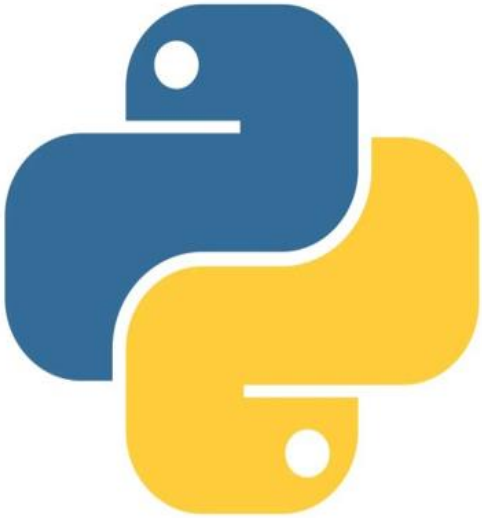


Python



By

Dr. rabih ali sbera

Department of Computer Engineering Techniques

2024-2025

FlowChart

What is a Flowchart?
A Flowchart is a diagram that uses graphic symbols to depict the nature and flow of the steps in a process (Viewgraph 1). Another name for this tool is "flow diagram."

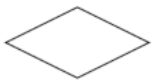
When should teams use Flowcharts?
At the beginning of your process improvement efforts, an as-is Flowchart helps your team and others involved in the process to understand how it currently works. The team may find it helpful to compare this as-is Flowchart with a diagram of the way the process is supposed to work. Later, the team will develop a Flowchart of the modified process—again, to record how it actually functions. At some point, your team may want to create an ideal Flowchart to show how you would ultimately like the process to be performed.

Benefits of Using Flowcharts

- ✓ Promote process understanding
- ✓ Provide tool for training
- ✓ Identify problem areas and improvement opportunities
- ✓ Depict customer-supplier relationships

What symbols are used in Flowcharts?
The **symbols** that are commonly used in Flowcharts (Viewgraph 3) have **specific meanings** and are **connected by arrows indicating the flow** from one step to another:

Oval. Ovals indicate both the starting point and the ending point of the process steps.
Box. A box represents an individual step or activity in the process.
Diamond. A diamond shows a decision point, such as yes/no or go/no-go. Each path emerging from the diamond must be labeled with one of the possible answers.
Circle. A circle indicates that a particular step is connected to another page or part of the Flowchart. A letter placed in the circle clarifies the continuation.

Symbol Name	Symbol	function
Oval		Used to represent start and end of flowchart
Parallelogram		Used for input and output operation
Rectangle		Processing: Used for arithmetic operations and data-manipulations
Diamond		Decision making. Used to represent the operation in which there are two/three alternatives, true and false etc

Mathematical Operators:

Operator	Meaning	Example
+	Addition	A + B
-	Subtraction	A – B
*	Multiplication	A * B
/	Division	A / B
^	Power	A^3 for A^3
%	Reminder	A % B

Relational Operators

Operator	Meaning	Example
<	Less than	A < B
<=	Less than or equal to	A <= B
= or ==	Equal to	A = B
# or !=	Not equal to	A # B or A !=B
>	Greater than	A > B
>=	Greater tha or equal to	A >= B

Logical Operators

Operator	Example	Meaning
AND	A < B AND B < C	Result is True if both A<B and B<C are true else false
OR	A< B OR B < C	Result is True if either A<B or B<C are true else false
NOT	NOT (A >B)	Result is True if A>B is false else true

Selection control Statements

Selection Control	Example	Meaning

IF (Condition) Then
…
ENDIF
	IF (X > 10) THEN
	Y=Y+5
	ENDIF
		If condition X>10 is True execute the statement between THEN and ENDIF

IF (Condition) Then
…
ELSE
…..
 	
