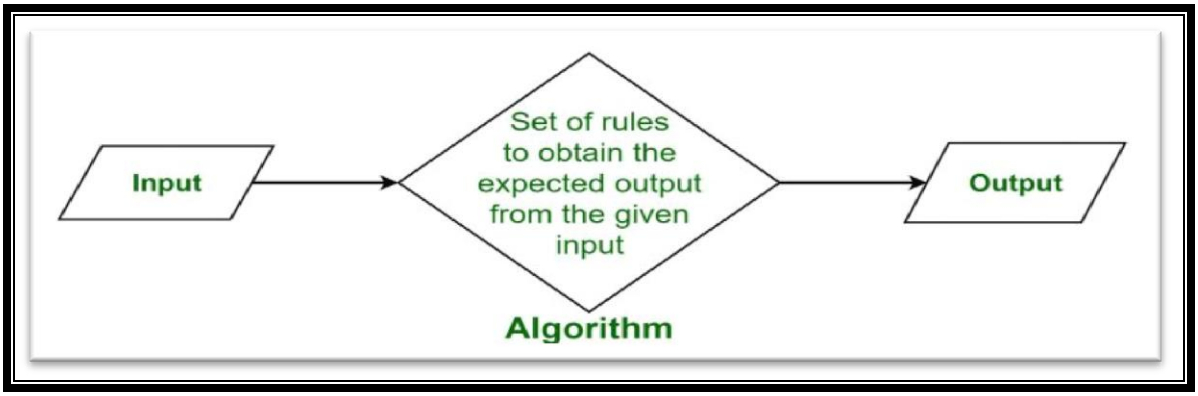
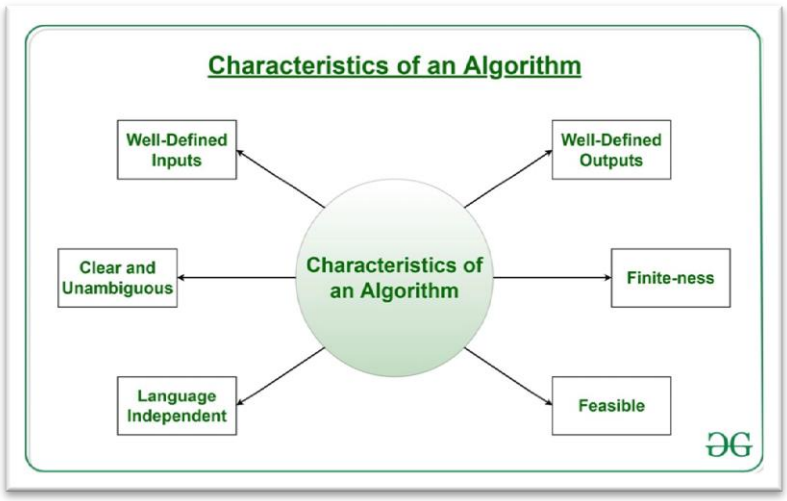
Fundamentals of algorithms

Explain the term algorithm

Algorithm is a stepwise set of instructions written to perform a specific task..



What are the Characteristics of an Algorithm?



- **Clear and Unambiguous**: Algorithm should be clear and unambiguous. Each of its steps should be clear in all aspects and must lead to only one meaning.
- **Well-Defined Inputs**: If an algorithm says to take inputs, it should be well-defined inputs.
- **Well-Defined Outputs:** The algorithm must clearly define what output will be yielded and it should be well-defined as well.
- **Finite-ness:** The algorithm must be finite, i.e. it should not end up in an infinite loops or similar.
- **Feasible:** The algorithm must be simple, generic and practical, such that it can be executed upon will the available resources. It must not contain some future technology, or anything.
- **Language Independent:** The Algorithm designed must be language-independent, i.e. it must be just plain instructions that can be implemented in any language, and yet the output will be same, as expected.

Example:

Write an algorithm to calculate the addition of two given numbers.

Step 1: Start

Step 2: Read two numbers A and B

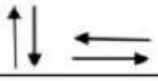
Step 3: Add A and B and store result in C

Step 4: Display C

Step 5: Stop

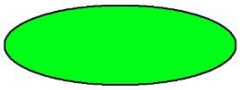
Explain the term Flowchart

Flowchart is a graphical representation of an algorithm.

