

MODULE 2 :conditional and loops

Lecture -1

Today's Target

- Conditional statement
- AKTU PYQs

By **PRAGYA RAJVANSHI**

B.Tech, M.Tech(C.S.E)

- A conditional statement as the name suggests itself, is used to handle conditions in your program. These statements guide the program while making decisions based on the conditions encountered by the program., we will explore three conditional statements in Python: if statement, if-else statements, if-elif-else ladder.

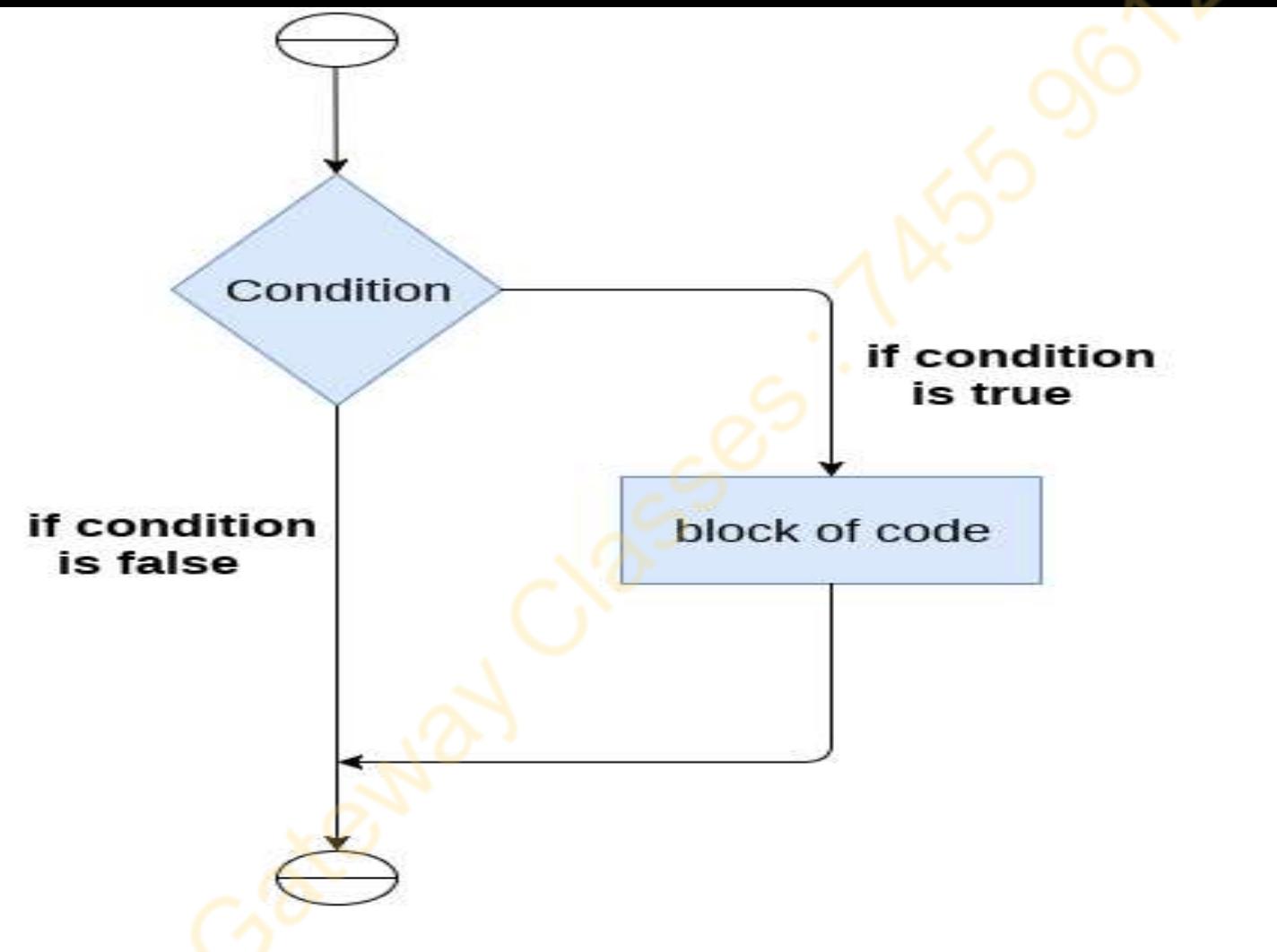
- For the ease of programming and to achieve simplicity, python doesn't allow the use of parentheses for the block level code. In Python, indentation is used to declare a block. If two statements are at the same indentation level, then they are the part of the same block.
- Generally, four spaces are given to indent the statements which are a typical amount of indentation in python
- Indentation is the most used part of the python language since it declares the block of code. All the statements of one block are intended at the same level indentation. We will see how the actual indentation takes place in decision making and other stuff in python.

- The if statement is used to test a particular condition and if the condition is true, it executes a block of code known as if-block. The condition of if statement can be any valid logical expression which can be either evaluated to true or false.
- Syntax

if expression:

statement

The if statement(flowchart)



Example

- num = int(input("enter the number:"))
- if num%2 == 0:
- print("The Given number is an even number")
- Output
- Enter the number 8
- The given number is an even number

Example

```
> num = int(input("enter the number:"))

> if num<12:
>     print("hello")

> if num>3 and num<12:
>     print("hi")
>     print("how are you?")

> if num>4 and num<14:
>     print("bye")

> if num>12:
>     print("jjj")
```

Example

- output
- enter the number:5
- hello
- hi
- how are you?
- bye

Write a program to find out the largest number among three

```
a = int (input("Enter a: "))

b = int (input("Enter b: "))

c = int (input("Enter c: "))

if a>b and a>c:

    print ("From the above three numbers given a is largest")

if b>a and b>c:

    print ("From the above three numbers given b is largest")

if c>a and c>b:

    print ("From the above three numbers given c is largest")
```

Write a program to find out the largest number among three

```
a = int (input("Enter a: "))

b = int (input("Enter b: "))

c = int (input("Enter c: "))

if a>b and a>c:

    print (a,"is the largest number")

if b>a and b>c:

    print (b," is the largest number" )

if c>a and c>b:

    print (c, "is the largest number")
```

Write a program to find out the largest number among three

output

Enter a: 6

Enter b: 7

Enter c: 5

7 is the largest number

The if-else statement provides an else block combined with the if statement which is executed in the false case of the condition.

if the condition is true, then the if-block is executed. Otherwise, the else-block is executed.

Syntax

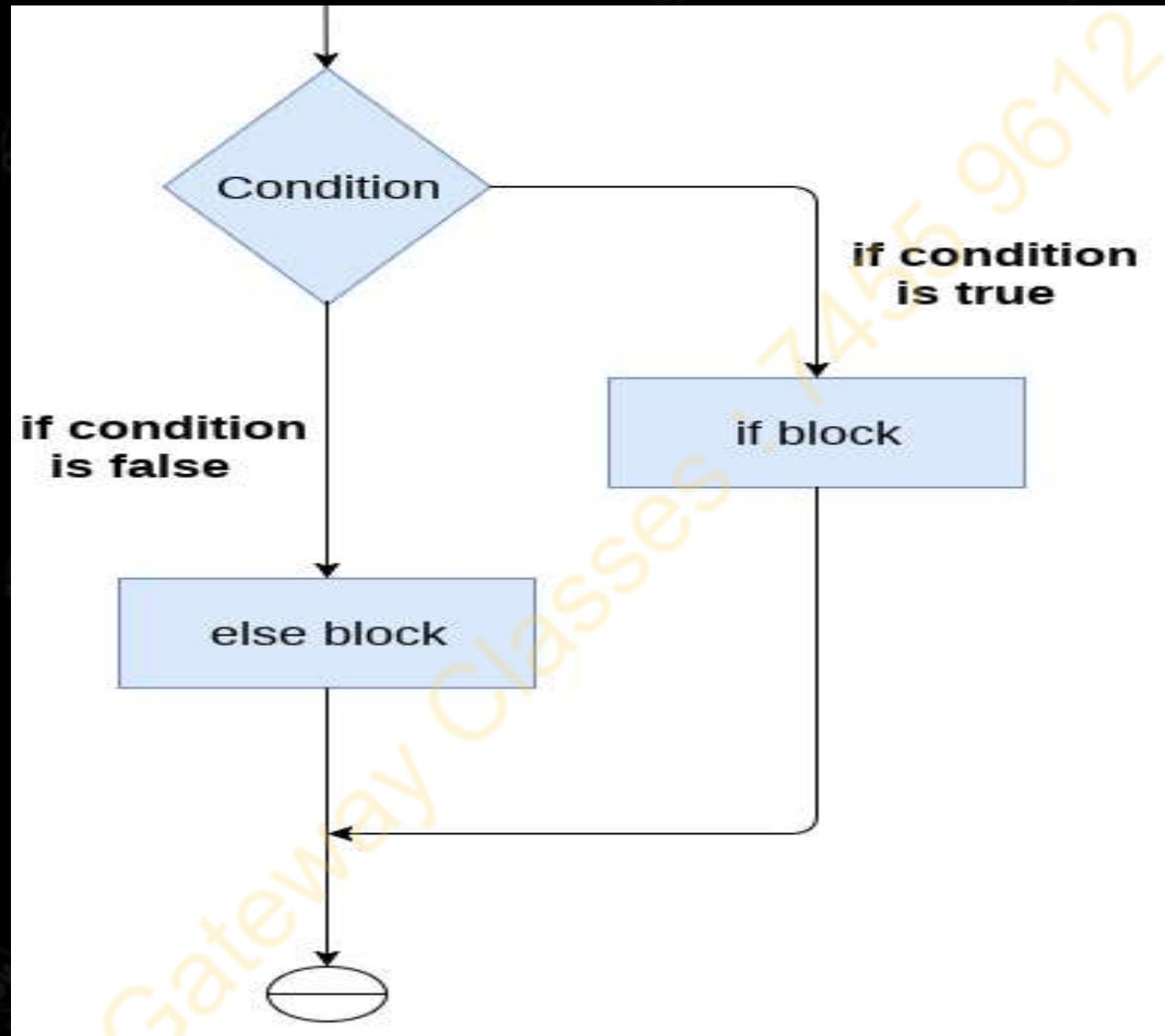
if condition:

#block of statements

else:

#another block of statements (else-block)

If-else statement



Write a program to check whether you are eligible to vote or not

```
age = int (input("Enter your age: "))

if age>=18:
    print("You are eligible to vote !!")

else:
    print("Sorry! you have to wait !!")
```

Output

Enter your age: 27

You are eligible to vote !!