ENDIF
	IF (X > 10) THEN
	Y=Y+5
	ELSE
	Y=Y+8
	Z=Z+3
	ENDIF
		If condition X>10 is True execute the statement between THEN and ELSE otherwise execute the statements between ELSE and ENDIF

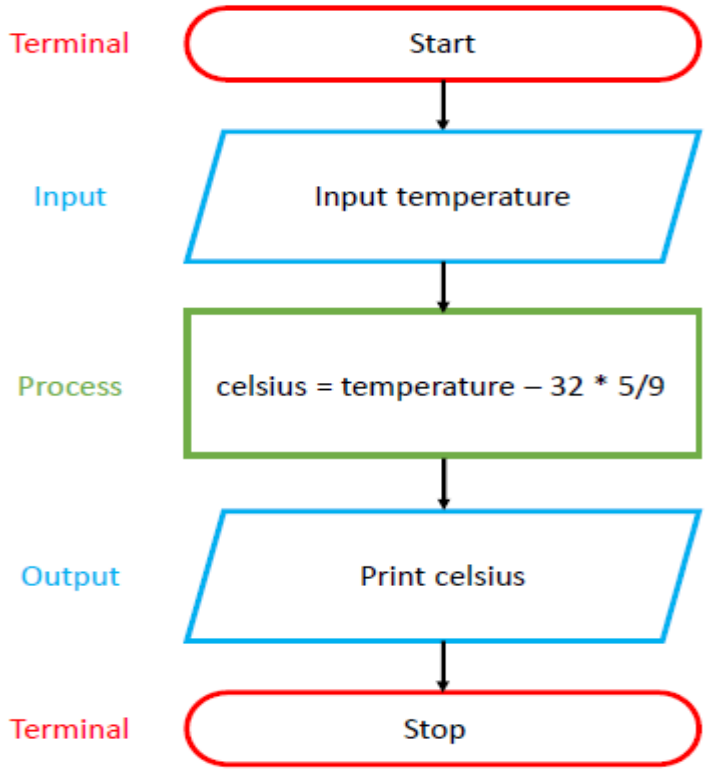
Loop control Statements

Selection Control	Example	Meaning

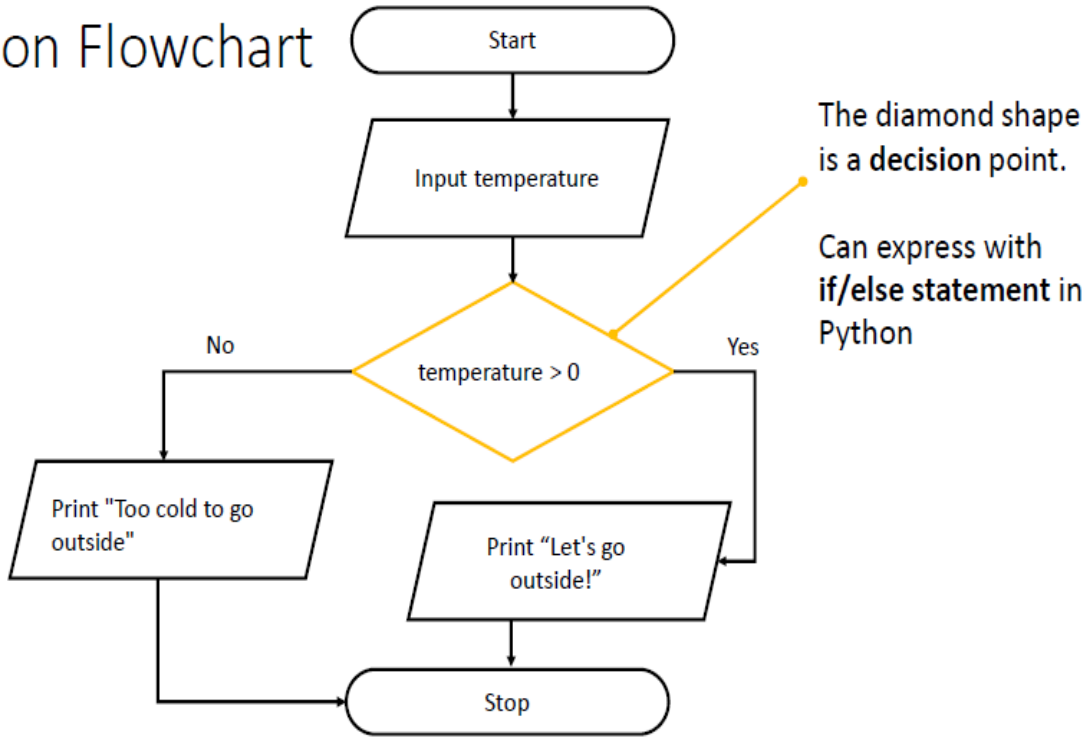
WHILE (Condition)
DO
..
..
ENDDO
	WHILE (X < 10)
	DO
	print x
	x=x+1
	ENDDO
		Execute the loop as long as the condition is TRUE

