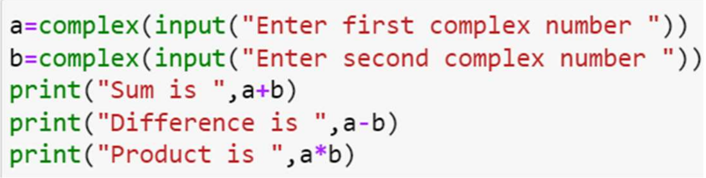
```
x=12
print(x<<2)
48
```

11. What are Boolean data types

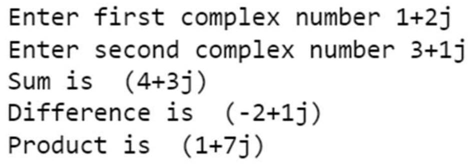
The Boolean value can be of two types only i.e. either True or False.It is used to represent the truth values of the expressions. For example, 1==1 is True whereas 2<1 is False.

Basic Scripting

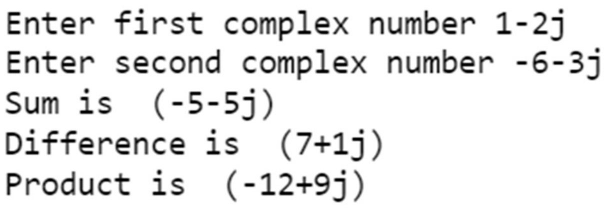
12. Write a Python script which will read two complex numbers and find their sum, difference and product.



```
a=complex(input("Enter first complex number "))
b=complex(input("Enter second complex number "))
print("Sum is ",a+b)
print("Difference is ",a-b)
print("Product is ",a*b)
```

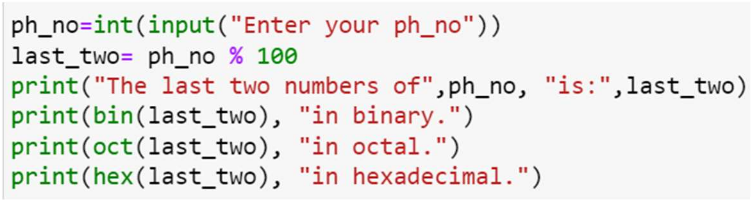


```
Enter first complex number 1+2j
Enter second complex number 3+1j
Sum is  (4+3j)
Difference is  (-2+1j)
Product is  (1+7j)
```

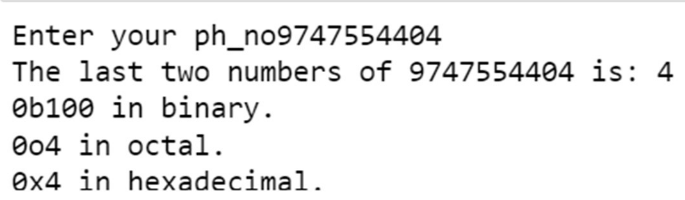


```
Enter first complex number 1-2j
Enter second complex number -6-3j
Sum is  (-5-5j)
Difference is  (7+1j)
Product is  (-12+9j)
```

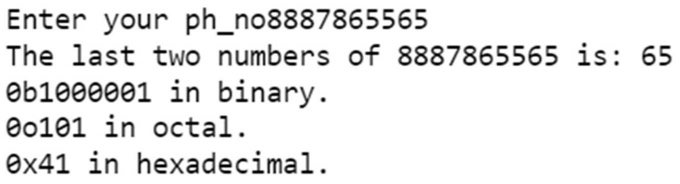
13. Read your phone number and print the last two digit in binary, octal and hex.



```
ph_no=int(input("Enter your ph_no"))
last_two= ph_no % 100
print("The last two numbers of",ph_no, "is:",last_two)
print(bin(last_two), "in binary.")
print(oct(last_two), "in octal.")
print(hex(last_two), "in hexadecimal.")
```

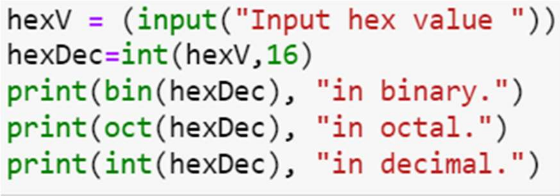


```
Enter your ph_no9747554404
The last two numbers of 9747554404 is: 4
0b100 in binary.
0o4 in octal.
0x4 in hexadecimal.
```

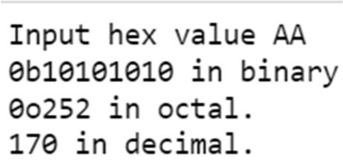


```
Enter your ph_no8887865565
The last two numbers of 8887865565 is: 65
0b1000001 in binary.
0o101 in octal.
0x41 in hexadecimal.
```

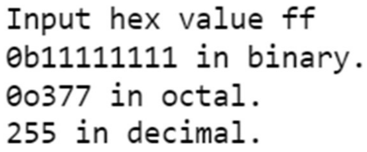
14. Read a hexa decimal number and print the dec, bin and octal equivalent.



```
hexV = (input("Input hex value "))
hexDec=int(hexV,16)
print(bin(hexDec), "in binary.")
print(oct(hexDec), "in octal.")
print(int(hexDec), "in decimal.")
```