Write a program to check whether number is even or not

```
num = int(input("enter the number:"))

if num%2 == 0:

    print("The Given number is an even number")

else:

    print("The Given Number is an odd number")
```

Output

enter the number:6

The Given number is an even number

Write a program to check whether number is divisible by 3 or not

```
num = int(input("enter the number:"))

if num%3 == 0:

    print("The Given number is a divisible by 3 ")

else:

    print("The Given Number is not divisible by 3")
```

Output

enter the number:6

The Given number is divisible by 3

MODULE 2 :conditional and loops

Lecture -2

Today's Target

- Control statement

By PRAGYA RAJVANSHI

B.Tech, M.Tech(C.S.E)

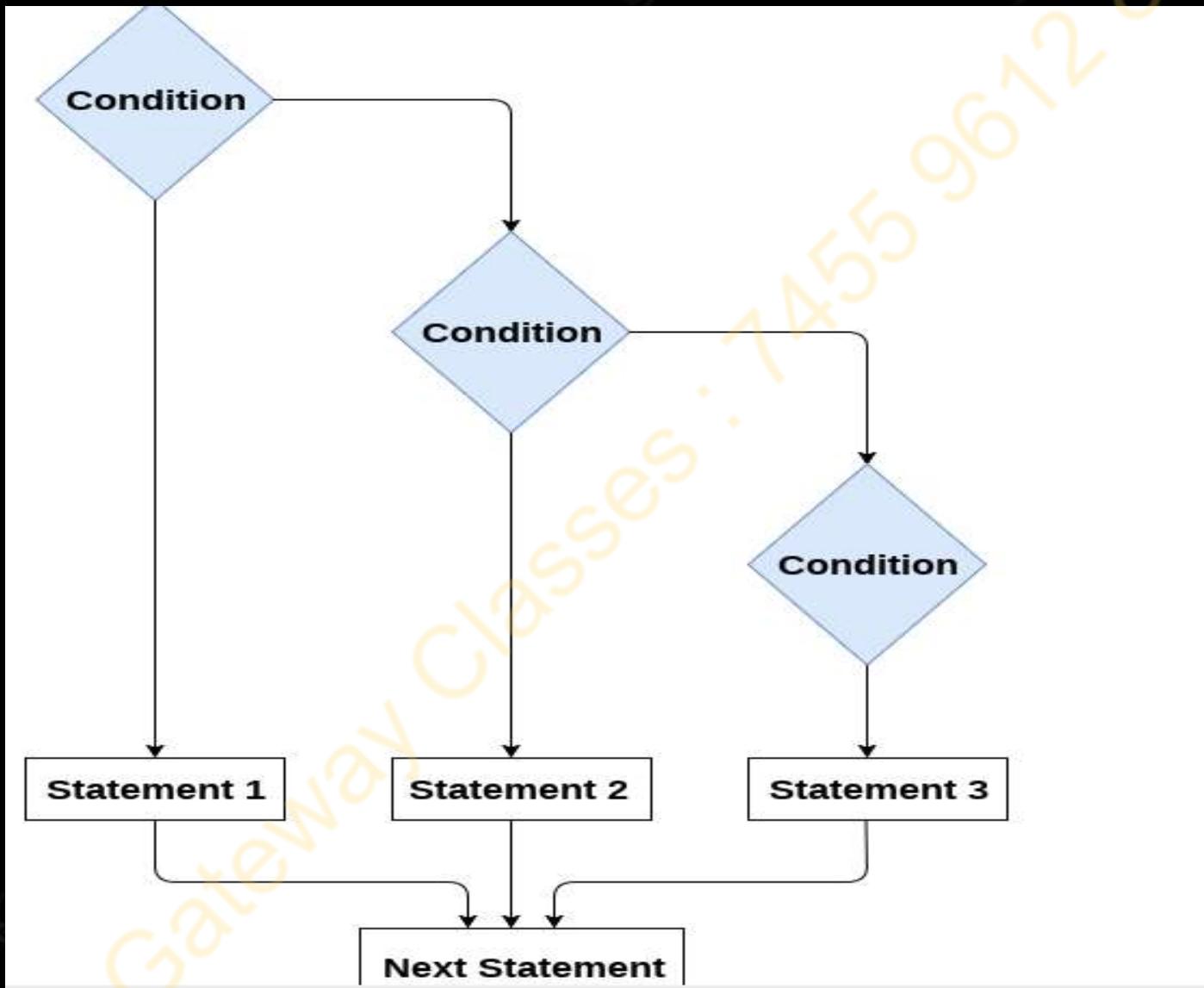
The elif statement

- The elif statement enables us to check multiple conditions and execute the specific block of statements depending upon the true condition among them. We can have any number of elif statements in our program depending upon our need. However, using elif is optional.
- The elif statement works like an if-else-if ladder statement in C. It must be succeeded by an if statement.
- The syntax of the elif statement is given below.

The elif statement(syntax)

```
if expression 1:  
    # block of statements  
  
elif expression 2:  
    # block of statements  
  
elif expression 3:  
    # block of statements  
  
else:  
    # block of statements
```

The elif statement(flowchart)



Example1

```
number = int(input("Enter the number?"))

if number==10:
    print("The given number is equals to 10")

elif number==50:
    print("The given number is equal to 50")

elif number==100:
    print("The given number is equal to 100")

else:
    print("The given number is not equal to 10, 50 or 100")
```

Example

output

Enter the number?6

The given number is not equal to 10, 50 or 100

Example2

```
marks = int(input("Enter the marks? "))

if marks > 85 and marks <= 100:
    print("Congrats ! you scored grade A ...")

elif marks > 60 and marks <= 85:
    print("You scored grade B + ...")

elif marks > 40 and marks <= 60:
    print("You scored grade B ...")

elif (marks > 30 and marks <= 40):
    print("You scored grade C ...")

else:
    print("Sorry you are fail ?")
```

Example2

output

Enter the marks? 70

You scored grade B + ...

Write a program to Convert month name to a number of days

```
print("List of months: January, February, March, April, May, June, July, August, September, October,  
November, December")  
  
month_name = input("Input the name of Month: ")  
  
if month_name == "February":  
    print("No. of days: 28/29 days")  
  
elif month_name in ("April", "June", "September", "November"):  
    print("No. of days: 30 days")  
  
elif month_name in ("January", "March", "May", "July", "August", "October", "December"):  
    print("No. of days: 31 day")  
  
else:  
    print("Wrong month name")
```

Write a program to Convert month name to a number of days

output

List of months: January, February, March, April, May, June, July, August, September, October, November, December

Input the name of Month: April

No. of days: 30 days

MODULE 2 :conditional and loops

Lecture -3

Today's Target

- Triangle side and angle program
- Character, vowel, consonants program
- Largest number among three
- AKTU PYQs

By PRAGYA RAJVANSI

B.Tech, M.Tech(C.S.E)

Write a program to input a character and check whether a character is vowel and consonants

```
ch=input("enter the character\n")
if ch in("a","e","i","o","u","A","E","I","O","U"):
    print("vowels")
else:
    print("consonants")
```

Write a program to input a character and check whether a character is vowel and consonants

output

```
input  
enter the character  
t  
consonants
```

Write a program to check whether a character is vowel and consonants or digit and special symbol

```
ch=input("enter the character\n")
if ch in("a","e","i","o","u","A","E","O","U"):
    print("vowels")
elif ch>="A" and ch<="Z" or ch>="a" and ch<="z":
    print("consonants")
elif ch>="0" and ch<="9":
    print("digit")
else:
    print("special symbol")
```

Write a program to check whether a character is vowel and consonants or digit and special symbol

output



enter the character

6

digit

Write a program to check whether a character is alphabet (uppercase and lower case and non-alphabet symbol)

```
1 ch= input("enter the character\n")
2 if ch>="A" and ch<="Z":
3     print("upper case alphabet")
4 elif ch>="a" and ch<="z":
5     print("lower case alphabet")
6 else:
7     print("non-alphabet symbol ")
```

Write a program to check whether a character is alphabet ,non-alphabet and non-alphabet symbol

```
enter the character
A
upper case alphabet
...Program finished with exit code 0
Press ENTER to exit console.
```

Write a program to enter the angle and check which type of triangle is

```
1 a=int(input("enter angle"))
2 b=int(input("enter angle"))
3 c=int(input("enter angle"))
4 if a+b+c==180 and a>0 and b>0 and c>0:
5     print("valid")
6     if a==b and b==c and c==a:
7         print("equi")
8     if a==b or b==c or c==a:
9         print("iso")
10    if a!=b and b!=c and c!=a:
11        print("scalene")
12    if a==90 or b==90 or c==90:
13        print("right")
14 else:
15     print("invalid")
```

Write a program to enter the angle and check which type of triangle is

```
input
▼ ▶ ↵ ↺
enter angle50
enter angle100
enter angle30
valid
scalene
```

...Program finished with exit code 0
Press ENTER to exit console. □

Write a program to enter the side and check which type of triangle is

```
1 a=int(input("enter the first sides of triangle\n"))
2 b= int(input("enter the second side of triangle\n"))
3 c=int(input("enter the third side of triangle\n"))
4 if a+b>c and b+c>a and c+a>b:
5     print("valid triangle")
6     if a==b and b==c and c==a:
7         print("equilateral triangle")
8     if a==b or b==c or c==a:
9         print("iscosceles triangle")
10    if a!=b and b!=a and c!=a:
11        print("scalene triangle")
12    if a*a+b*b==c*c or b*b+c*c==a*a or a*a+c*c==b*b:
13        print("right angled triangle")
14 else:
15     print("invalid triangle")
16
17
```

Write a program to enter the side and check which type of triangle is

```
enter the first sides of triangle
3
enter the second side of triangle
3
enter the third side of triangle
3
valid triangle
equilateral triangle
iscosceles triangle

...Program finished with exit code 0
Press ENTER to exit console.
```

Write a program to find out the largest number among three

```
1 a=int(input("enter the first value\n"))
2 b=int(input("enter the second value\n"))
3 c=int(input("enter the third value\n"))
4 if a>b and a>c:
5     print(a,"is the largest")
6 elif b>a and b>c:
7     print(b , "is the largest")
8 else:
9     print(c, " is the largest")
10
```

Write a program to find out the largest number among three

```
enter the first value
4
enter the second value
5
enter the third value
6
6 is the largest

...Program finished with exit code 0
Press ENTER to exit console.
```

MODULE 2 :conditional and loops

Lecture -4

Today's Target

- Leap year
- Largest number among three using nested –if
- Quadratic equation
- Calculator using elif

By PRAGYA RAJVANSI

B.Tech, M.Tech(C.S.E)

Write a program to check which is largest among three using nested if -else

```
1 a=int(input("enter the value of a\n"))
2 b=int(input("enter the value of b\n"))
3 c=int(input("enter the value of c\n"))
4 if a>b:
5     if a>c:
6         print(a, "is the largest")
7     else:
8         print(c,"is the largest")
9
10 else:
11     if b>c:
12         print(b,"is the largest")
13     else:
14         print(c, "is the largest")
15
```

Write a program to check which is largest among three using nested if -else

output

```
enter the value of a  
5  
enter the value of b  
6  
enter the value of c  
7  
7 is the largest
```

Write a program to check whether year is leap year or not

```
x=int(input("enter the year\n"))
if x%400==0:
    print(x," is leap year")
elif x%100==0:
    print(x,"is not a leap year")
elif x%4==0:
    print(x,"is leap year")
else:
    print(x,"is not leap year")
```

Write a program to check whether year is leap year or not

output

```
enter the year
2016
2016 is leap year
...Program finished with exit code 0
Press ENTER to exit console.
```

Write a program to find out the roots of quadratic equation

```
import math
print("enter the coefficient of quadratic equation")
a=float(input("enter the x^2cofficient\n"))
b=float(input("enter the x coefficient\n"))
c=float(input("entr the x^0 coefficient"))
d=b*b-4*a*c
sqrt_val = math.sqrt(abs(d))
if d>0:
    print("roots are real and distinct")
    r1=(-b+sqrt_val)/(2*a)
    r2=(-b-sqrt_val)/(2*a)
    print("root 1=",r1)
    print("root2=",r2)
```

Write a program to find out the roots of quadratic equation

```
    print("roots are real")
elif d==0:
    print("roots are equal")
    r1=r2=-b/(2*a)
    print("root 1=",r1)
    print("root2=",r2)
else:
    print("roots are imaginary")
    real=-b/(2*a)
    imaginary = sqrt_val / (2 * a)
    print(real, "+ i", imaginary)
    print(real, "- i", imaginary)
```

write a program to find the root of quadratic equation

```
enter the coefficient of quadratic equation
enter the x^2 coefficient
5
enter the x coefficient
6
enter the x^0 coefficient
roots are imaginary
-0.6 + i 1.1135528725660042
-0.6 - i 1.1135528725660042
```