Symbol	Name	Function
	**Process**	Indicates any type of internal operation inside the Processor or Memory
	input/output	Used for any Input / Output (I/O) operation. Indicates that the computer is to obtain data or output results
	Decision	Used to ask a question that can be answered in a binary format (Yes/No, True/False)
	Connector	Allows the flowchart to be drawn without intersecting lines or without a reverse flow.
	Predefined Process	Used to invoke a subroutine or an Interrupt program.
	Terminal	Indicates the starting or ending of the program, process, or interrupt program
	Flow Lines	Shows direction of flow.

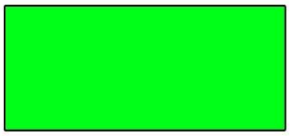
Basic Symbols used in Flowchart Designs

1. **Terminal:** Terminal is the first and last symbols in the flowchart.

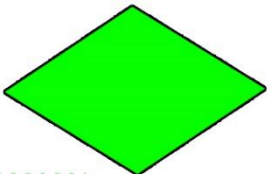


2. **Input/Output:** Program instructions that take input from input devices and display output on output devices are indicated with parallelogram in a flowchart.

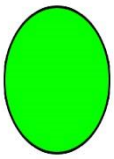
3. **Processing:** All arithmetic processes such as adding, subtracting, multiplication and division are indicated by action or process symbol.



4. **Decision** Diamond symbol represents a decision point. Decision based operations such as yes/no question or true/false are indicated by diamond in flowchart.



5. **Connectors:** Whenever flowchart becomes complex or it spreads over more than one page, it is useful to use connectors to avoid any confusions. It is represented by a circle.



6. **Flow lines:** Flow lines indicate the exact sequence in which instructions are executed. Arrows represent the direction of flow of control and relationship among different symbols of flowchart.

What are the advantages Of Using FLOWCHARTS?

• Communication: Flowcharts are better way of communicating the logic of a system to all concerned or involved.

• Effective analysis: With the help of flowchart, problem can be analysed in more effective way therefore reducing cost and wastage of time.

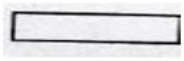
• Proper documentation: Program flowcharts serve as a good program documentation, which is needed for various purposes, making things more efficient.

• Efficient Coding: The flowcharts act as a guide or blueprint during the systems analysis and program development phase.

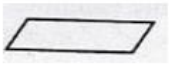
• Proper Debugging: The flowchart helps in debugging process.

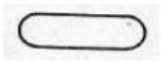
• Efficient Program Maintenance: The maintenance of operating program becomes easy with the help of flowchart. It helps the programmer to put efforts more efficiently on that part

State the use of following symbols used for flowchart drawing:

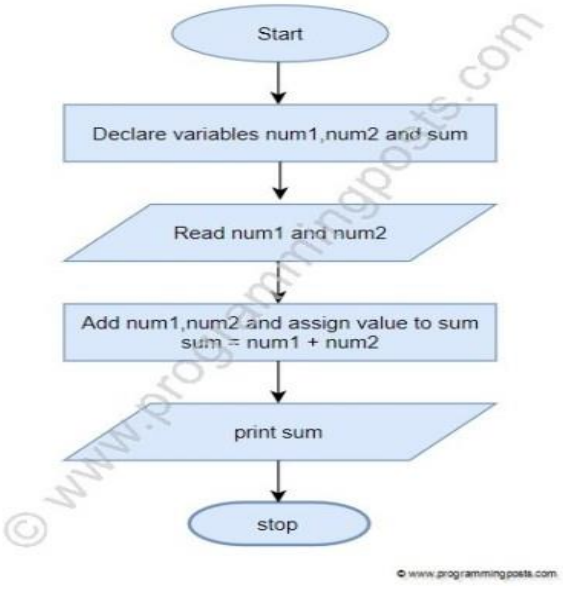
(i)  General processing

(ii)  Decision making

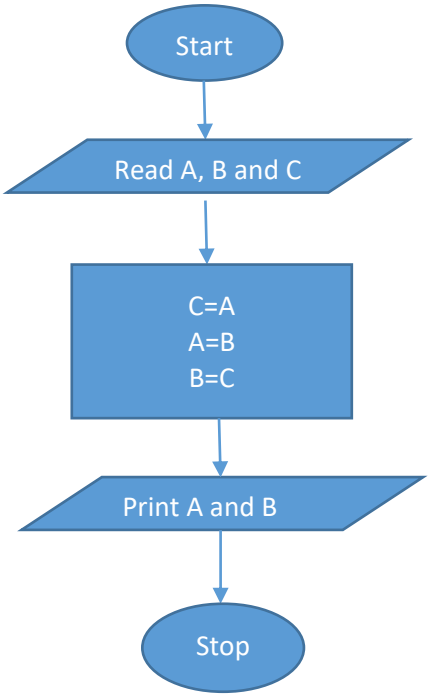
(iii)  Input/ Output statements

(iv)  Start / Stop

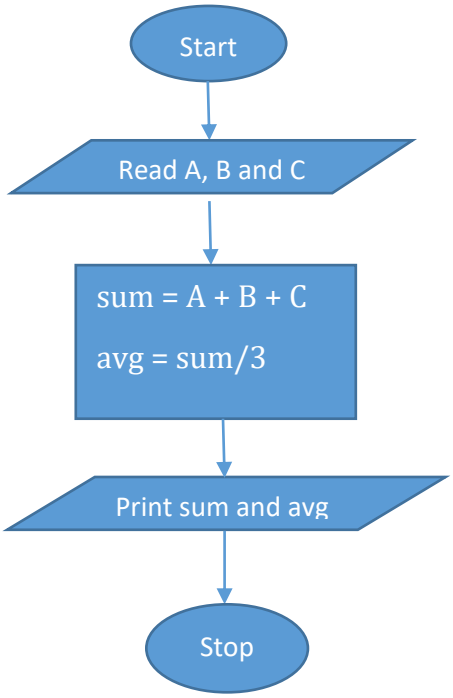
Write algorithm and flowchart of addition of two numbers

Algorithm	Flowchart
Step 1: Start Step 2: Read two numbers A and B Step 3: Add A and B and store result in C Step 4: Display C Step 5: Stop	

Write algorithm and flowchart for swapping of two numbers

Algorithm	Flowchart
Step 1: Start Step 2: Read three numbers A, B and C Step 3: C = A Step 4: A = B Step 5: B = C Step 6: Print A and B Step 7: Stop	

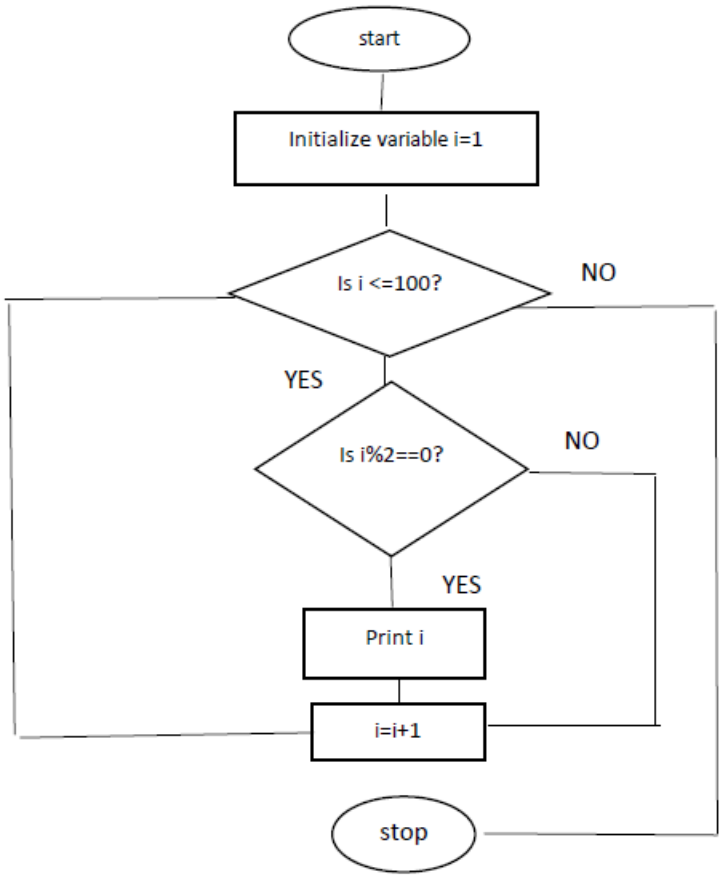
Write algorithm and flowchart to find sum and average of three numbers

Algorithm	Flowchart
Step 1: Start Step 2: Read three numbers A, B and C Step 3: sum = A + B + C Step 4: avg = sum/3 Step 5: Print sum and avg Step 6: Stop	