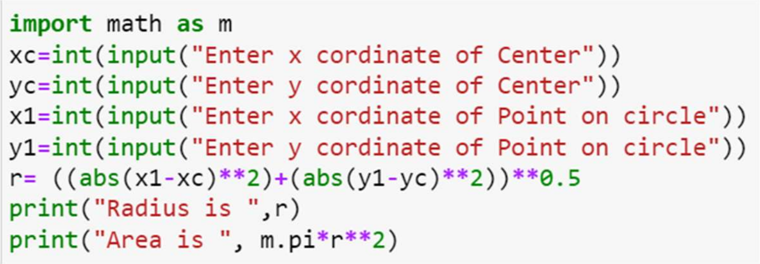


```
Input hex value AA
0b10101010 in binary
0o252 in octal.
170 in decimal.
```

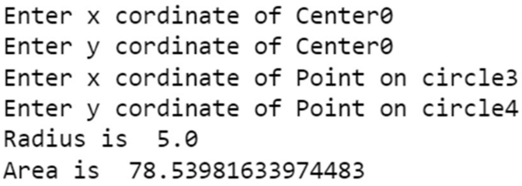


```
Input hex value ff
0b11111111 in binary.
0o377 in octal.
255 in decimal.
```

15. Given the center (xc,yc) and a point on the circle(x1,y1). Find the area

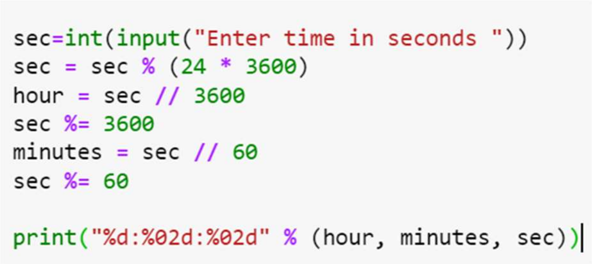


```
import math as m
xc=int(input("Enter x cordinate of Center"))
yc=int(input("Enter y cordinate of Center"))
x1=int(input("Enter x cordinate of Point on circle"))
y1=int(input("Enter y cordinate of Point on circle"))
r= ((abs(x1-xc)**2)+(abs(y1-yc)**2))**0.5
print("Radius is ",r)
print("Area is ", m.pi*r**2)
```

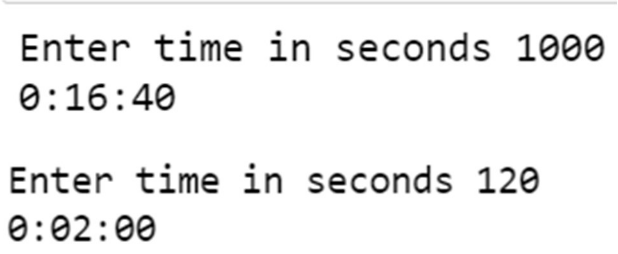


```
Enter x cordinate of Center0
Enter y cordinate of Center0
Enter x cordinate of Point on circle3
Enter y cordinate of Point on circle4
Radius is  5.0
Area is  78.53981633974483
```

16. Write a Python program to read time in seconds and Print in HH:MM:SS format. i/p in seconds :1000 o/p:00:16:40



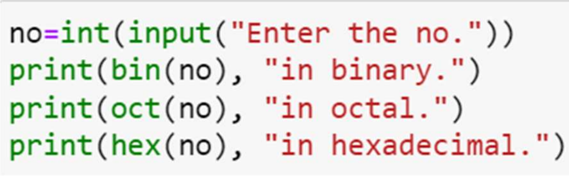
```
sec=int(input("Enter time in seconds "))
sec = sec % (24 * 3600)
hour = sec // 3600
sec %= 3600
minutes = sec // 60
sec %= 60

print("%d:%02d:%02d" % (hour, minutes, sec))
```



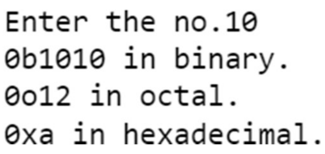
```
Enter time in seconds 1000
0:16:40

Enter time in seconds 120
0:02:00
```

Use built in functions

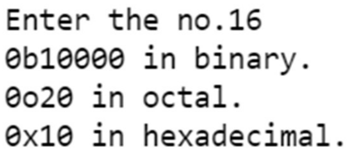
17. Read a number and Print the corresponding binary, oct, hex



```
no=int(input("Enter the no."))
print(bin(no), "in binary.")
print(oct(no), "in octal.")
print(hex(no), "in hexadecimal.")
```

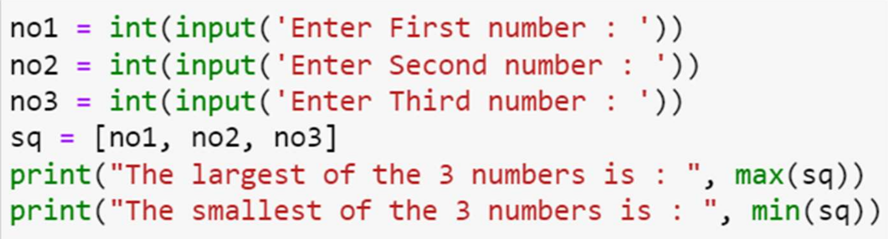


```
Enter the no.10
0b1010 in binary.
0o12 in octal.
0xa in hexadecimal.
```