Write a program to make calculator using elif statement

```
x=float(input("enter the first number\n"))
y=float(input("enter the second number\n"))
z=input("enter the operataor{+, -, *, /, //}\n")
if z=="+":
    w=x+y
    print(w, "the sum of two number")
elif z=="-":
    w=x-y
    print(w, "subtraction of x-y number")
elif z=="*":
    w=x*y
    print(w,"multiplication of two number")
elif z=="/":
    w=x/y
    print(w,"divison of x/y")
```

Write a program to make calculator using elif statement

```
elif z=="//":  
    w=x//y  
    print(w,"floor division of x//y")  
else:  
    print("enter the wrong choice")
```

Write a program to make calculator using elif statement

```
enter the first number
5
enter the second number
6
enter the operataor{+, -, *, /, //}
+
11.0 the sum of two number
PS C:\Users\gwcla> 
```

MODULE 2 :conditional and loops

Lecture -5

Today's Target

- Match case program
- Days in a week
- Calculator
- Vowel or consonant
- Month in a year

By PRAGYA RAJVANSI

B.Tech, M.Tech(C.S.E)

For developers coming from languages like C/C++ or Java know that there was a conditional statement known as **Switch Case**. This **Match-Case** is the Switch Case of Python which was introduced in Python 3.10. Here we have to first pass a parameter then try to check with which case the parameter is getting satisfied. If we find a match we will do something and if there is no match at all we will do something else

```
parameter = "Geeksforgeeks"
```

```
match parameter:
```

```
    case first :
```

```
        do_something(first)
```

```
    case second :
```

```
        do_something(second)
```

```
case third :  
    do_something(third)  
.....  
.....  
case n :  
    do_something(n)  
case _ :  
    nothing_matched_function()
```

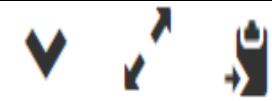
P Write to print the day in a week using match case

```
print(" enter 1-monday, 2-tuesday, 3-wednesday,4-thrusday")
print ("5-friday, 6-staurday, 7- sunday")
x=input("enter the number")
match x:
    case "1":
        print("monday")
    case "2":
        print("tuesday")
    case "3":
        print("wednesday")
    case "4":
        print("thrusday")
    case "5":
        print("friday")
    case "6":
        print("staurday")
```

Write to print the day in a week using match case

```
case "7":  
    print("sunday")  
case _:  
    print("wrong choice enter again!")
```

Write a program to print the number of days in a week



input

```
enter 1-monday, 2-tuesday, 3-wednesday, 4-thrusday  
5-friday, 6-staurday, 7- sunday  
enter the number6  
staurday
```

Write a program to check whether is a vowel or consonants

```
x=input("enter the string of length 1 \n")
match x:
    case "A":
        print("vowel")
    case "E":
        print("VOWEL")
    case "I":
        print("vowel")
    case "O":
        print("vowel")
    case "a":
        print("vowel")
    case "i":
        print("vowel")
    case "o":
        print("vowel")
```

Write a program to check whether is a vowel or consonants

```
case "u":  
    print("vowel")  
case "e":  
    print("vowel")  
case _:  
    print("consonants")
```

Write a program to check whether is a vowel or consonants

```
enter the string of length 1
t
consonants
...Program finished with exit code 0
Press ENTER to exit console.
```

Write to print the number of month in a year

```
print("1-jan,2-feb, 3-march , 4-april, 5-may, 6-june, 7-july")
print("8-aug , 9-sep, 10-oct,11-nov,12-dec")
x=input("enter the number for displaying the particular year")
match x:
    case "1":
        print("jan")
    case "2":
        print("feb")
    case "3":
        print("march")
    case "4":
        print("april")
    case "5":
        print("may")
    case "6":
        print("june")
```

Write to print the number of month in a year

```
case "7":  
    print("july")  
case "8":  
    print("aug")  
case "9":  
    print("sep")  
case "10":  
    print("oct")  
case "11":  
    print("nov")  
case "12":  
    print("dec")  
case _:  
    print("invalid choice! enter again")
```

Write to print the number of month in a year

input

1-jan, 2-feb, 3-march , 4-april, 5-may, 6-june, 7-july

8-aug , 9-sep, 10-oct, 11-nov, 12-dec

enter the number for displaying the particular year

6
june

...Program finished with exit code 0

Press ENTER to exit console. []

Write a program to make calculator using match case

```
x=float(input("enter the first number\n"))
y=float(input("enter the second number\n"))
z=input("enter the operator{+,*,/,-,//}\n")
match z:
    case "+":
        w=x+y
        print(w,"the sum of two number")
    case "-":
        w=x-y
        print(w,"subtraction of x-y number")
    case "*":
        w=x*y
        print(w,"multiplication of two number")
```

Write a program to make calculator using match case

```
case "/":  
    w=x/y  
    print(w,"division of x/y")  
case "//":  
    w=x//y  
    print(w,"floor division of x//y")  
case _:  
    print("enter the wrong choice")
```

Write a program to make calculator using match case

```
enter the first number
8
enter the second number
3
enter the operataor{+, *, /, -, //}
/
2.666666666666666 division of x/y

...Program finished with exit code 0
Press ENTER to exit console.[]
```

MODULE 2 :conditional and loops

Lecture -6

Today's Target

- While loop
- Loops program

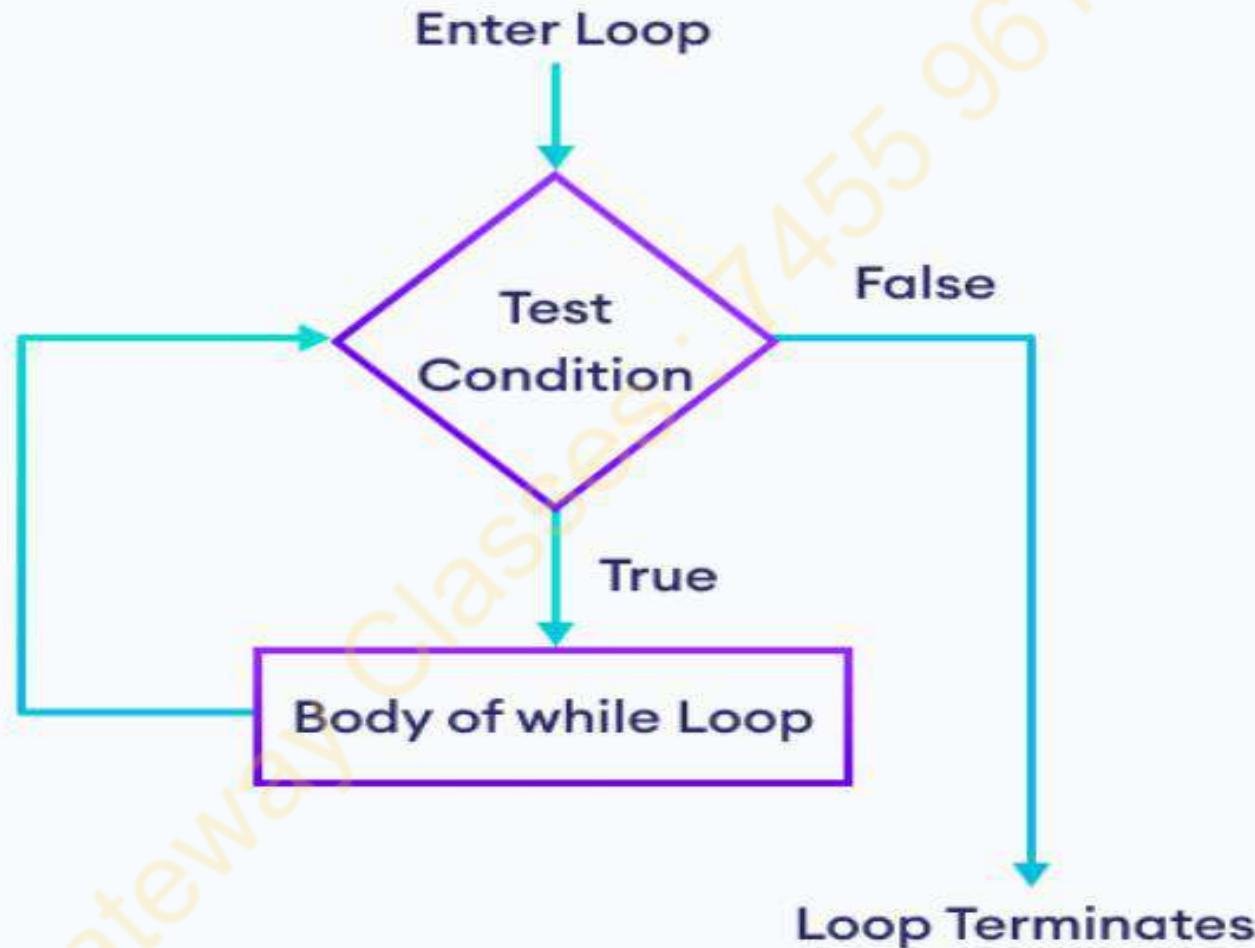
By PRAGYA RAJVANSI

B.Tech, M.Tech(C.S.E)

- A loop is a programming structure that repeats a sequence of instructions until a specific condition is satisfied
- A loop statement allows us to execute a statement or group of statement multiple times
- Python programming language provides following types of loops to handle looping requirement:
- While(entry controlled loop)
- For
- nested

- In python, a while loop is used to execute a block of statements repeatedly until a given condition is satisfied. And when the condition becomes false, the line immediately after the loop in the program is executed.
- Syntax
- While expression:
statement
- All the statements indented by the same number of character spaces after a programming construct are considered to be part of a single block of code. Python uses indentation as its method of grouping statements.

while loop(flowchart)



while Loop

```
1 x=int(input("enter the number\n"))
2 while x<11:
3     print("hello")
4     x=x+1
```

```
enter the number
6
hello
hello
hello
hello
hello
...
Program finished with exit code 0
Press ENTER to exit console.
```

write a program to print first n natural number

```
1 x=int(input("enter the number\n"))
2 i=1
3 while(i<=x):
4     print(i)
5     i=i+1
```

```
enter the number
6
1
2
3
4
5
6
```

write a program to print the sum of first n natural number

```
1 x=int(input("enter the number\n"))
2 sum=0
3 i=1
4 while(i<=x):
5     print(i)
6     sum=sum+i
7     i=i+1
8 print("sum=",sum)
```

```
enter the number
4
1
2
3
4
sum= 10
```

write a program to reverse a n digit number

```
1 n=int(input("enter n digit number\n"))
2 rev=0
3 while n>0:
4     d=n%10
5     rev=rev*10+d
6     n=n//10
7 print("reverse",rev)
```

```
input
enter n digit number
5678
reverse 8765
```

write a program to count the number of digit in a n digit number

```
num=int(input("enter the number"))
count = 0
while num >0 :
    num=num//10
    count= count+ 1
print("Number of digits: ", count)
```

write a program to count the number of digit in a n digit number

```
num = 123456  
print(len(str(num))).7455961282
```

write a program to check whether a number is palindrome or not

```
1 n=int(input("enter n digit number\n"))
2 num=n
3 rev=0
4 while n>0:
5     d=n%10
6     rev=rev*10+d
7     n=n//10
8 print("reverse",rev)
9 if rev== num:
10    print("palindrome")
11 else:
12    print("not palindrome")
```

input
enter n digit number
171
reverse 171
palindrome

write a program to check whether a number is Armstrong or not

```
1 n=int(input("enter n digit number\n"))
2 num=n
3 sum=0
4 while n>0:
5     d=n%10
6     sum=sum+(d*d*d)
7     n=n//10
8 if sum== num:
9     print("Armstrong")
10 else:
11     print("not armstrong")
```