DO …. … UNTILL (Condition)	DO print x x=x+1 UNTILL (X >10)	Execute the loop as long as the condition is false

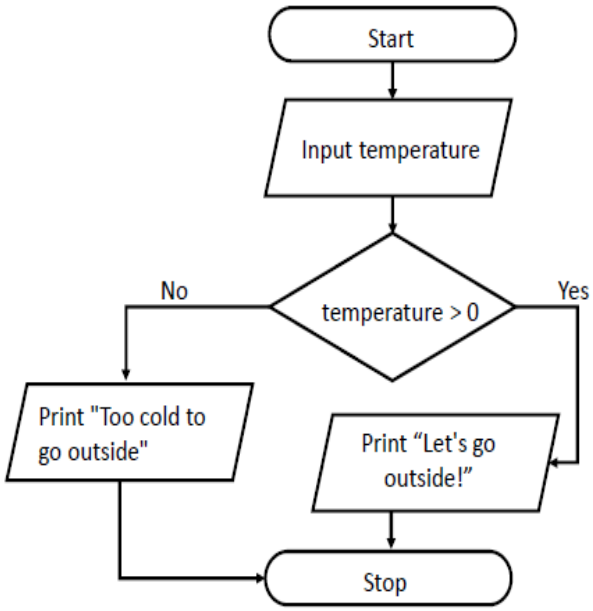
Parts of a Flowchart
• Terminal: start or end of a flowchart
• Input/Output: input or output operations
• Process: indicates computations
or data manipulations (e.g. assigning a variable)
Note that each part has a defined shape



Decision Flowchart



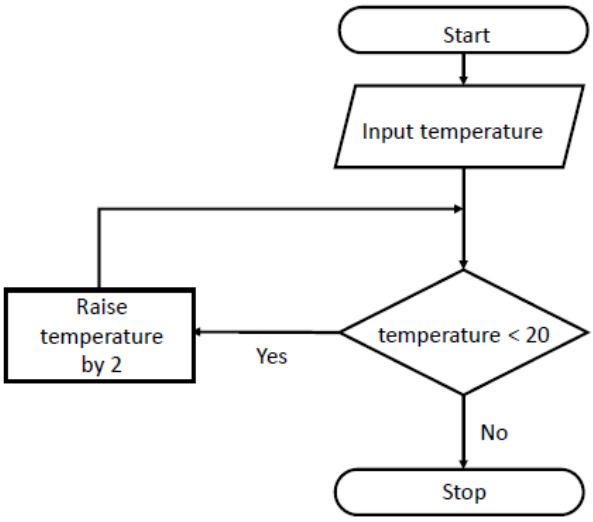
Decision Flowchart



Python Code

```
temperature = float(input("Temperature: "))
if temperature > 0:
    print("Let's go outside!")
else:
    print("Too cold to go outside")
```