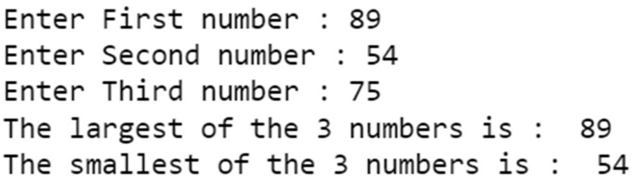


```
Enter the no.16
0b10000 in binary.
0o20 in octal.
0x10 in hexadecimal.
```

18. Read 3 numbers and find the largest and smallest



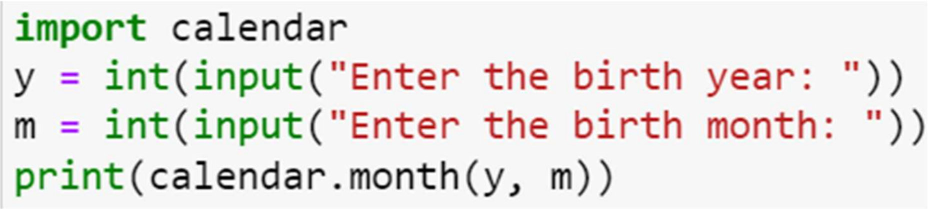
```
no1 = int(input('Enter First number : '))
no2 = int(input('Enter Second number : '))
no3 = int(input('Enter Third number : '))
sq = [no1, no2, no3]
print("The largest of the 3 numbers is : ", max(sq))
print("The smallest of the 3 numbers is : ", min(sq))
```



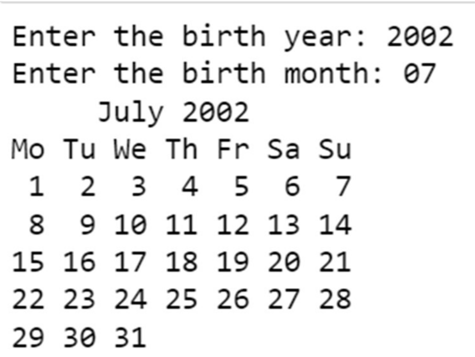
```
Enter First number : 89
Enter Second number : 54
Enter Third number : 75
The largest of the 3 numbers is :  89
The smallest of the 3 numbers is :  54
```

Use modules/packages

19. Print the month calendar depending on your DOB

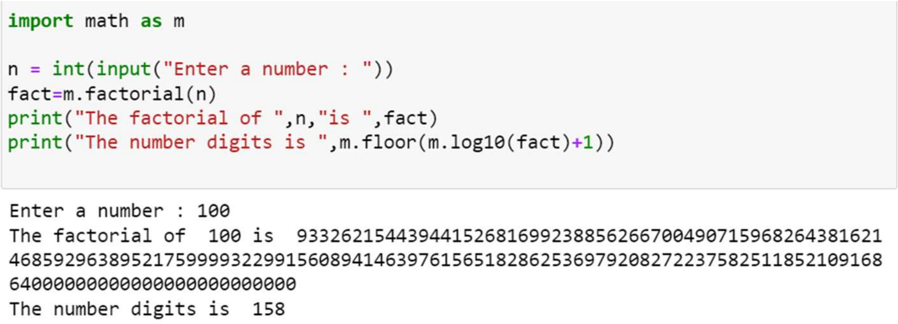


```
import calendar
y = int(input("Enter the birth year: "))
m = int(input("Enter the birth month: "))
print(calendar.month(y, m))
```



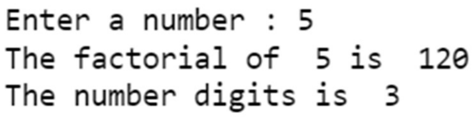
```
Enter the birth year: 2002
Enter the birth month: 07
     July 2002
Mo Tu We Th Fr Sa Su
 1  2  3  4  5  6  7
 8  9 10 11 12 13 14
15 16 17 18 19 20 21
22 23 24 25 26 27 28
29 30 31
```

20. Find the number of digits in the factorial of a given number



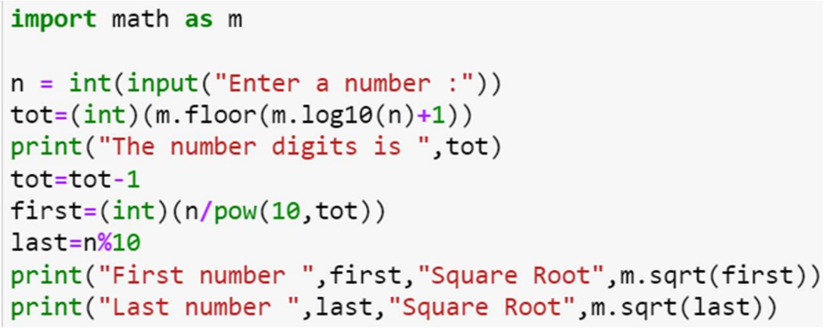
```
import math as m

n = int(input("Enter a number : "))
fact=m.factorial(n)
print("The factorial of ",n,"is ",fact)
print("The number digits is ",m.floor(m.log10(fact)+1))
```

```
Enter a number : 100
The factorial of  100 is  93326215443944152681699238856266700490715968264381621
468592963895217599993229915608941463976156518286253697920827223758251185210916 8
64000000000000000000000000
The number digits is  158
```