The screenshot shows a terminal window with the following content:

- Terminal title: input
- Input prompt: enter n digit number
- User input: 153
- Output: Armstrong

MODULE 2 :conditional and loops

Lecture -7

Today's Target

- Flowchart and algorithm

By PRAGYA RAJVANSHI

B.Tech, M.Tech(C.S.E)



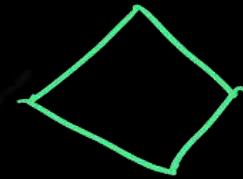
- oval Start | End



- || gm Print | Input



Rectangle Processing



Diamond Condition



arrow

Connect two or more
Symbol

Flowchart symbol

write a program to print first n natural number

```
1 x=int(input("enter the number\n"))
2 i=1
3 while(i<=x):
4     print(i)
5     i=i+1
```

```
enter the number
6
1
2
3
4
5
6
```

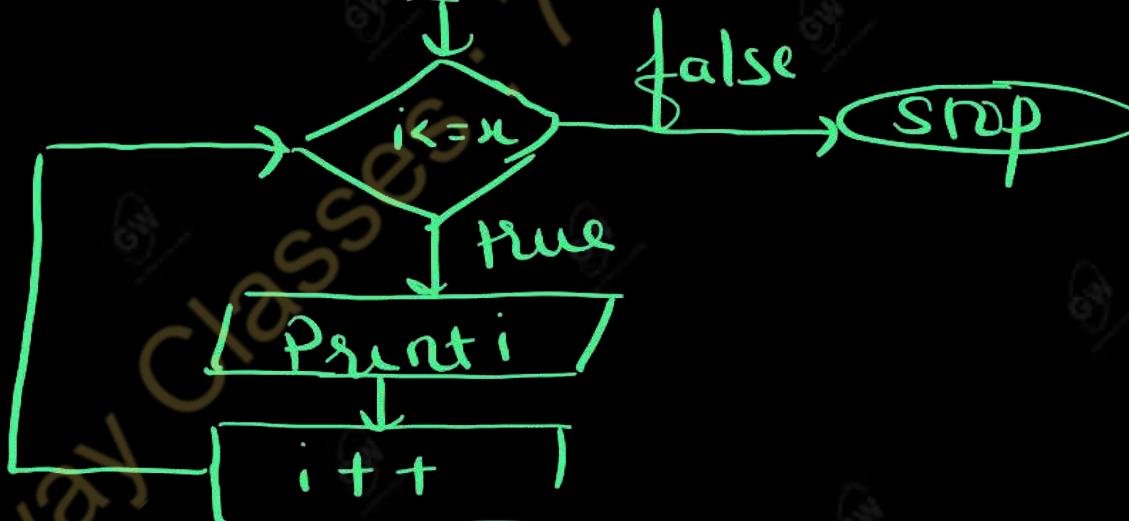
Algorithm

1. Start
2. Enter the number
3. $i = 1$
4. While $i \leq x$ true go to step 5 otherwise
 go to step 7
5. Print i
6. $i = i + 1$ then go to step 4
7. Stop

Flowchart

/ Input the number/

| i = 1 |



write a program to print the sum of first n natural number

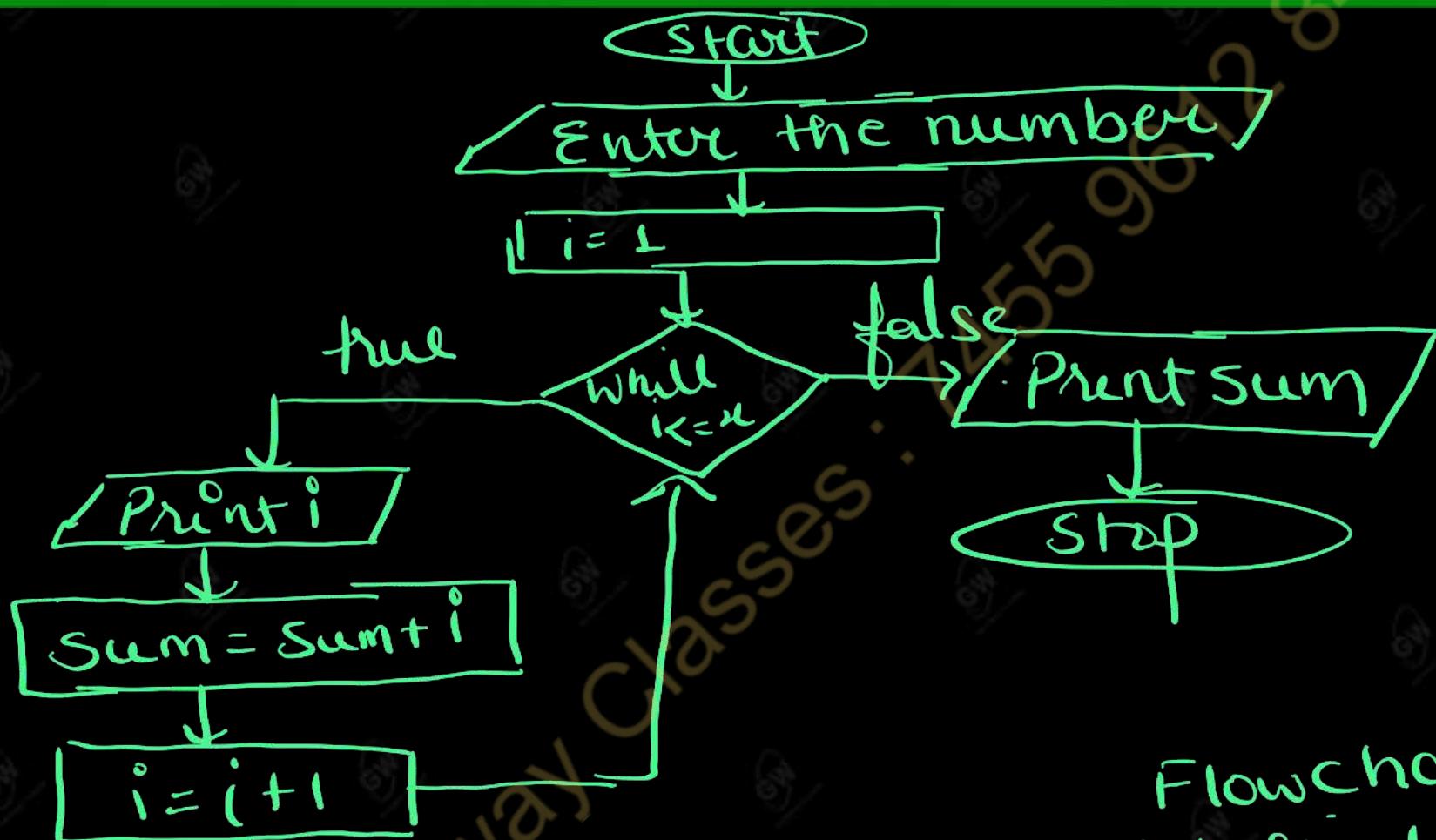
```
1 x=int(input("enter the number\n"))
2 sum=0
3 i=1
4 while(i<=x):
5     print(i)
6     sum=sum+i
7     i=i+1
8 print("sum=",sum)
```

```
enter the number
4
1
2
3
4
sum= 10
```

Algorithm

1. Start
2. Enter the number
3. $i = 1$
4. While $i \leq n$ true go to step 5 otherwise go to step 8
5. Print i
6. $sum = sum + i$
7. $i = i + 1$ then go to step 4
8. Print sum
9. Stop

Flowchart

NOTE

Flowchart is the pictorial representation of Algorithm

write a program to reverse a n digit number

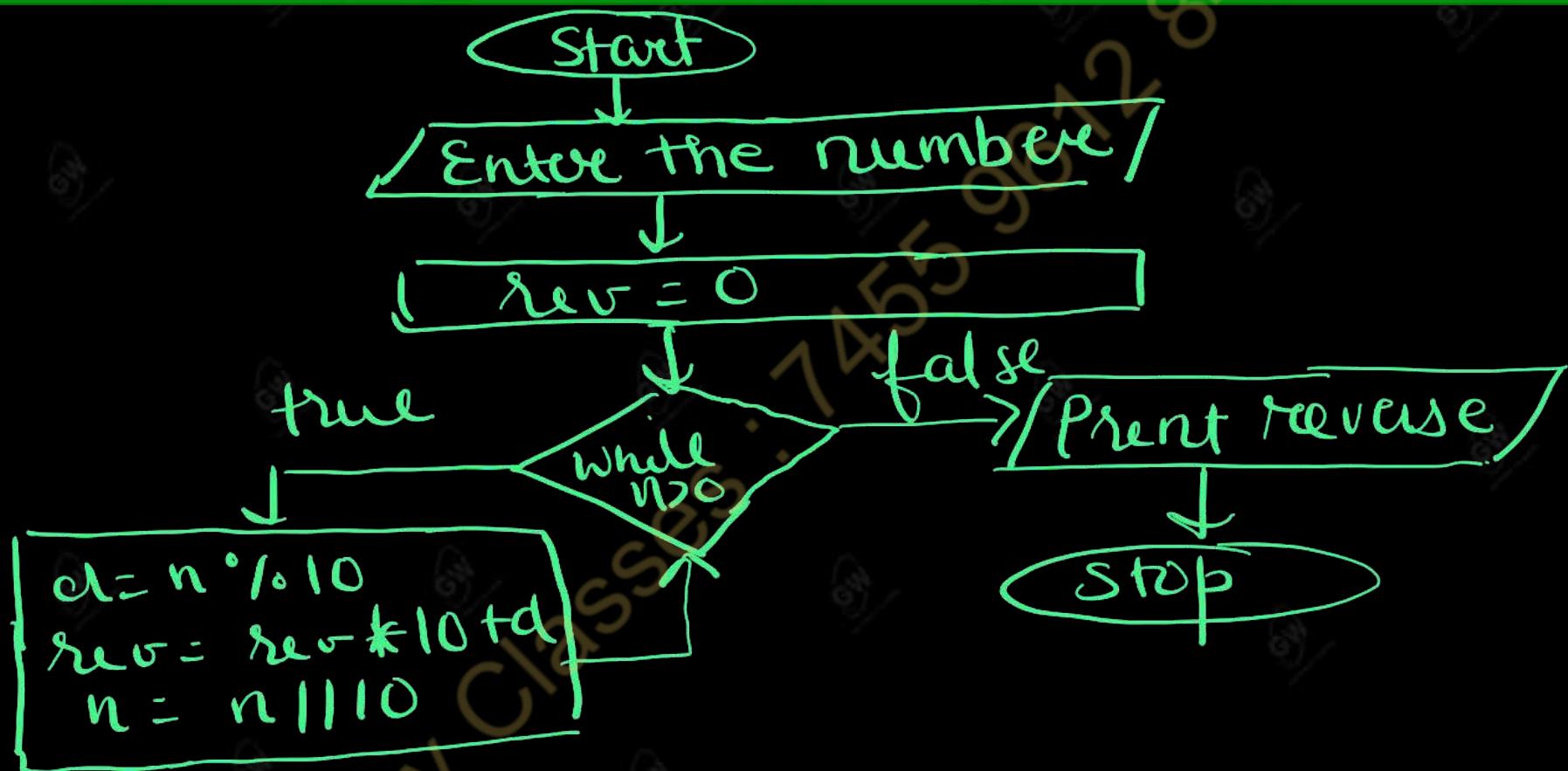
```
1 n=int(input("enter n digit number\n"))
2 rev=0
3 while n>0:
4     d=n%10
5     rev=rev*10+d
6     n=n//10
7 print("reverse",rev)
```

```
input
enter n digit number
5678
reverse 8765
```

write a program to reverse a n digit number (Algorithm)