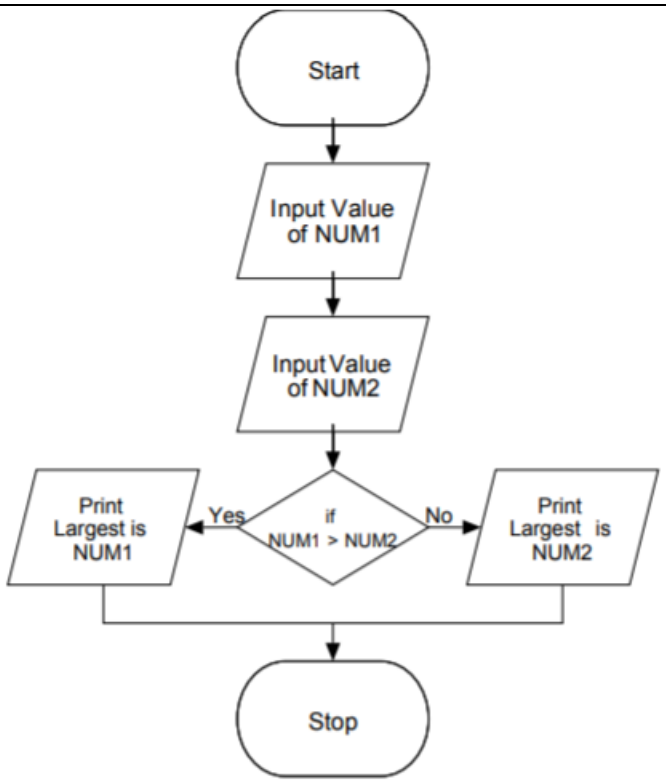
Write an algorithm to determine whether a given number is divisible by 5 or not

Step 1- Start
Step 2- Read / input the number.
Step 3- if n%5==0 then goto step 5.
Step 4- else number is not divisible by 5 goto step 6.
Step 5- display the output number is divisible by 5.
Step 6- Stop

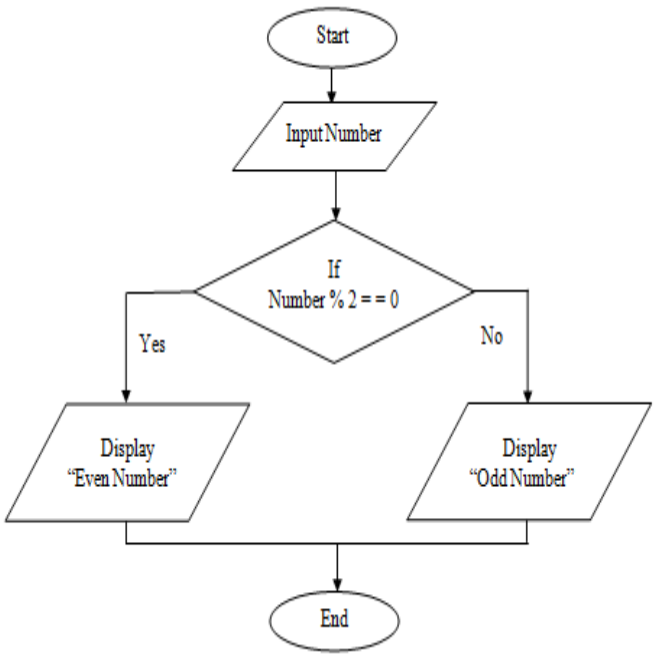
Write algorithm and draw flow-chart to print even numbers from 1 to 100.

Algorithm	Flowchart
Algorithm 1. Start 2. Initialize the variable i to 1. 3. while i<=100 4. if i%2==0 5. print the number 6. increment value of i 7. stop	