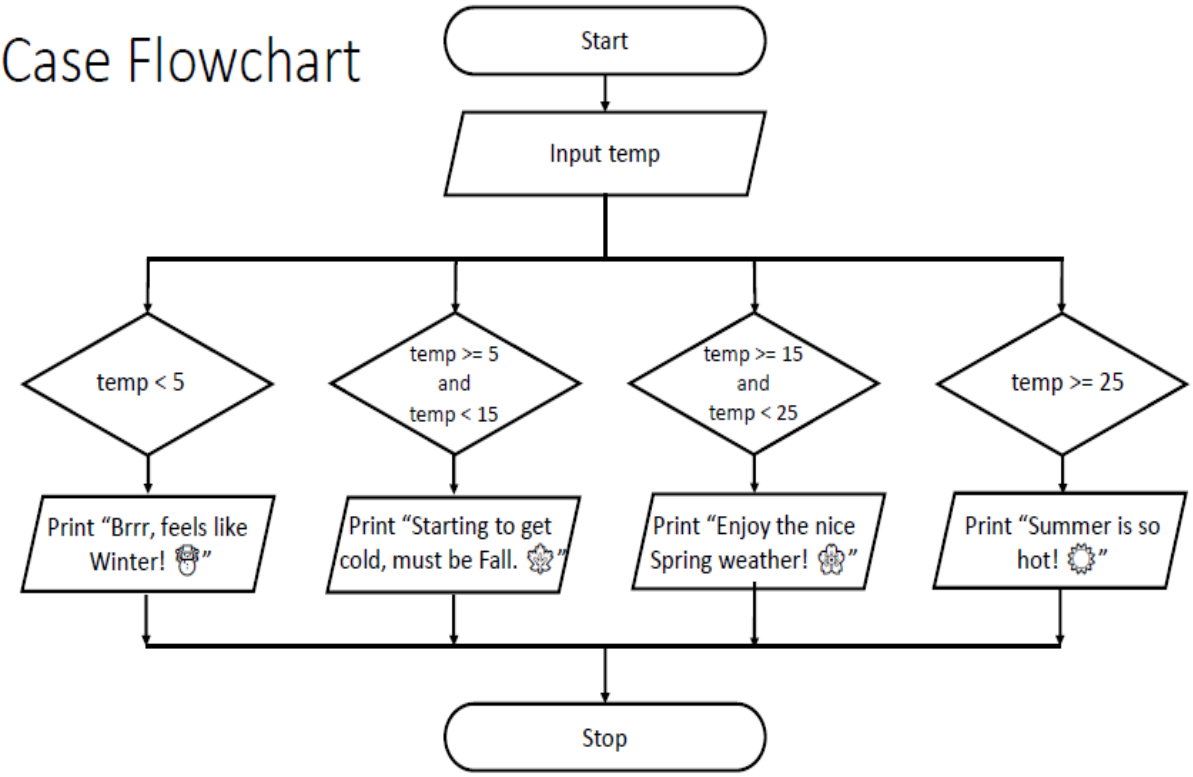
Repetition Flowchart



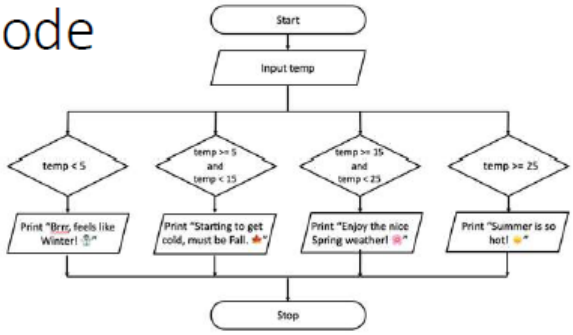
Python Code

```
temperature = float(input("Temperature: "))
while temperature < 20:
    temperature += 2
```

