Module Code: CSE401

Session 1a:Algorithm & flow chart

Session Speaker:

Prema Monish
premamonish.tlll@msruas.ac.in

Objectives

- At the end of this lecture, student will be able to
  - explain the process of arriving at a computer solution
  - describe the nature of a computer algorithm
  - discuss the role of memory in a computer program and hence the use of variable in algorithms

Contents

- A Problem
- Computable Algorithms
- Algorithms
- Problem Solving Approach
- Examples of Problem Solving

A Problem

Multiply

129837382

with

914147324

Given:

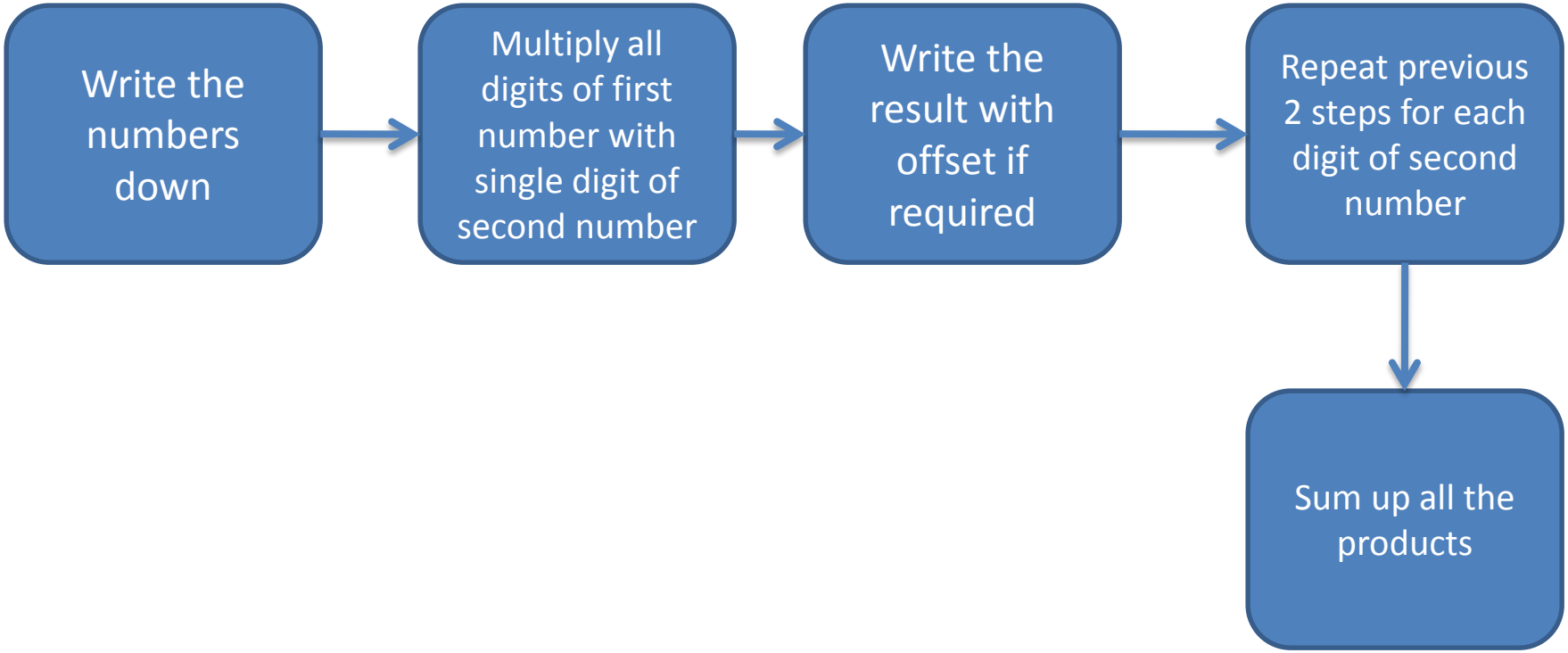
129837382

X 914147324

????????????

129837382
X 914147324
519349528
259674764
389512146
908861674
519349528
129837382
519349528
129837382
1168536438
11869049531046576 8

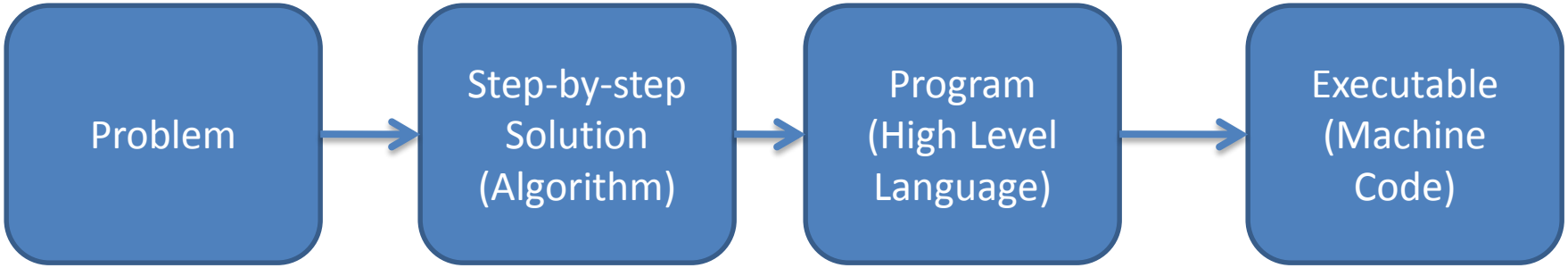
What Did You Do?