1. Start
2. Enter a n digit number
3. $rev = 0$
4. while $n > 0$ true then go to step 5 otherwise go to step 6
5. $d = n \% 10$
 $rev = rev * 10 + d$
 $n = n / 10$ then go to step 4
6. Print reverse
7. Stop

write a program to reverse a n digit number (Flowchart)



write a program to check whether a number is palindrome or not

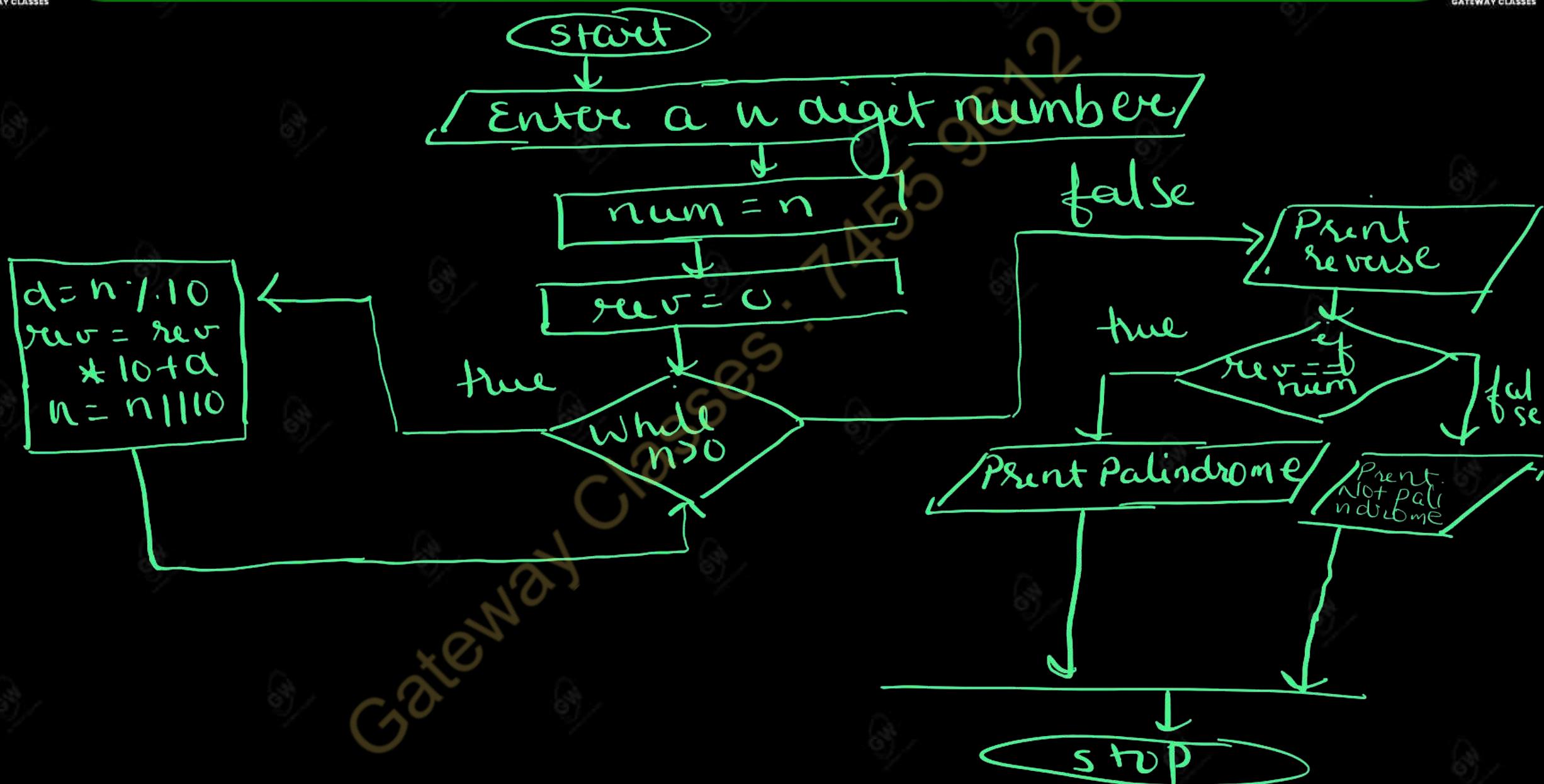
```
1 n=int(input("enter n digit number\n"))
2 num=n
3 rev=0
4 while n>0:
5     d=n%10
6     rev=rev*10+d
7     n=n//10
8 print("reverse",rev)
9 if rev== num:
10    print("palindrome")
11 else:
12    print("not palindrome")
```

```
input
enter n digit number
171
reverse 171
palindrome
```

Algorithm

1. Start
2. Enter a n digit number
3. num = n
4. rev = 0
5. while $n > 0$ true then go to step 6 otherwise step 7
6. $d = n \% 10$
 $rev = rev * 10 + d$
 $n = n / 10$ then go to step 5
7. Print rev
8. if (rev == num) true go to step 9 else step 10
9. Print Palindrome then go to step 11
10. Print not palindrome
11. Stop

Flowchart



write a program to check whether a number is Armstrong or not

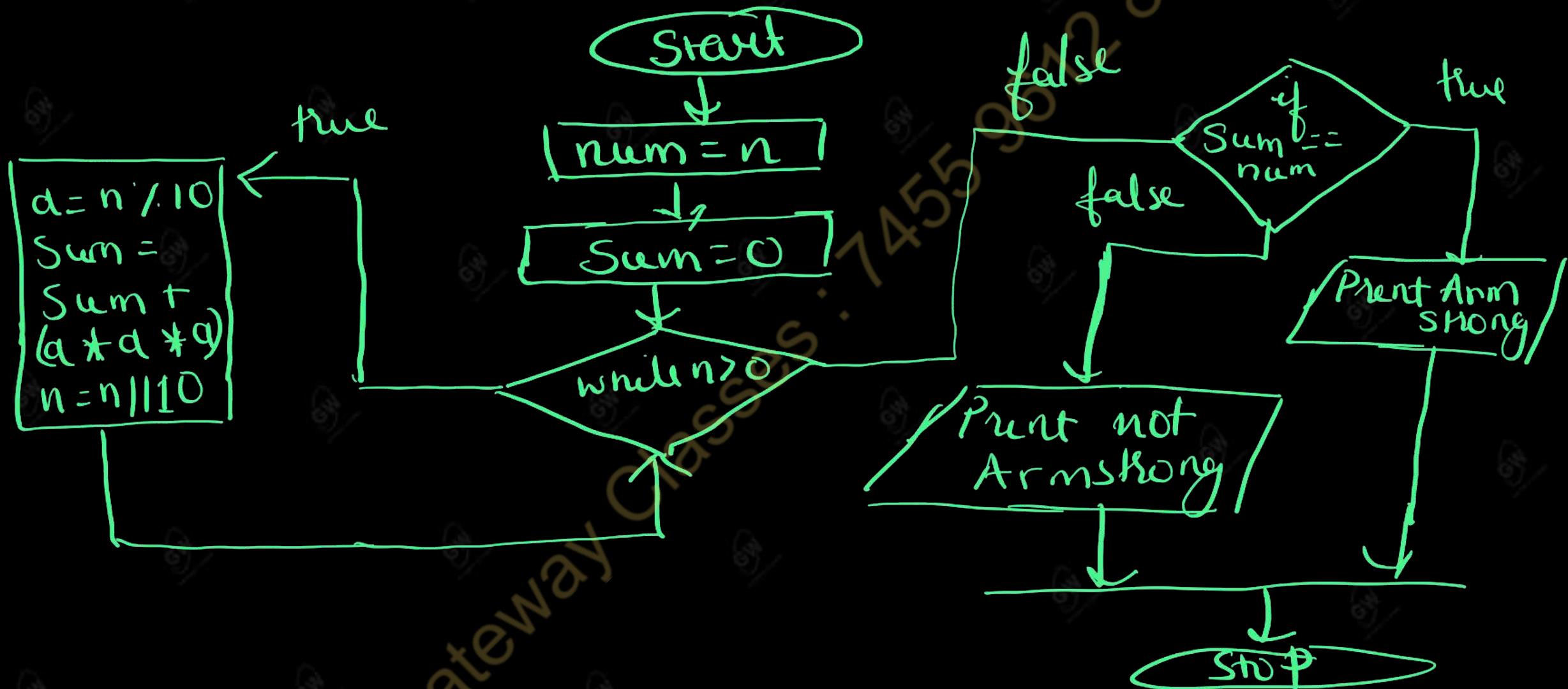
```
1 n=int(input("enter n digit number\n"))
2 num=n
3 sum=0
4 while n>0:
5     d=n%10
6     sum=sum+(d*d*d)
7     n=n//10
8 if sum== num:
9     print("Armstrong")
10 else:
11     print("not armstrong")
12
```

```
enter n digit number
153
Armstrong
```

Algorithm

1. Start
2. Enter a n digit number
3. num = n
4. rev = 0
5. while $n > 0$ True then goto step 6 otherwise goto step 7
6. $d = n \% 10$
Sum = Sum + (d * d * d)
 $n = n / 10$ then goto step 5
7. if Sum == num true then goto step 8.
else goto step 9
8. Print Armstrong then go to step 10
9. Print not Armstrong
10. Stop

(Flow chart)