Case Flowchart

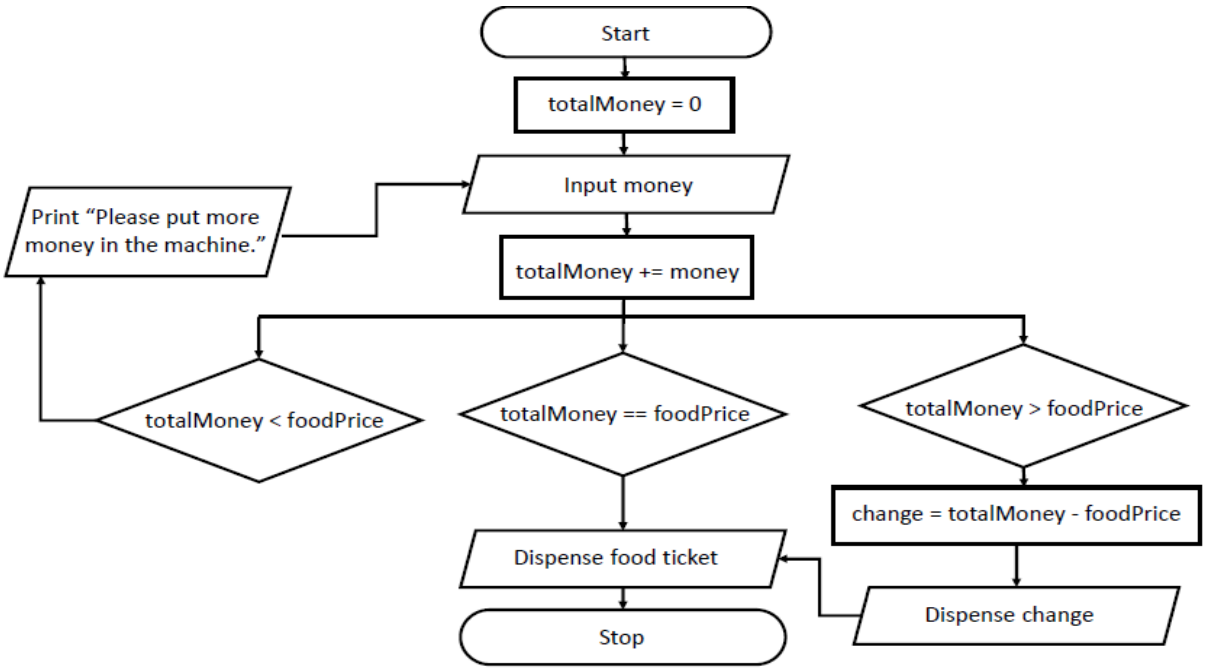


Case Flowchart: Python Code

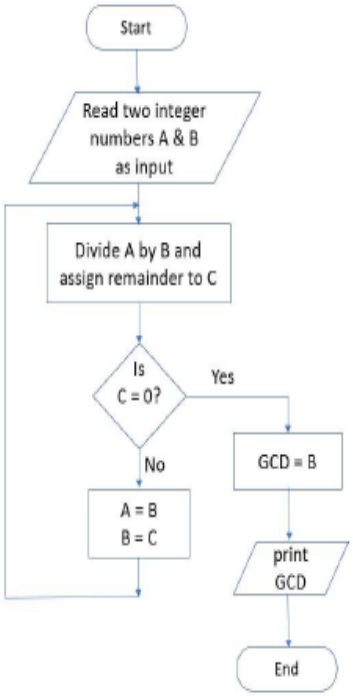



```
temp = float(input("Temperature:"))

if temp < 5:
   print("Brrr, feels like Winter! ")
elif temp >= 5 and temp < 15:
   print("Starting to get cold, must be Fall. ")
elif temp >= 15 and temp < 25:
   print("Enjoy the nice Spring weather! ")
else:
   print("Summer is so hot! ")
```



Write a program for the following flowchart:



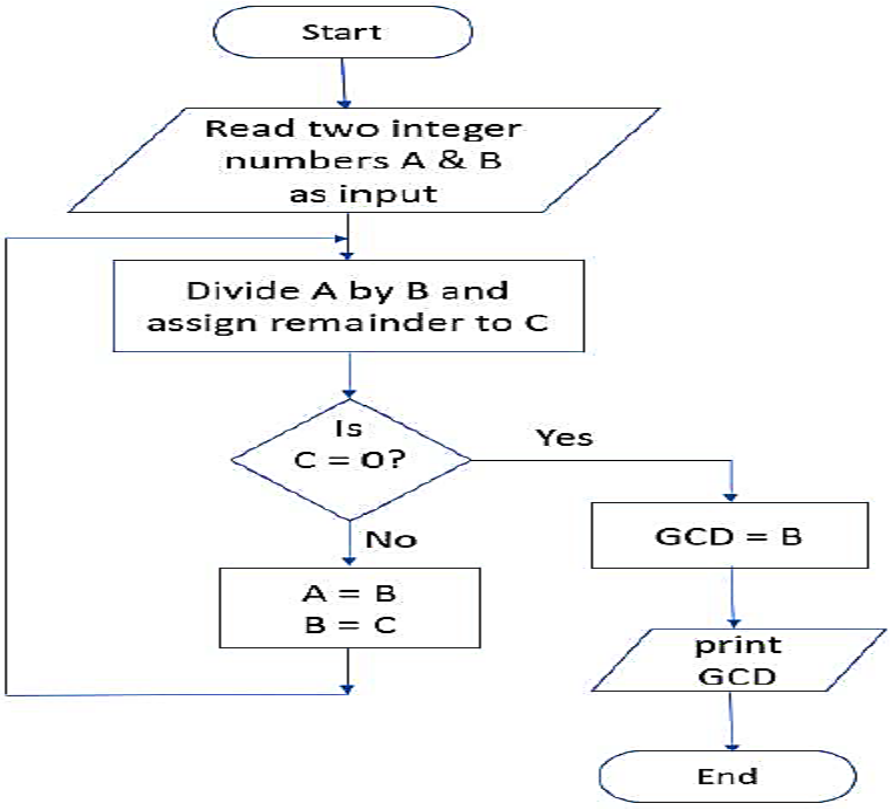
- This flowchart corresponds to calculation of the greatest common divisor (GCD) of two numbers using the Euclidian method
- Write Python code (without using a function) to calculate the greatest common divisor of two integers A & B following this flowchart