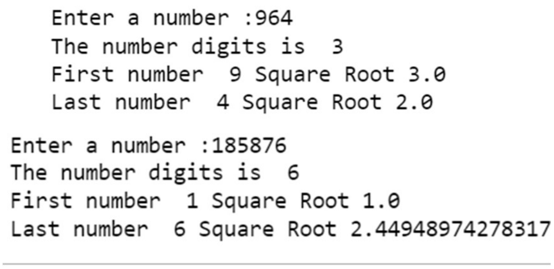


```
Enter a number : 5
The factorial of  5 is  120
The number digits is  3
```

21. Find the sqrt of first and last digit of a number



```
import math as m

n = int(input("Enter a number :"))
tot=(int)(m.floor(m.log10(n)+1))
print("The number digits is ",tot)
tot=tot-1
first=(int)(n/pow(10,tot))
last=n%10
print("First number ",first,"Square Root",m.sqrt(first))
print("Last number ",last,"Square Root",m.sqrt(last))
```

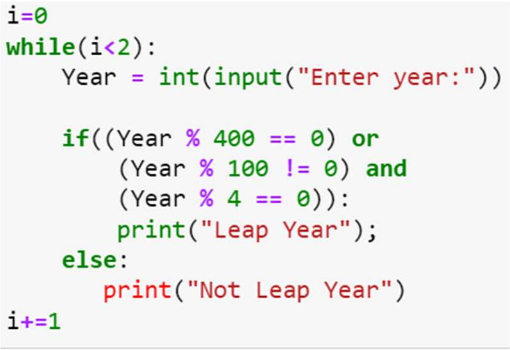


```
Enter a number :964
The number digits is  3
First number  9 Square Root 3.0
Last number  4 Square Root 2.0
```

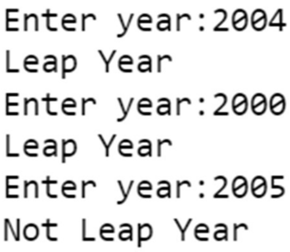
```
Enter a number :185876
The number digits is  6
First number  1 Square Root 1.0
Last number  6 Square Root 2.449489742783178
```

Using if

22. Check whether the given year is leap year or not.

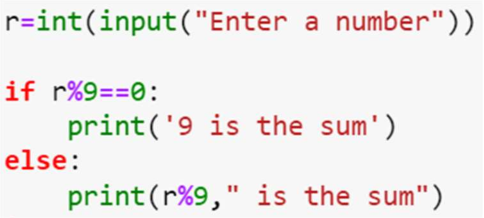


```
i=0
while(i<2):
    Year = int(input("Enter year:"))

    if((Year % 400 == 0) or
        (Year % 100 != 0) and
        (Year % 4 == 0)):
        print("Leap Year");
    else:
       print("Not Leap Year")
i+=1
```

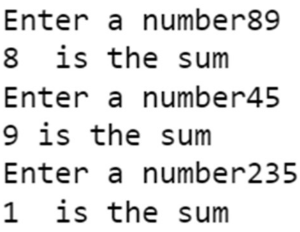


```
Enter year:2004
Leap Year
Enter year:2000
Leap Year
Enter year:2005
Not Leap Year
```

23. Write a python program to read a number and recursively add the digits in it

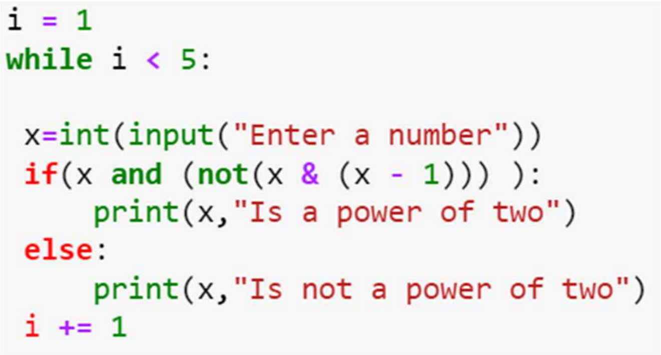


```
r=int(input("Enter a number"))

if r%9==0:
    print('9 is the sum')
else:
    print(r%9," is the sum")
```

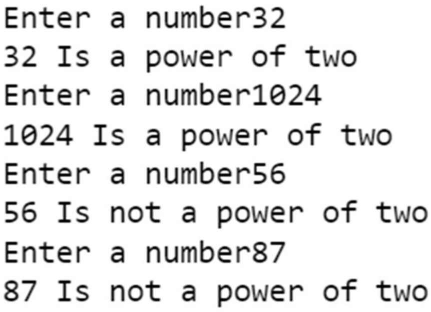


```
Enter a number89
8  is the sum
Enter a number45
9 is the sum
Enter a number235
1  is the sum
```