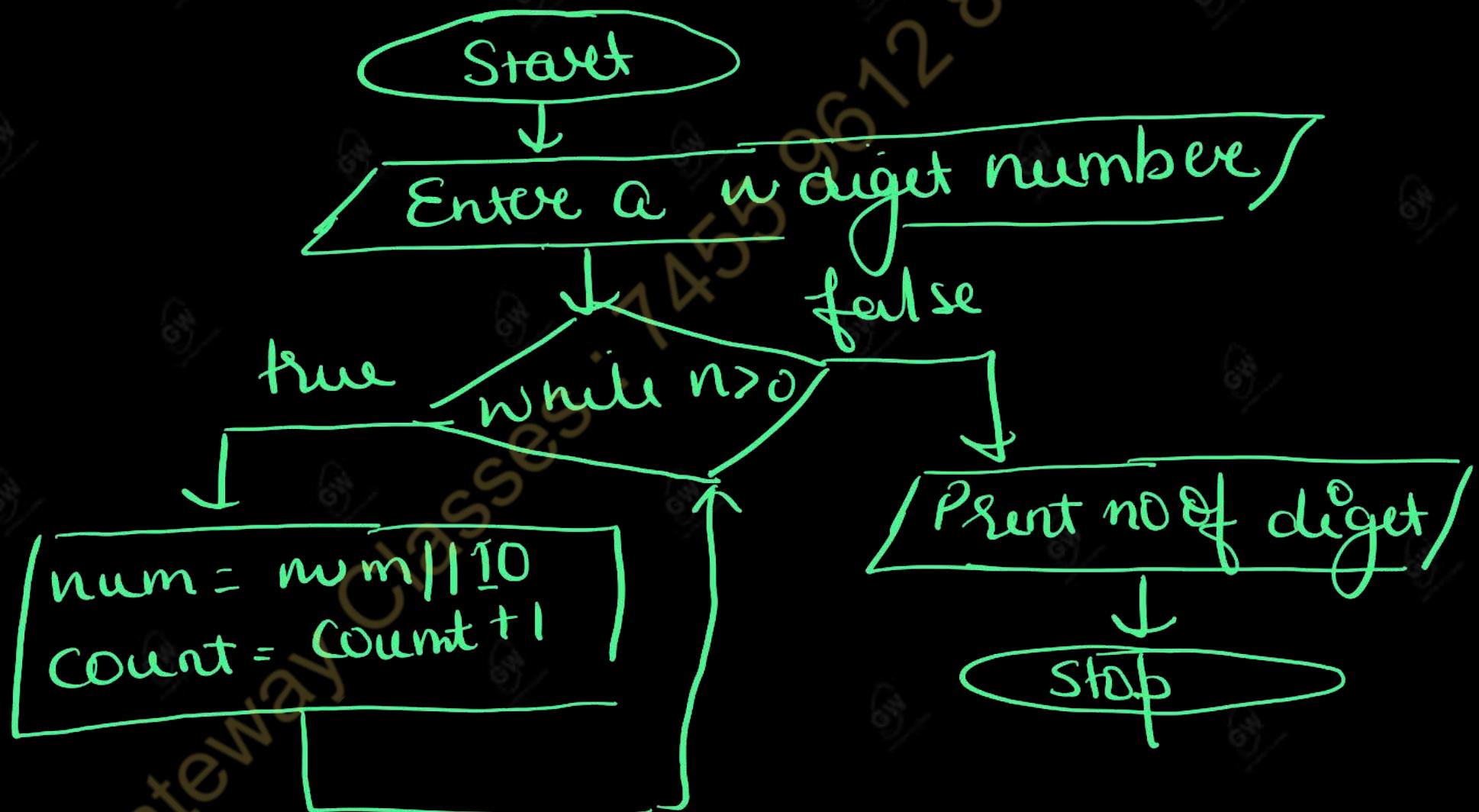
write a program to count the number of digit in a n digit number

```
num=int(input("enter the number"))
count = 0
while num >0 :
    num=num//10
    count= count+ 1
print("Number of digits: ", count)
```

Algorithm

1. Start
2. Enter the n digit number
3. Count = 0
4. while $\text{num} > 0$ true then go to step 5
otherwise go to step 6
5. $\text{num} = \text{num} // 10$
Count = Count + 1
6. Print no of digit
7. Stop

Flowchart



MODULE 2 :conditional and loops

Lecture -8

Today's Target

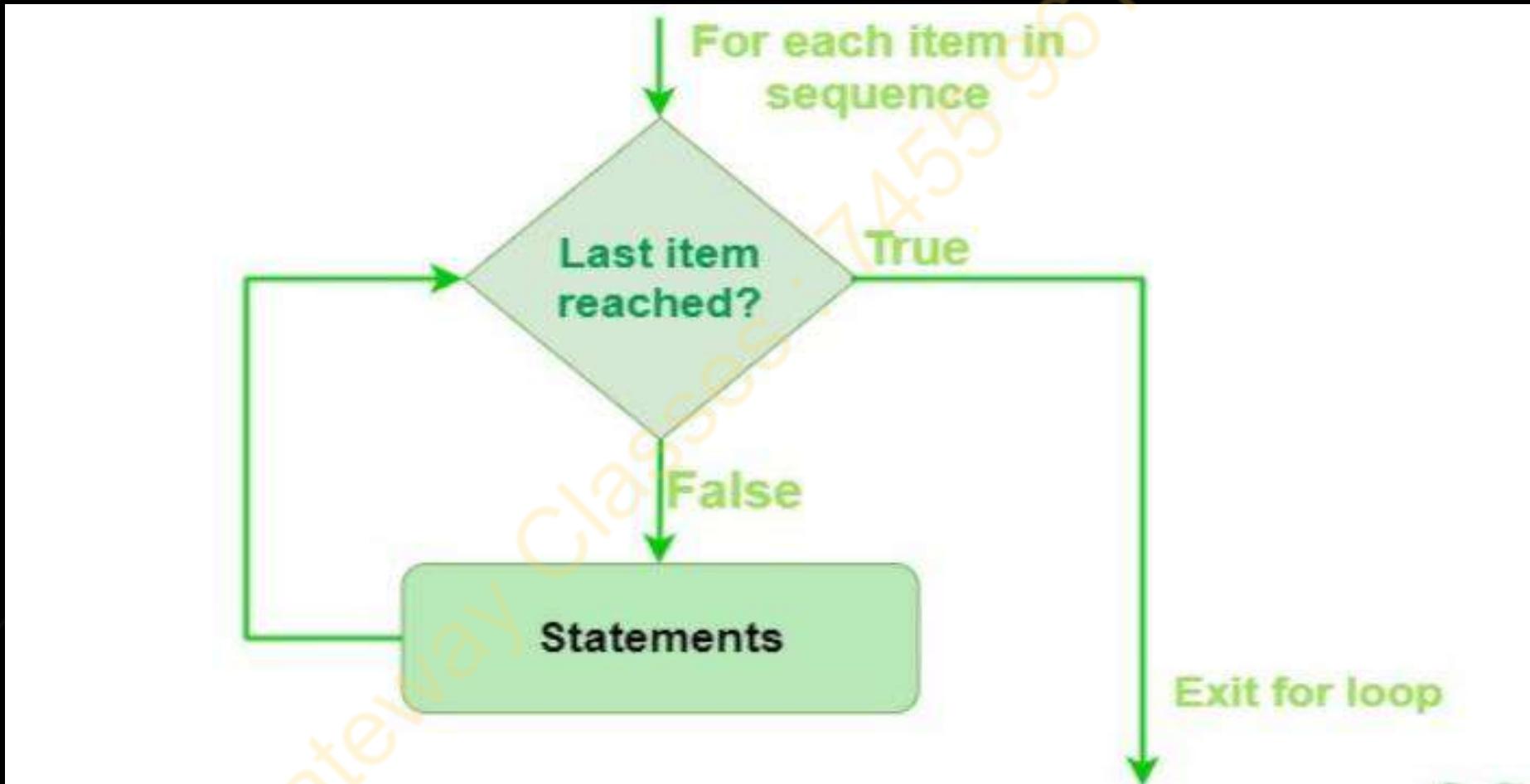
- For loop

By PRAGYA RAJVANSI

B.Tech, M.Tech(C.S.E)

- Python For loop is used for sequential traversal i.e. it is used for iterating over an iterable like String, Tuple, List, Set, or Dictionary
- `for var in iterable:`
- `# statements`
- Here the iterable is a collection of objects like lists, and tuples. The indented statements inside the for loops are executed once for each item in an iterable. The variable var takes the value of the next item of the iterable each time through the loop.

For loop



For loop with list

```
1 l = ["gateway", "classes", "hello"]
2 for i in l:
3     print(i)
4
5
```

```
input
gateway
classes
hello
```

```
... Program finished with exit code 0
Press ENTER to exit console. □
```

For loop with dictionary

```
1 person = {"name": "John", "age": 30, "city": "New York"}  
2 for key in person.keys():  
3     print(key)
```

4

5

6

input

- ▼
- ▶
- ✖
- ✖

name
age
city

p

```
1 person = {"name": "John", "age": 30, "city": "New York"}  
2 for value in person.values():  
3     print(value)  
4  
5
```

A screenshot of a Python code editor. The code in the editor is:

```
1 person = {"name": "John", "age": 30, "city": "New York"}  
2 for value in person.values():  
3     print(value)  
4  
5
```

The output window shows the results of the print statements:

```
John  
30  
New York
```

The word "input" is visible in the status bar at the bottom right.

For loop with dictionary

```
1 person = {"name": "John", "age": 30, "city": "New York"}  
2 for key, value in person.items():  
3     print(key, ":", value)  
4  
5  
6
```

```
name : John  
age : 30  
city : New York
```

input

For loop with string

main.py

```
1 print("String Iteration")
2 s = "python"
3 for i in s:
4     print(i)
5
6
```

```
String Iteration
p
y
t
h
o
n
```

input

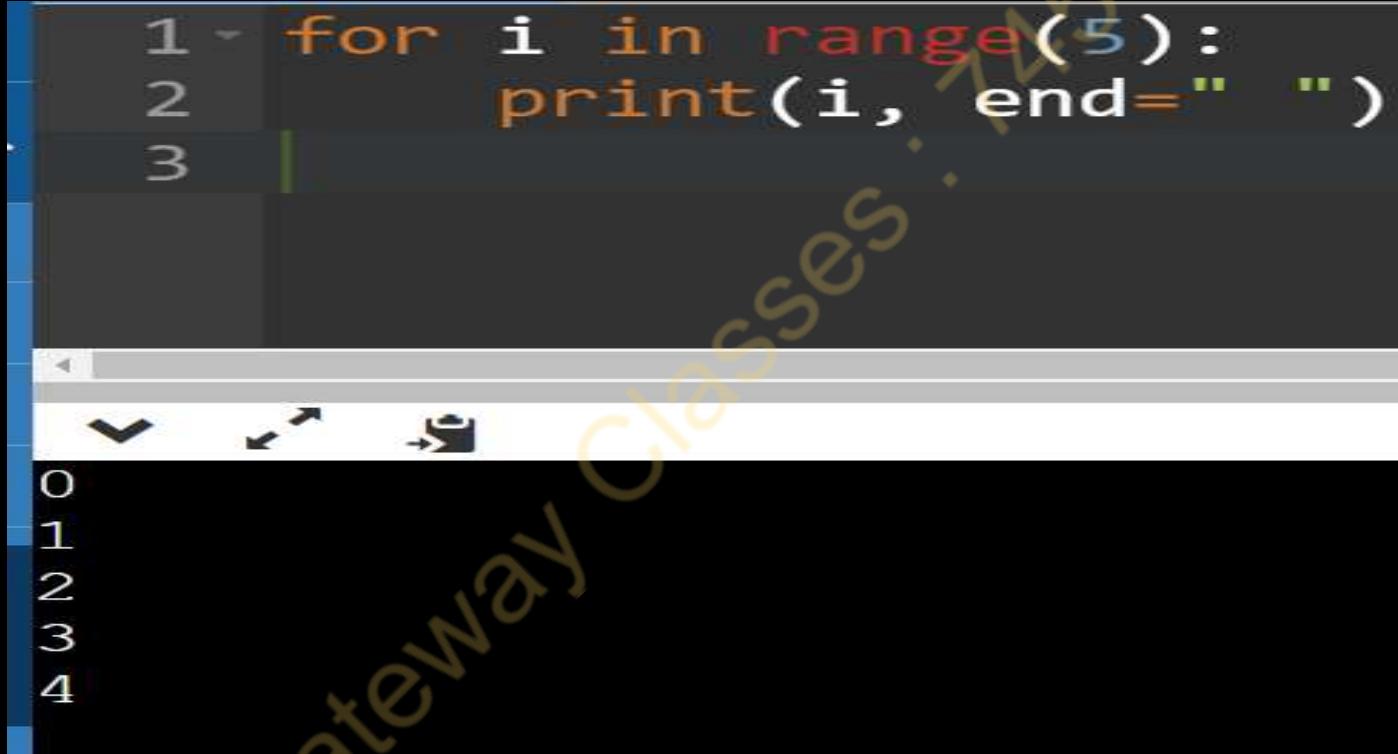
Gateway Classes .1455961284

Python For Loop with a step size

- This code uses a for loop in conjunction with the `range()` function to generate a sequence of numbers starting from 0, up to (but not including) 10, and with a step size of 2. For each number in the sequence, the loop prints its value using the `print()` function. The output will show the numbers 0, 2, 4, 6, and 8.

```
1 for i in range(0, 10, 2):  
2     print(i)  
3  
0  
2  
4  
6  
8
```

- The Python **range()** function returns a sequence of numbers, in a given range. The most common use of it is to iterate sequences on a sequence of numbers using Python loop



```
1 for i in range(5):  
2     print(i, end=" ")  
3  
0  
1  
2  
3  
4
```

syntax: range(start, stop, step)

Parameter :

start: [optional] start value of the sequence

stop: next value after the end value of the sequence

step: [optional] integer value, denoting the difference between any two numbers in the sequence

Return : Returns an object that represents a sequence of numbers

- In simple terms, `range()` allows the user to generate a series of numbers within a given range. Depending on how many arguments the user is passing to the function, the user can decide where that series of numbers will begin and end, as well as how big the difference will be between one number and the next. Python `range()` function takes can be initialized in 3 ways.
- `range (stop)` takes one argument.
- `range (start, stop)` takes two arguments.
- `range (start, stop, step)` takes three arguments.