Solution – gcd.py

```
a = int(input("Enter A: "))
b = int(input("Enter B: "))
gcd = 0

while(a > 0 and b > 0):
    c = a%b
    if c == 0:
        gcd = b
        break
    a = b
    b = c

print("The greatest common divisor is: ", gcd)
```



Algorithm Example

How we thinking to build a flow chart diagram

Algorithm & Flowchart to find the sum of two numbers

Algorithm

Step-1 Start

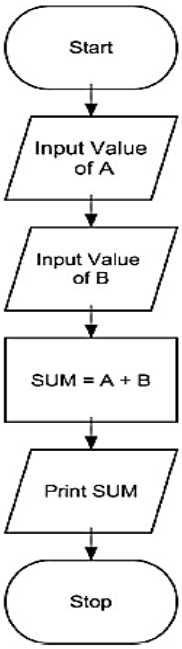
Step-2 Input first numbers say A

Step-3 Input second number say B

Step-4 SUM = A + B

Step-5 Display SUM

Step-6 Stop



OR

Algorithm

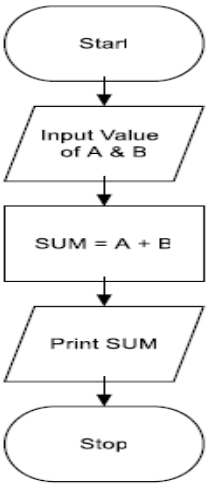
Step-1 Start

Step-2 Input two numbers say A & B

Step-3 SUM = A + B

Step-4 Display SUM

Step-5 Stop



Algorithm & Flowchart to convert temperature from Celsius to Fahrenheit

C : temperature in Celsius
F : temperature Fahrenheit

Algorithm

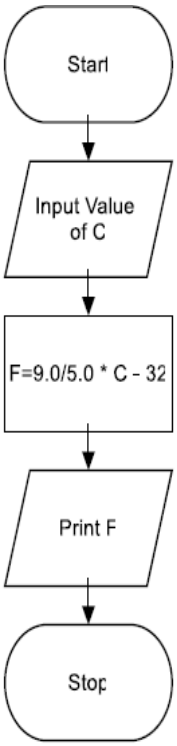
Step-1 Start

Step-2 Input temperature in Celsius say C

Step-3 F = (9.0/5.0 x C) + 32

Step-4 Display Temperature in Fahrenheit F

Step-5 Stop



Algorithm & Flowchart to find Area and Perimeter of Square

L : Side Length of Square
AREA : Area of Square
PERIMETER : Perimeter of Square

Algorithm

Step-1 Start

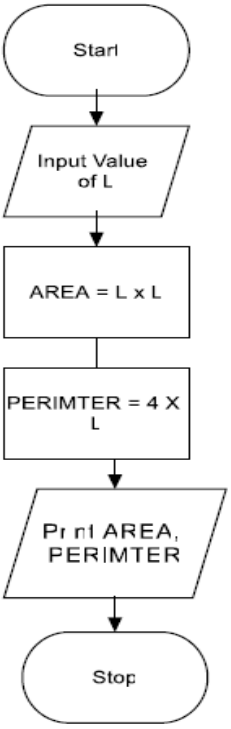
Step-2 Input Side Length of Square say L