24. Check whether the given number is power of 2

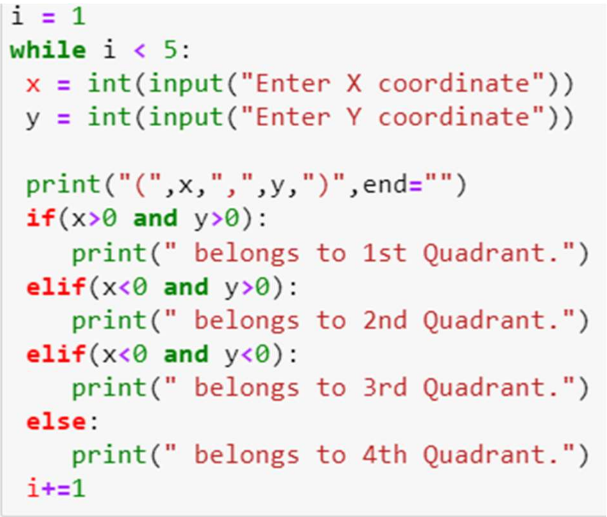


```
i = 1
while i < 5:

 x=int(input("Enter a number"))
 if(x and (not(x & (x - 1))) ):
     print(x,"Is a power of two")
 else:
     print(x,"Is not a power of two")
 i += 1
```

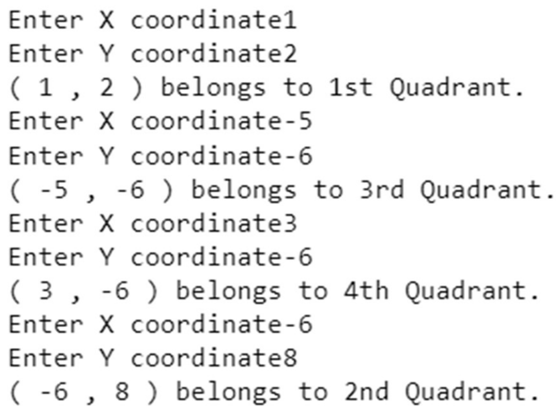


```
Enter a number32
32 Is a power of two
Enter a number1024
1024 Is a power of two
Enter a number56
56 Is not a power of two
Enter a number87
87 Is not a power of two
```

25. Write a program to check the quadrant of a given point(x,y)**(University question)**

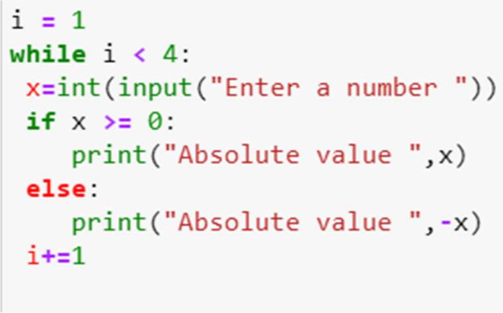


```
i = 1
while i < 5:
 x = int(input("Enter X coordinate"))
 y = int(input("Enter Y coordinate"))

 print("(",x,",",y,")",end="")
 if(x>0 and y>0):
    print(" belongs to 1st Quadrant.")
 elif(x<0 and y>0):
    print(" belongs to 2nd Quadrant.")
 elif(x<0 and y<0):
    print(" belongs to 3rd Quadrant.")
 else:
    print(" belongs to 4th Quadrant.")
 i+=1
```

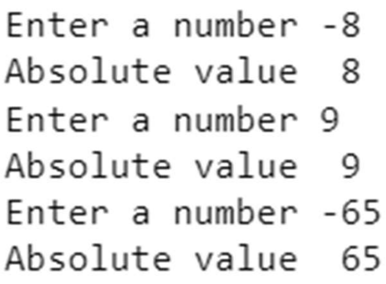


```
Enter X coordinate1
Enter Y coordinate2
( 1 , 2 ) belongs to 1st Quadrant.
Enter X coordinate-5
Enter Y coordinate-6
( -5 , -6 ) belongs to 3rd Quadrant.
Enter X coordinate3
Enter Y coordinate-6
( 3 , -6 ) belongs to 4th Quadrant.
Enter X coordinate-6
Enter Y coordinate8
( -6 , 8 ) belongs to 2nd Quadrant.
```

26. Write a program to get the absolute value of a number without using the abs() function.**(university question)**

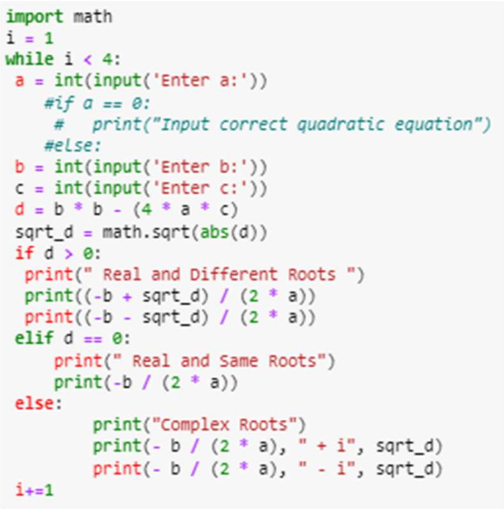


```
i = 1
while i < 4:
 x=int(input("Enter a number "))
 if x >= 0:
    print("Absolute value ",x)
 else:
    print("Absolute value ",-x)
 i+=1
```