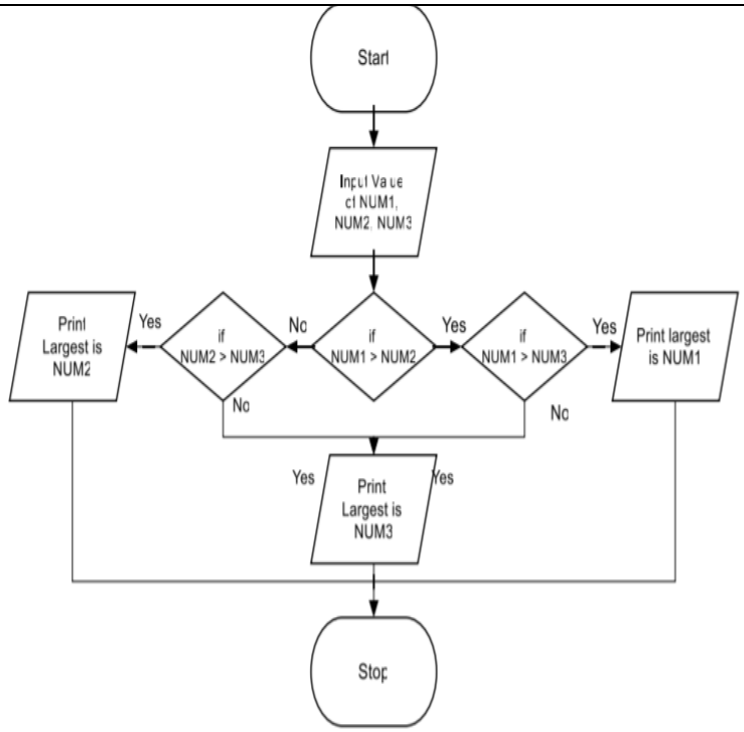
Algorithm & Flowchart to find the largest of two numbers

Algorithm	Flowchart
Algorithm Step-1 Start Step-2 Input two numbers say NUM1,NUM2 Step-3 IF NUM1 > NUM2 THEN print largest is NUM1 ELSE print largest is NUM2 ENDIF Step-4 Stop	

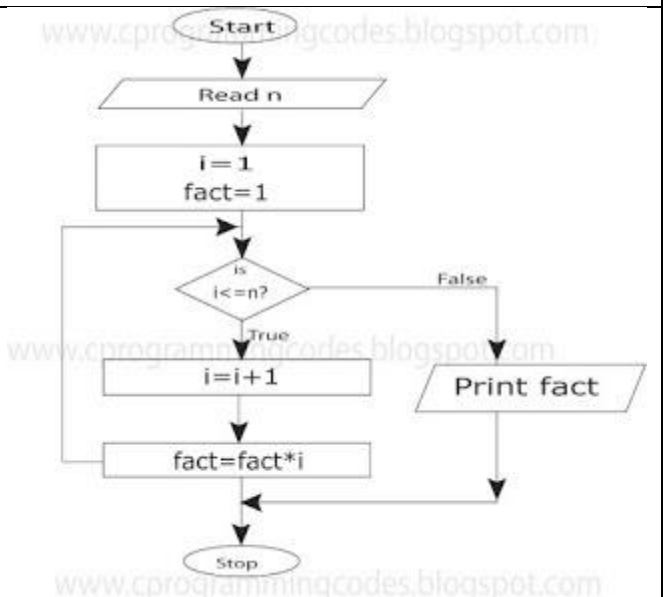
Algorithm & Flowchart to find the given number is odd or even.

Algorithm	Flowchart
Step 1: Start Step 2: [Take Input] Read: Number Step 3: Check: If Number%2 == 0 Then Print : N is an Even Number. Else Print : N is an Odd Number. Step 4: Exit	

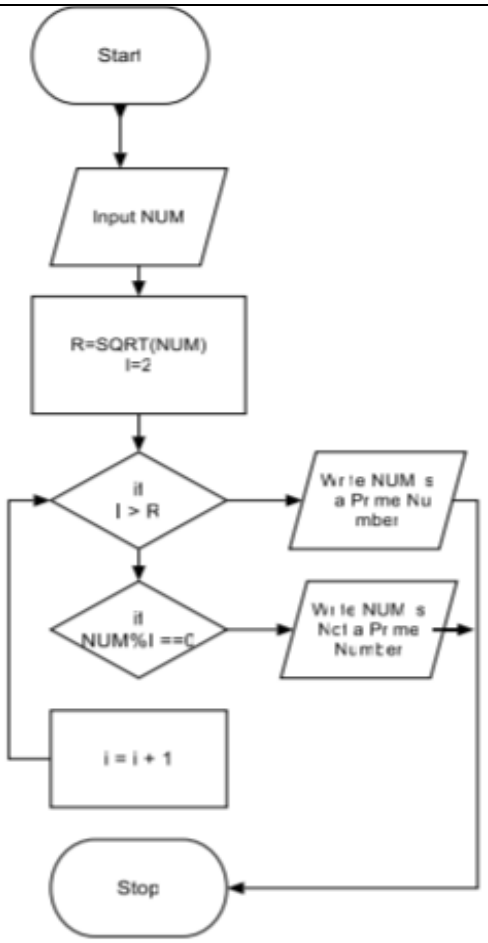
Algorithm & Flowchart to find the largest of three numbers