- When the user call range() with one argument, the user will get a series of numbers that starts at 0 and includes every whole number up to, but not including, the number that the user has provided as the stop.

```
1 for i in range(6):  
2     print(i, end=" ")
```

- When the user call **range()** with two arguments, the user gets to decide not only where the series of numbers stops but also where it starts, so the user doesn't have to start at 0 all the time. Users can use **range()** to generate a series of numbers from X to Y using **range(X, Y)**.

The screenshot shows a code editor window with a blue title bar labeled "main.py". The code in the editor is:

```
1 for i in range(5, 20):
2     print(i, end=" ")
```

Below the code editor is a terminal window showing the output of the program. The output consists of a series of numbers from 5 to 19, each followed by a space, indicating that the loop iterated from 5 to 19 with a step of 1. The output is:

```
5 6 7 8 9 10 11 12 13 14 15 16 17 18 19
```

At the bottom of the terminal window, there is a message indicating the program has finished and prompting the user to press Enter to exit.

```
...Program finished with exit code 0
Press ENTER to exit console.
```

Python range (start, stop, step)

```
1 for i in range(0, 30, 4):  
2     print(i, end=" ")
```

0 4 8 12 16 20 24 28

Incrementing the Range using a Positive Step

```
1 for i in range(0, 30, 4):  
2     print(i, end=" ")
```

0 4 8 12 16 20 24 28

Python range() using Negative Step

main.py

```
1 for i in range(25,2,-2):  
2     print(i,end=" ")
```

```
25 23 21 19 17 15 13 11 9 7 5 3
```

Python range() with Float Values

```
1 for i in range(3.3):  
2     print(i)  
3
```

input

Traceback (most recent call last):

File "/home/main.py", line 1, in <module>
 for i in range(3.3):

TypeError: 'float' object cannot be interpreted as an integer

MODULE 2 :conditional and loops

Lecture -9

Today's Target

- For loop program
- Print table of a number
- Factorial of a number
- Fibonacci series and many more

By PRAGYA RAJVANSI

B.Tech, M.Tech(C.S.E)

Write a program to print the factorial of a number

```
1 num=int(input("enter the number\n"))
2 factorial = 1
3 if num < 0:
4     print("Sorry, factorial does not exist for negative numbers")
5 elif num == 0:
6     print("The factorial of 0 is 1")
7 else:
8     for i in range(1,num + 1):
9         factorial = factorial*i
10    print("The factorial of",num,"is",factorial)
11
```

Write a program to print the Fibonacci series up to n term

```
1 nterms = int(input("How many terms? "))
2 n1, n2 = 0, 1
3 count = 0
4 if nterms <= 0:
5     print("Please enter a positive integer")
6 elif nterms == 1:
7     print("Fibonacci sequence upto",nterms,:")
8     print(n1)
9 elif nterms== 2:
10    print("Fibonacci sequence upto",nterms,:")
11    print(n1)
12    print(n2)
```

Write a program to print the Fibonacci series up to n term

```
else:  
    print("Fibonacci sequence:")  
    print(n1)  
    print(n2)  
    for i in range(3,nterms+1):  
        nth = n1 + n2  
        print(nth)  
        n1 = n2  
        n2 = nth
```

Write a program to print the table of a number

```
number=int(input("enter the number\n"))
print("the multiplication table of", number)
for count in range (1,11):
    print(number,"x",count,"=",number* count)
```

Write a program to print the number of even number in a range

```
1 maximum = int(input(" Please Enter the Maximum Value : "))
2
3 for number in range(1, maximum+1):
4     if(number % 2 == 0):
5         print("{0}".format(number))
6
7
8
```

Write a program to count the number of even number in a range

```
1 maximum = int(input(" Please Enter the Maximum Value : "))
2 count=0
3 for number in range(1, maximum+1):
4     if(number % 2 == 0):
5         print("{0}".format(number))
6         count=count+1
7 print("total even number in range",count)
8
9
10
```

```
maximum = int(input(" Please Enter the Maximum Value : "))
total = 0

for number in range(1, maximum+1):
    if(number % 2 == 0):
        print("{0}".format(number))
        total = total + number

print("The Sum of Even Numbers from 1 to {0} = {1}".format(number, total))
```

Write a program to print the number of odd number in a range

```
1 maximum = int(input(" Please Enter the Maximum Value : "))
2 for number in range(1, maximum+1):
3     if(number % 2 != 0):
4         print("{0}".format(number))
5
6
```

Write a program to count the number of odd number in a range

```
1 maximum = int(input(" Please Enter the Maximum Value : "))
2 count=0
3 for number in range(1, maximum+1):
4     if(number % 2 != 0):
5         print("{0}".format(number))
6         count=count+1
7 print("total number of odd number in a range", count)
8
```

write a python program to display the n terms of odd natural numbers and their sum.

```
1 maximum = int(input(" Please Enter the Maximum Value : "))
2 total = 0
3
4 for number in range(1, maximum+1):
5     if(number % 2 != 0):
6         print("{0}".format(number))
7         total = total + number
8
9 print("The Sum of odd Numbers from 1 to {0} = {1}".format(number, total))
10
11
```

MODULE 2 :conditional and loops

Lecture -10

Today's Target

- Pattern printing questions

By PRAGYA RAJVANSHI

B.Tech, M.Tech(C.S.E)

Write a program to print the pattern

main.py

```
1 rows = int(input("Enter number of rows: "))
2
3 for i in range(rows):
4     for j in range(i+1):
5         print("* ", end="")
6     print("\n")
```

```
input
Enter number of rows: 5
*
**
*
***
****
*****
```

Write a program to print the pattern

```
1 rows = int(input("Enter number of rows: "))
2 for i in range(rows):
3     for j in range(i+1):
4         print(j+1, end=" ")
5     print("\n")
```

```
input
Enter number of rows: 5
1
1 2
1 2 3
1 2 3 4
1 2 3 4 5
```

Write a program to print the pattern

```
1 rows = int(input("Enter number of rows: "))
2 ascii_value = 65
3 for i in range(rows):
4     for j in range(i+1):
5         alphabet = chr(ascii_value)
6         print(alphabet, end="")
7
8     ascii_value += 1
9     print("\n")
```

```
Enter number of rows: 3
A
B B
C C C
```

input

Write a program to print the pattern

```
1 rows = int(input("Enter number of rows: "))
2 for i in range(rows, 0, -1):
3     for j in range(0, i):
4         print("* ", end="")
5     print("\n")
```

The screenshot shows a terminal window with a light gray background and a dark gray border. At the top, there are standard window control buttons (minimize, maximize, close) and the word "input". Below the title bar, the command "Enter number of rows: 3" is displayed in black text. The main area of the terminal shows the output of the program, which is a right-angled triangle of asterisks. The first row has three asterisks, the second row has two, and the third row has one. The asterisks are printed in black, and each row ends with a new line character.

```
input
Enter number of rows: 3
* * *
* *
*
```

Write a program to print the pattern

```
1 rows = int(input("Enter number of rows: "))  
2 for i in range(rows, 0, -1):  
3     for j in range(1, i+1):  
4         print(j, end=" ")  
5  
6     print("\n")
```

```
input  
Enter number of rows: 4  
1 2 3 4  
  
1 2 3  
  
1 2  
  
1
```

Write a program to print the pattern

main.py

```
1 rows = int(input("Enter number of rows: "))
2 for i in range(rows, 0, -1):
3     for j in range(0, i):
4         print("* ", end="")
5
6 print("\n")
```

input

Enter number of rows: 3

* * *

* *

*

Write a program to the pattern

```
1 rows = int(input("Enter number of rows: "))
2
3 k = 0
4
5 for i in range(1, rows+1):
6     for space in range(1, (rows-i)+1):
7         print(end=" ")
8
9     while k!=(2*i-1):
10        print("* ", end="")
11        k += 1
12    k=0
13    print()
```

```
input
Enter number of rows: 4
*
 *
 *
 *
 ****
*****
```

MODULE 2 :conditional and loops

Lecture -11

Today's Target

- Armstrong and palindrome within range
- Break and continue

By PRAGYA RAJVANSI

B.Tech, M.Tech(C.S.E)

Write a program to find out the Armstrong number within range

main.py

```
1 low = int(input("Enter low range: "))
2 up = int(input("Enter up range: "))
3 for num in range(low, up + 1):
4     sum = 0
5     temp = num
6     while temp > 0:
7         digit = temp % 10
8         sum += digit ** 3
9         temp //= 10
10    if num == sum:
11        print(num)
```

input

```
Enter low range: 100
Enter up range: 999
153
370
371
407
```

main.py

```
1 low = int(input("Enter low range:"))
2 up = int(input("Enter up range:"))
3 for num in range(low, up + 1):
4     rev = 0
5     temp = num
6     while temp > 0:
7         digit = temp % 10
8         rev = rev*10+digit
9         temp //= 10
10    if num == rev:
11        print(num)
12
```