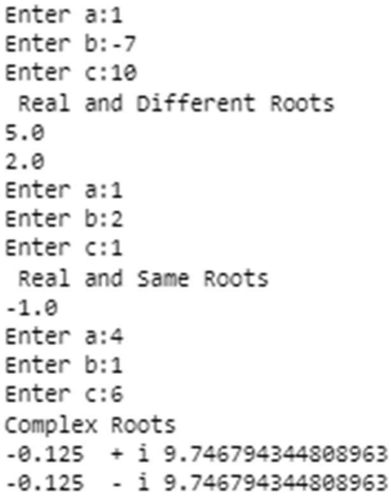


```
Enter a number -8
Absolute value  8
Enter a number 9
Absolute value  9
Enter a number -65
Absolute value  65
```

27. Find the roots of a quadratic equation

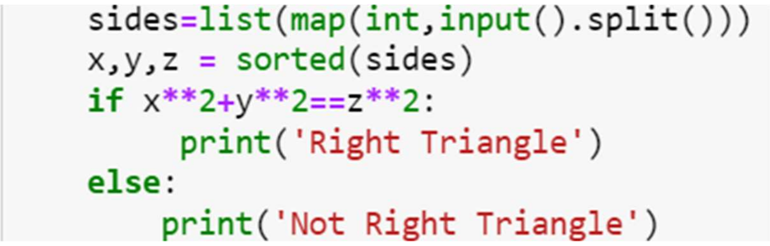


```
import math
i = 1
while i < 4:
 a = int(input('Enter a:'))
    #if a == 0:
     #   print("Input correct quadratic equation")
    #else:
 b = int(input('Enter b:'))
 c = int(input('Enter c:'))
 d = b * b - (4 * a * c)
 sqrt_d = math.sqrt(abs(d))
 if d > 0:
  print(" Real and Different Roots ")
  print((-b + sqrt_d) / (2 * a))
  print((-b - sqrt_d) / (2 * a))
 elif d == 0:
     print(" Real and Same Roots")
     print(-b / (2 * a))
 else:
         print("Complex Roots")
         print(- b / (2 * a), " + i", sqrt_d)
         print(- b / (2 * a), " - i", sqrt_d)
 i+=1
```

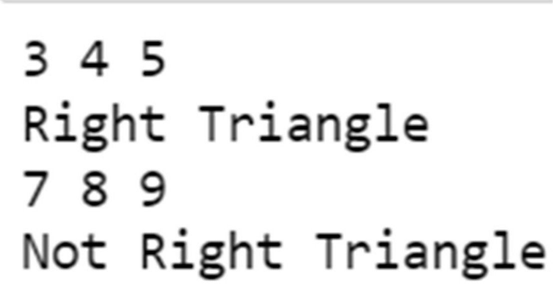


```
Enter a:1
Enter b:-7
Enter c:10
 Real and Different Roots
5.0
2.0
Enter a:1
Enter b:2
Enter c:1
 Real and Same Roots
-1.0
Enter a:4
Enter b:1
Enter c:6
Complex Roots
-0.125  + i 9.746794344808963
-0.125  - i 9.746794344808963
```

28. Write a program that accepts the length of three sides of a triangle as input and determine whether or not the triangle is a right triangle.**(university question)**



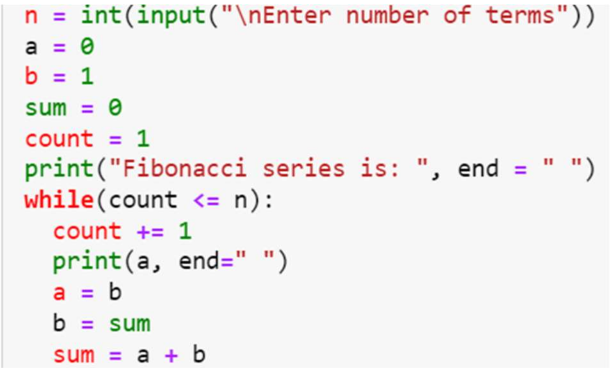
```
sides=list(map(int,input().split()))
x,y,z = sorted(sides)
if x**2+y**2==z**2:
      print('Right Triangle')
else:
     print('Not Right Triangle')
```



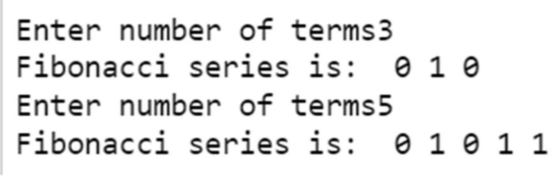
```
3 4 5
Right Triangle
7 8 9
Not Right Triangle
```