Step-3 Area = L x L

Step-4 PERIMETER = 4 x L

Step-5 Display AREA, PERIMETER

Step-6 Stop



Algorithm & Flowchart to find Area and Perimeter of Circle

R : Radius of Circle
AREA : Area of Circle
PERIMETER : Perimeter of Circle

Algorithm

Step-1 Start

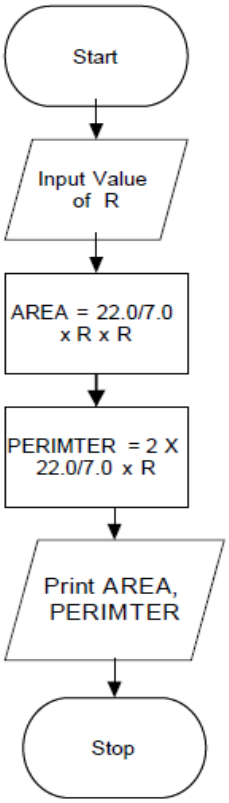
Step-2 Input Radius of Circle say R

Step-3 Area = 22.0/7.0 x R x R

Step-4 PERIMETER = 2 x 22.0/7.0 x R

Step-5 Display AREA, PERIMETER

Step-6 Stop



Algorithm & Flowchart to find Simple Interest

P : Principle Amount
N : Time in Years
R : % Annual Rate of Interest
SI : Simple Interest