Enter low range: 12
Enter up range: 34
22
33

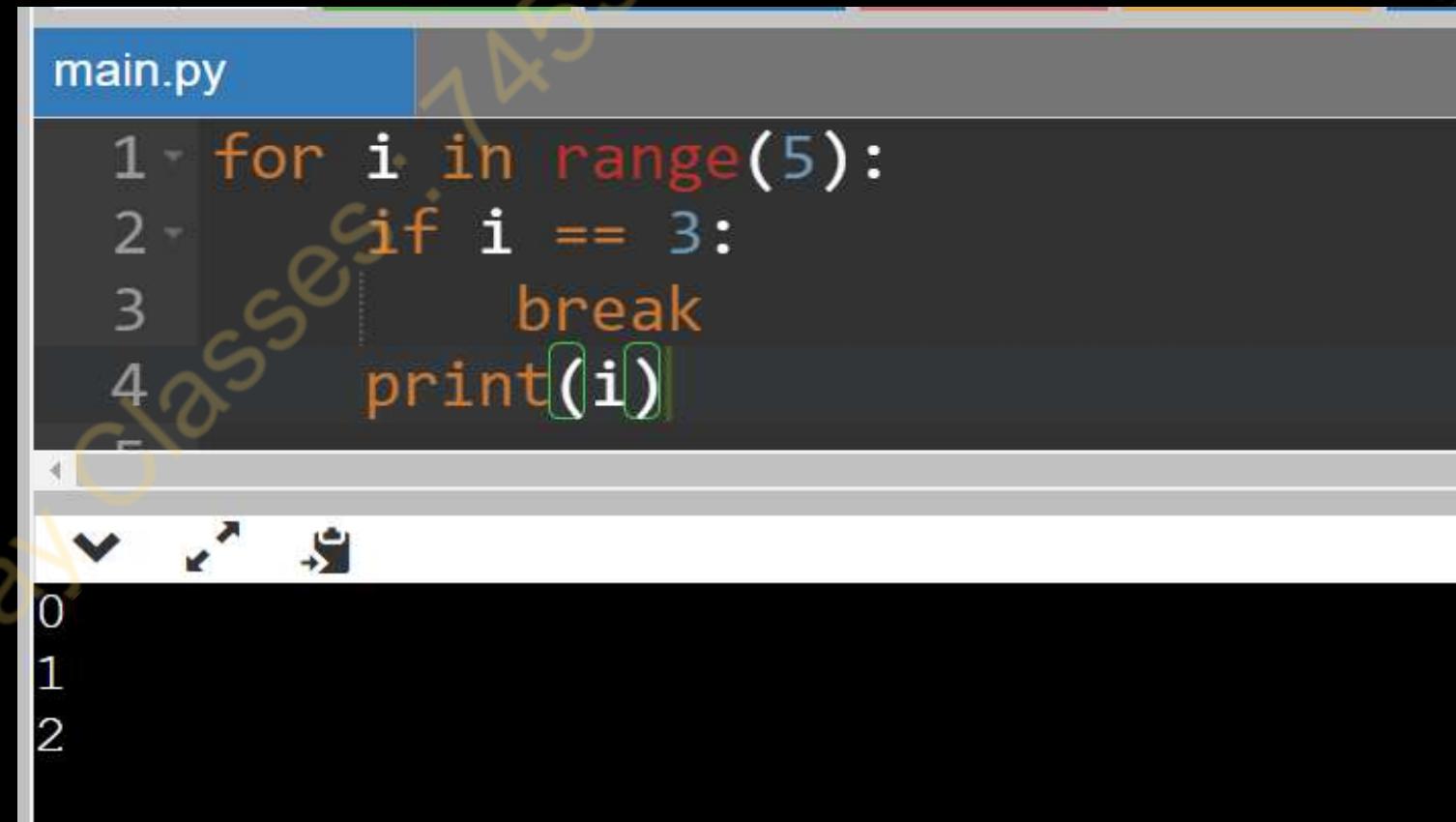
input

break statement

the break statement is used to terminate the loop immediately when it is encountered.

The syntax of the break statement is:

```
break;
```



A screenshot of a code editor window titled "main.py". The code is as follows:

```
1 for i in range(5):
2     if i == 3:
3         break
4     print(i)
```

The code uses a for loop to iterate over the range from 0 to 4. Inside the loop, an if statement checks if the value of i is 3. If it is, the break statement is executed, which immediately terminates the loop. The print statement is only executed for the first two iterations (i=0 and i=1). The output window below the code shows the numbers 0, 1, and 2, corresponding to the printed values.

break statement

```
for val in sequence:  
    # code  
    if condition:  
        break
```

code

```
while condition:  
    # code
```

code

```
if condition:  
    break
```

code

code

break statement

```
1 i = 1
2 while i <= 10:
3     print('6 * ',(i),'=',6 * i)
4     if i >= 5:
5         break
6     i = i + 1
7
```

```
6 * 1 = 6
6 * 2 = 12
6 * 3 = 18
6 * 4 = 24
6 * 5 = 30
```

inp

continue statement

The continue statement is used to skip the current iteration of the loop and the control flow of the program goes to the next iteration.

The syntax of the continue statement is:

Continue;

```
1 for i in range(5):  
2     if i == 3:  
3         continue  
4     print(i)
```

```
0  
1  
2  
4
```

continue statement

```
for val in sequence:  
    # code  
    if condition:  
        continue
```

code

```
while condition:  
    # code  
    if condition:  
        continue
```

code

continue statement

main.py

```
1 num = 0
2 while num < 10:
3     num += 1
4     if (num % 2) == 0:
5         continue
6     print(num)
```

```
1
3
5
7
9
```

Write a program to check whether a number is prime number or not

```
1 num=int(input("enter the number\n"))
2 flag= False
3 if num==1:
4     print(num,"is not a prime number")
5 else:
6     for i in range(2, num):
7         if(num%i==0):
8             flag= True
9             break
10 if flag== True:
11     print("not prime")
12 else:
13     print("prime")
```

MODULE 2 :conditional and loops

Lecture -12

Today's Target

- Pass statement
- Star pattern
- AKTU PYQS'

By PRAGYA RAJVANSHI

B.Tech, M.Tech(C.S.E)

- Python pass statement is a null statement. But the difference between pass and comment is that comment is ignored by the interpreter whereas pass is not ignored.
- SYNTAX
- pass
- **What is pass statement in Python?**
- When the user does not know what code to write, So user simply places a pass at that line. Sometimes, the pass is used when the user doesn't want any code to execute. So users can simply place a pass where empty code is not allowed, like in loops, function definitions, class definitions, or in if statements. So using a pass statement user avoids this error.

Why Python Needs “pass” Statement?

If we do not use pass or simply enter a comment or a blank here, we will receive an **Indentation Error** error message.

```
1 a = 10
2 b = 20
3 if(a<b):
4     pass
5 else:
6     print("b<a")
```



```
...Program finished with exit code 0
Press ENTER to exit console. □
```

p

main.py

```
1 li =['a', 'b', 'c', 'd']
2 for i in li:
3     if(i == 'a'):
4         pass
5     else:
6         print(i)
7
```

b
c
d

Use of pass keyword in Python Class

The pass keyword can also be used in an empty class in Python.

```
class geekClass:  
    pass
```

Purpose of loops

- The purpose of loops is to repeat the same or similar code a number of times . This number of times could be specified to a certain number or number of times to be dictated by certain condition being met

write a program to print the prime number within an interval

```
lower = int(input("enter the upper range\n"))
upper = int(input("enter the lower range\n"))
print("Prime numbers between", lower, "and", upper, "are:")
for num in range(lower, upper + 1):

    if num > 1:
        for i in range(2, num):
            if (num % i) == 0:
                break
        else:
            print(num)
```

for loop	While loop
For loop is used to iterate over a sequence of items.	While loop is used to repeatedly execute a block of statements while a condition is true.
For loops are designed for iterating over a sequence of items. Eg. list, tuple, etc.	While loop is used when the number of iterations is not known in advance or when we want to repeat a block of code until a certain condition is met.
For loop require a sequence to iterate over.	While the loop requires an initial condition that is tested at the beginning of the loop.
For loop is typically used for iterating over a fixed sequence of items	While loop is used for more complex control flow situations.

difference between

for loop

for loop is more efficient than a while loop when iterating over sequences, since the number of iterations is predetermined and the loop can be optimized accordingly.

While loop

While a loop may be more efficient in certain situations where the condition being tested can be evaluated quickly.

Pattern question(AKTU 2021-22)

p

```
    *
   ** 
  *** 
 * *** 
*** *** 
  *** 
 *** 
  ** 
   *
```

```
n = 5
# upper triangle
for i in range(n):
    for j in range(i + 1):
        print('*', end="")
    print()
# Lower triangles
for i in range(n):
    for j in range(n - i - 1):
        print('*', end="")
    print()
```

Pattern question

main.py

```
1 size = 5
2 for i in range(size):
3     for j in range(1, size - i):
4         print(" ", end="")
5     for k in range(0, i + 1):
6         print("*", end="")
7     print()
```

```
*  
**  
***  
****  
*****
```

for loop with tuple

- A tuple is also a sequence of values just like a list. It is immutable and enclosed in parentheses instead of square brackets

```
1
2 items = ('one', 'two', 'three')
3 for item in items:
4     print(item)
5
6
```

```
one
two
three
```

for loop with list

```
1
2 fruits = ['orange', 'apple', 'pear', 'banana', 'kiwi']
3 for fruit in fruits:
4     print(fruit)
```

```
orange
apple
pear
banana
kiwi
```

input

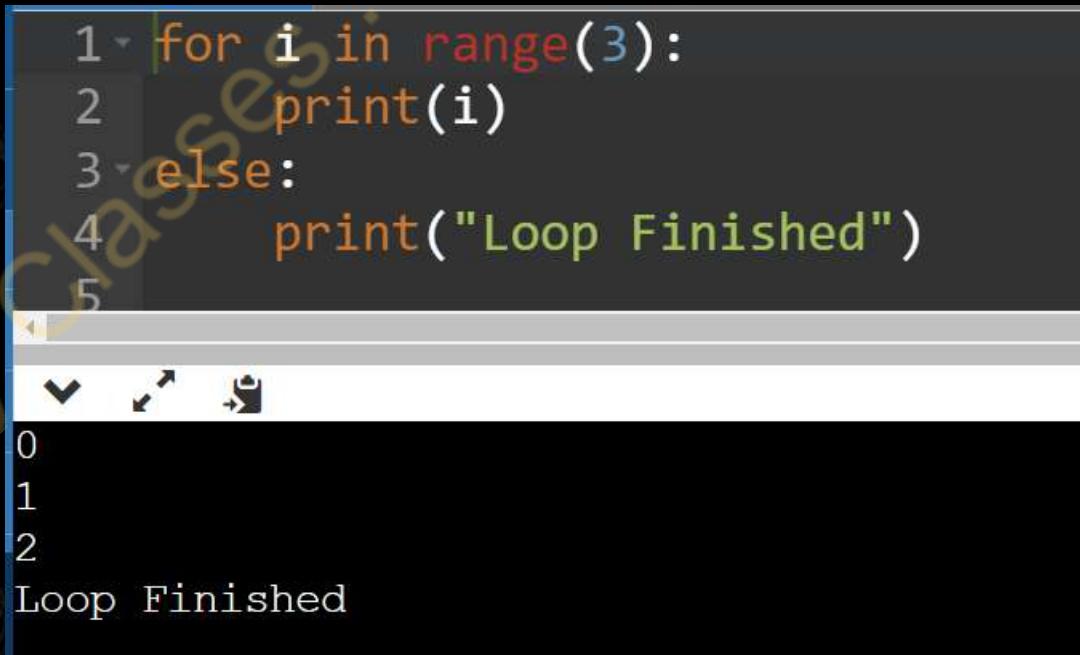
for loop with string

```
1 items = 'looping'
2 for item in items:
3     print(item)
4
```

```
looping
```

for loop with else

- you can also use else keyword in python for loop. This is useful when you want to execute some code when the loop is finished.
- The else block is executed when the loop is finished.



```
1 for i in range(3):  
2     print(i)  
3 else:  
4     print("Loop Finished")  
5  
0  
1  
2  
Loop Finished
```

Q1	Explain all the conditional statement in python using small code example	AKTU 2019-20
Q2	Write a program to check if the input year is a leap year or not	AKTU 2021-22
Q3	Write a program to print the Fibonacci series in python	AKTU 2021-22 AKTU 2019-20
Q4	Any pattern question	AKTU 2021-22
Q5	Explain the following loops with flowdaigram, synatz and with suitable example	AKTU 2022-23

Q6

What is the purpose and working of loops

AKTU 2019-20

AKTU Full Courses (Paid)

Download **Gateway Classes** Application

From Google Play store

All Subjects

Link in Description

Thank You