Using while/for

29. Generate the Fibonacci series 0 1 1 2 3 5 8…..n (use while..read n)

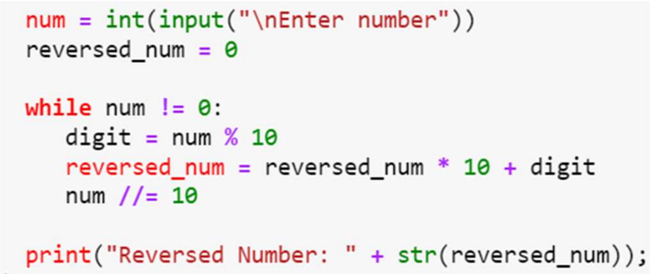


```
n = int(input("\nEnter number of terms"))
a = 0
b = 1
sum = 0
count = 1
print("Fibonacci series is: ", end = " ")
while(count <= n):
  count += 1
  print(a, end=" ")
  a = b
  b = sum
  sum = a + b
```

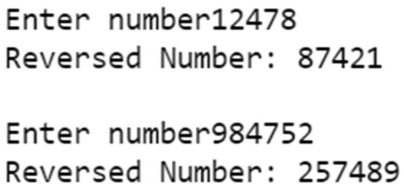


```
Enter number of terms3
Fibonacci series is:  0 1 0
Enter number of terms5
Fibonacci series is:  0 1 0 1 1
```

30. Reverse a Number (i/p: 123 o/P 321)

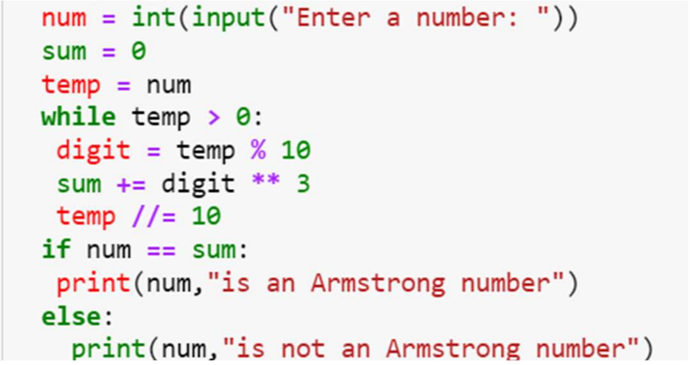


```
num = int(input("\nEnter number"))
reversed_num = 0

while num != 0:
   digit = num % 10
   reversed_num = reversed_num * 10 + digit
   num //= 10

print("Reversed Number: " + str(reversed_num));
```

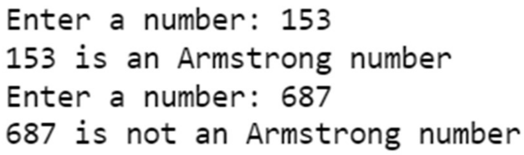


```
Enter number12478
Reversed Number: 87421

Enter number984752
Reversed Number: 257489
```

31. Check whether the given 3 digit number is **Armstrong Number**. Eg:153 (1+125+27=153) sum of cubes of digits=number.

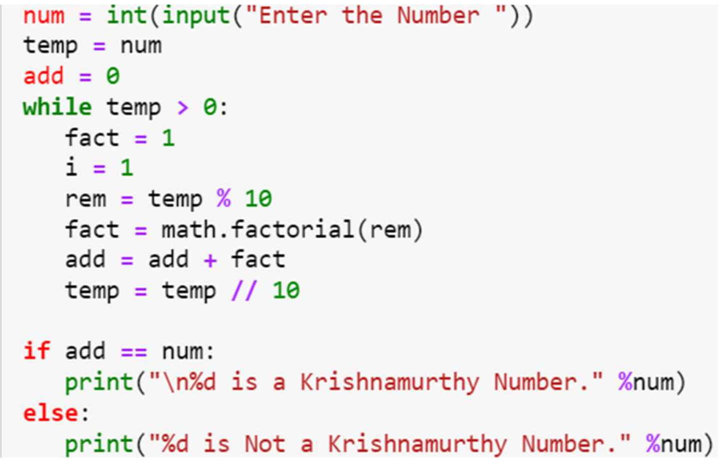


```
num = int(input("Enter a number: "))
sum = 0
temp = num
while temp > 0:
 digit = temp % 10
 sum += digit ** 3
 temp //= 10
if num == sum:
 print(num,"is an Armstrong number")
else:
  print(num,"is not an Armstrong number")
```

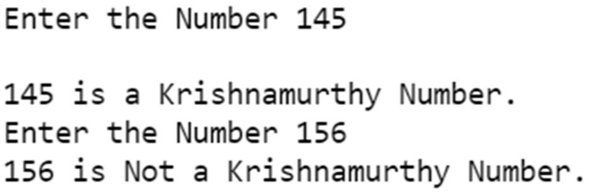


```
Enter a number: 153
153 is an Armstrong number
Enter a number: 687
687 is not an Armstrong number
```