Algorithm

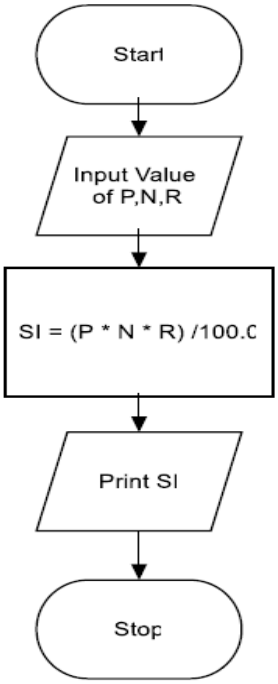
Step-1 Start

Step-2 Input value of P, N, R

Step-3 SI = (P x N x R)/100.0

Step-4 Display SI F

Step-6 Stop



Algorithm & Flowchart to find the smallest of two numbers

Algorithm

Step-1 Start

Step-2 Input two numbers say

NUM1,NUM2

Step-3 IF NUM1 < NUM2 THEN

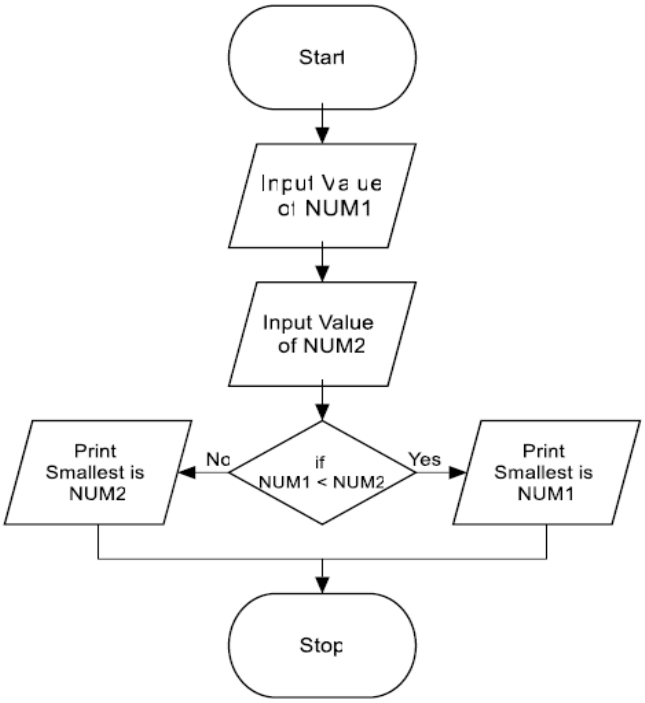
print smallest is NUM1

ELSE

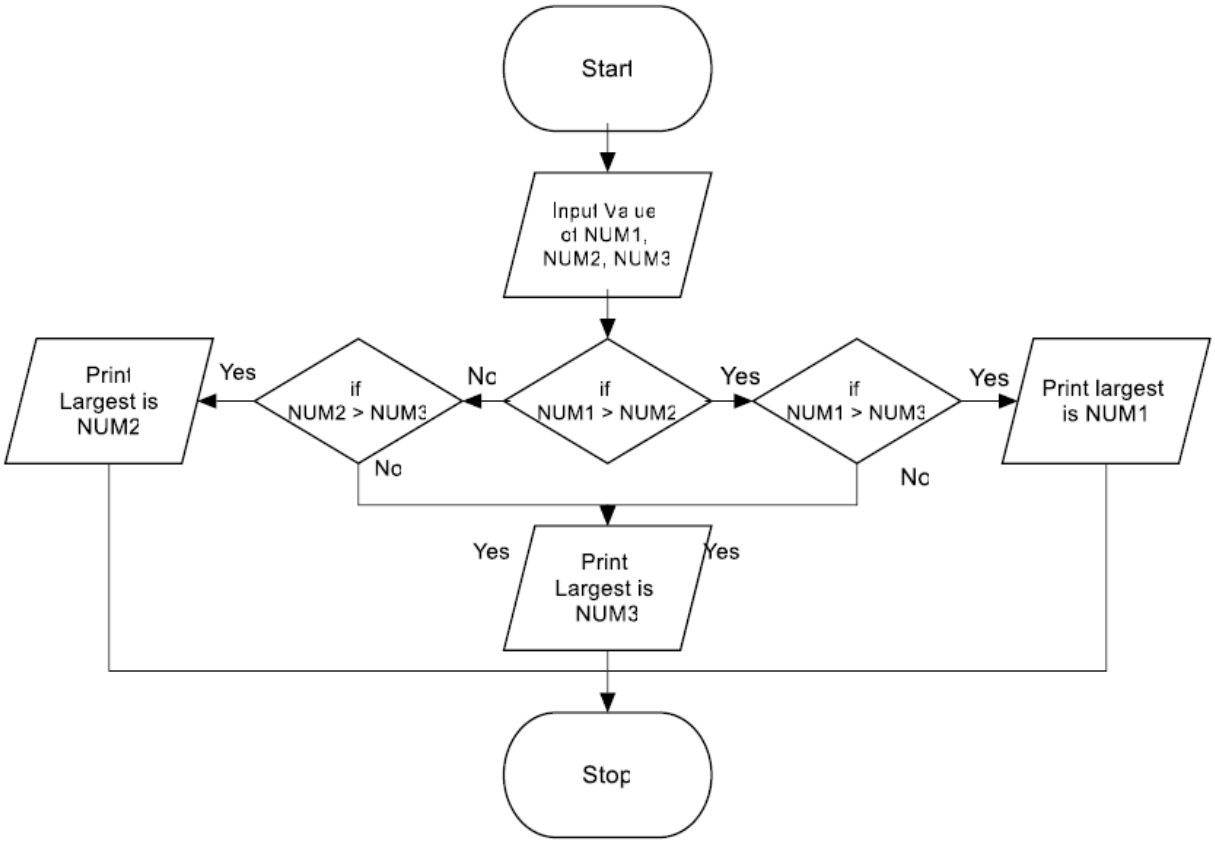
print smallest is NUM2

ENDIF

Step-4 Stop



Algorithm & Flowchart to find the largest of three numbers



```
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 sum of series 1+3+5+…..+N, Where N is positive odd Integer

Algorithm

Step-1 Start

Step-2 Input Value of N

Step-3 I = 1, SUM=0

Step-4 IF (I >N) THEN
GO TO step 8
ENDIF

Step-5 SUM = SUM + I

Step-6 I = I + 2

Step-7 Go to step-4

Step-8 Display value of SUM

Step-9 Stop