Algorithm	Flowchart
Step-1 Start **Step-2** Read three numbers say num1,num2, num3 **Step-3** if num1>num2 then go to step-5 **Step-4** IF num2>num3 THEN print num2 is largest ELSE print num3 is largest ENDIF GO TO Step-6 **Step-5** IF num1>num3 THEN print num1 is largest ELSE print num3 is largest ENDIF **Step-6** Stop	

Algorithm & Flowchart to find Factorial of number n (n!=1x2x3x…n)

Algorithm	Flowchart
step 1. Start step 2. Read the number n step 3. [Initialize] i=1, fact=1 step 4. Repeat step 4 through 6 until i=n step 5. fact=fact*i step 6. i=i+1 step 7. Print fact step 8. Stop	

Algorithm & Flowchart to find if a number is prime or not

Algorithm	Flowchart
Step-1 Start Step-2 Input NUM Step-3 R=SQRT(NUM) Step-4 I=2 Step-5 IF (I > R) THEN Write NUM is Prime Number Stop ENDIF Step 6 IF (NUM % I ==0) THEN Write NUM is Not Prime Stop ENDIF Step-7 I = I + 1 Step-8 Go to Step-5	

What is Pseudocode

Definition: Pseudocode is an informal way of programming description that does not require any strict programming language syntax or underlying technology considerations. It is used for creating an outline or a rough draft of a program.

Example:

Write pseudo code that will perform the following.

a) Read in 5 separate numbers.

b) Calculate the average of the five numbers.

c) Find the smallest (minimum) and largest (maximum) of the five entered numbers.

a) Write "please enter 5 numbers"

Read n1,n2,n3,n4,n5

b) Write "The average is"

Set avg to (n1+n2+n3+n4+n5)/5

Write avg

c) If(n1 < n2)

Set max to n2

Else

Set max to n1

If(n3 > max)

Set max to n3

If(n4 > max)

Set max to n4

If(n5 > max)

Set max to n5

Write "The max is"

Write max

C Program to Find Volume and Surface Area of Sphere

The formula used in this program are Surface_area = 4 * Pi * r2, Volume = 4/3 * Pi * r3 where r is the radius of the sphere, Pi= 22/7

```
/*
 * C Program to Find Volume and Surface Area of Sphere
 */
 #include <stdio.h>
#include <math.h>

int main()
{

   float radius;
   float surface_area, volume;

   printf("Enter radius of the sphere : \n");
   scanf("%f", &radius);
   surface_area =  4 * (22/7) * radius * radius;
   volume = (4.0/3) * (22/7) * radius * radius * radius;
   printf("Surface area of sphere is: %.3f", surface_area);
   printf("\n Volume of sphere is : %.3f", volume);
   return 0;
}
```

Output

```
Enter radius of the sphere :
40
Surface area of sphere is: 19200.000
Volume of sphere is : 256000.000
```

C program to swap two numbers (interchange the content of two variables)

```
#include <stdio.h>
 int main()
{
  int x, y, t;
   printf("Enter two integers\n");
  scanf("%d%d", &x, &y);
```

```
    printf("Before Swapping\nFirst integer = %d\nSecond integer = %d\n", x, y);
    t = x;
   x = y;
   y = t;
    printf("After Swapping\nFirst integer = %d\nSecond integer = %d\n", x, y);
    return 0;
  }
```

The output of the program:

Enter two integers

23

45

Before Swapping

First integer = 23

Second integer = 45

After Swapping

First integer = 45

Second integer = 23

If a five-digit number is input through the keyboard, write a program to calculate the sum of its digits. (Hint: Use the modulus operator '%')

```
#include <stdio.h>
#include < conio.h>

int main()
{
      long int Num;
      int Digit1, Digit2,Digit3, Digit4, Digit5 , Sum=0;
      Clrscr();
      Printf("Enter 5 digit Number : ");
      Scanf("%ld",&Num);
      Digit1 = Num % 10;
      Num = Num / 10;
      Digit2 = Num % 10;
```

```
Num = Num / 10;
Digit3 = Num % 10;
Num = Num / 10;
Digit4 = Num % 10;
Num = Num / 10;
Digit5 = Num % 10;
Num = Num / 10;
Sum = Digit1 + Digit2 + Digit3 + Digit4 + Digit5;
printf("Sum of Digits = %d", Sum);
getch();

Return 0;
}
```

Draw a flowchart for checking whether given number is prime or not.