32. Check whether the given number is a **Krishnamurthy number**

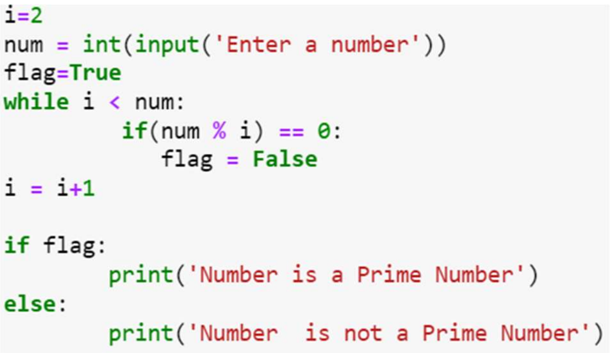


```
num = int(input("Enter the Number "))
temp = num
add = 0
while temp > 0:
   fact = 1
   i = 1
   rem = temp % 10
   fact = math.factorial(rem)
   add = add + fact
   temp = temp // 10

if add == num:
   print("\n%d is a Krishnamurthy Number." %num)
else:
   print("%d is Not a Krishnamurthy Number." %num)
```

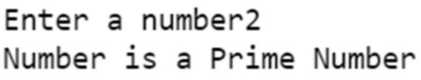


```
Enter the Number 145

145 is a Krishnamurthy Number.
Enter the Number 156
156 is Not a Krishnamurthy Number.
```

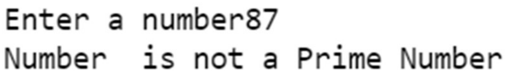
33. Check whether the given number is **Prime or not**



```
i=2
num = int(input('Enter a number'))
flag=True
while i < num:
        if(num % i) == 0:
           flag = False
i = i+1

if flag:
        print('Number is a Prime Number')
else:
        print('Number  is not a Prime Number')
```



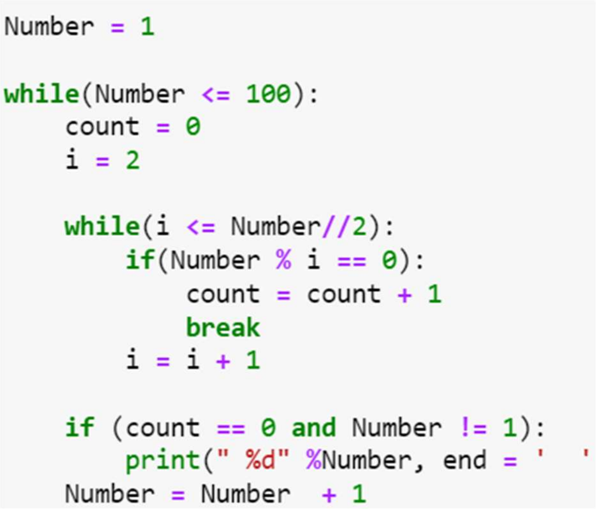
```
Enter a number2
Number is a Prime Number
```



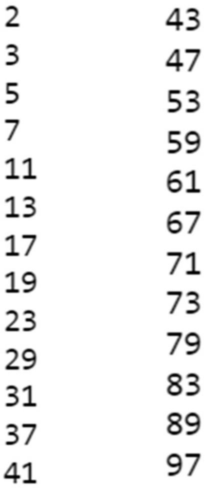
```
Enter a number87
Number  is not a Prime Number
```

Nested Loops(for/while)

34. Print all prime numbers less than 100.

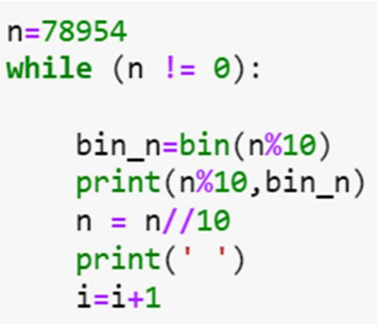


```
Number = 1

while(Number <= 100):
    count = 0
    i = 2

    while(i <= Number//2):
        if(Number % i == 0):
            count = count + 1
            break
        i = i + 1

    if (count == 0 and Number != 1):
        print(" %d" %Number, end = '  '
    Number = Number  + 1
```

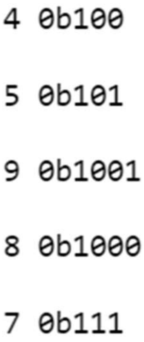


```
2          43
3          47
5          53
7          59
11         61
13         67
17         71
19         73
23         79
29         83
31         89
37         97
41
```

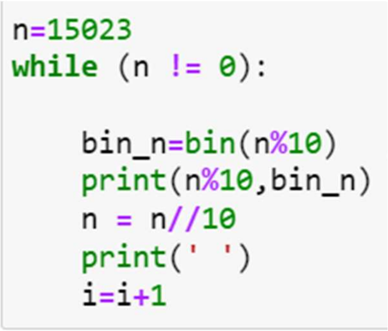
35. Print the binary equivalent of each digit of the given number.



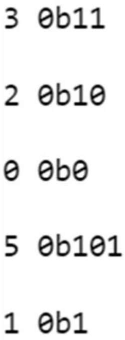
```
n=78954
while (n != 0):

    bin_n=bin(n%10)
    print(n%10,bin_n)
    n = n//10
    print(' ')
    i=i+1
```



```
4 0b100

5 0b101

9 0b1001

8 0b1000

7 0b111
```



```
n=15023
while (n != 0):

    bin_n=bin(n%10)
    print(n%10,bin_n)
    n = n//10
    print(' ')
    i=i+1
```



```
3 0b11

2 0b10

0 0b0

5 0b101

1 0b1
```