Why Do You Follow These Steps?

- General solution
  - Works for any 2 numbers
- Guarantee of solution
- Any one who has never done multiplication can also follow these steps and find the result
  - Clear, precise steps
- Computers do not know anything
  - You must tell the steps
  - It will do the job for you

Program Development



Example: Problem

Q. Design and Develop a program to find factorial of a number

$$n! = 1 \times 2 \times 3 \times \ldots \times (n-1) \times n$$

Example: Algorithm

```
function fact(var n:integer);
var     i { loop variable},
        ret{holds the return value}: integer;
begin {check n and calculate factorial}
        {assert : n>= 0}
        ret := 1;
        for i := 2  to n do
        begin
                ret: = ret * i;
        end
end
```

Example: C Program

```c
/*
 * File:   Factorial.c
 * Author: vsarma
 * Created on 18 July, 2014, 12:11 PM
 */

#include <stdio.h>
#include <stdlib.h>
#define NUMBER 4/*The number for calculating factorial*/
/* Function fact:
 * Calculates factorial of a given number.
 * Input: A positive Integer number
 * Output: The factorial of the number or -1 in case of an error
 */
int fact(int n){
    int i;
    int ret = 1;
    if(n<0){
        return -1;
    }
    for(i=2;i<=n;i++)
    {
        ret = ret * i;
    }
    return ret;
}
```

Example: Executable

```
00000000  7f 45 4c 46 01 01 01 00  00 00 00 00 00 00 00 00  |.ELF............|
00000010  02 00 03 00 01 00 00 00  30 83 04 08 34 00 00 00  |........0...4...|
00000020  b8 0b 00 00 00 00 00 00  34 00 20 00 08 00 28 00  |........4. ...(.|
00000030  25 00 22 00 06 00 00 00  34 00 00 00 34 80 04 08  |%."....4...4...|
00000040  34 80 04 08 00 01 00 00  00 01 00 00 05 00 00 00  |4...............|
00000050  04 00 00 00 03 00 00 00  34 01 00 00 34 81 04 08  |........4...4...|
00000060  34 81 04 08 13 00 00 00  13 00 00 00 04 00 00 00  |4...............|
00000070  01 00 00 00 01 00 00 00  00 00 00 00 00 80 04 08  |................|
00000080  00 80 04 08 dc 05 00 00  dc 05 00 00 05 00 00 00  |................|
00000090  00 10 00 00 01 00 00 00  dc 05 00 00 dc 95 04 08  |................|
000000a0  dc 95 04 08 20 01 00 00  24 01 00 00 06 00 00 00  |.... ...$.......|
000000b0  00 10 00 00 02 00 00 00  e8 05 00 00 e8 95 04 08  |................|
000000c0  e8 95 04 08 f0 00 00 00  f0 00 00 00 06 00 00 00  |................|
000000d0  04 00 00 00 04 00 00 00  48 01 00 00 48 81 04 08  |........H...H...|
000000e0  48 81 04 08 44 00 00 00  44 00 00 00 04 00 00 00  |H...D...D.......|
000000f0  04 00 00 00 50 e5 74 64  38 05 00 00 38 85 04 08  |....P.td8...8...|
00000100  38 85 04 08 24 00 00 00  24 00 00 00 04 00 00 00  |8...$...$.......|
00000110  04 00 00 00 51 e5 74 64  00 00 00 00 00 00 00 00  |....Q.td........|
00000120  00 00 00 00 00 00 00 00  00 00 00 00 06 00 00 00  |................|
00000130  04 00 00 00 2f 6c 69 62  2f 6c 64 2d 6c 69 6e 75  |..../lib/ld-linu|
00000140  78 2e 73 6f 2e 32 00 00  04 00 00 00 10 00 00 00  |x.so.2..........|
00000150  01 00 00 00 47 4e 55 00  00 00 00 00 02 00 00 00  |....GNU.........|
00000160  06 00 00 00 1a 00 00 00  04 00 00 00 14 00 00 00  |................|
00000170  03 00 00 00 47 4e 55 00  6f 05 cc 96 62 0c 6e 10  |....GNU.o...b.n.|
00000180  20 7f eb c3 d8 c0 3a 7d  f4 7d be b3 03 00 00 00  | .....:}.}......|
00000190  05 00 00 00 02 00 00 00  03 00 00 00 04 00 00 00  |................|
000001a0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  |................|
000001b0  01 00 00 00 02 00 00 00  04 00 00 00 01 00 00 00  |................|
000001c0  05 00 00 00 00 20 00 20  00 00 00 00 04 00 00 00  |..... . ........|
000001d0  ad 4b e3 c0 00 00 00 00  00 00 00 00 00 00 00 00  |.K..............|
000001e0  00 00 00 00 29 00 00 00  00 00 00 00 00 00 00 00  |....)...........|
000001f0  12 00 00 00 01 00 00 00  00 00 00 00 00 00 00 00  |................|
00000200  20 00 00 00 30 00 00 00  00 00 00 00 00 00 00 00  | ...0...........|
00000210  12 00 00 00 1a 00 00 00  1c 85 04 08 04 00 00 00  |................|
00000220  11 00 10 00 00 5f 5f 67  6d 6f 6e 5f 73 74 61 72  |....__gmon_star|
00000230  74 5f 5f 00 6c 69 62 63  2e 73 6f 2e 36 00 5f 49  |t__.libc.so.6._I|
00000240  4f 5f 73 74 64 69 6e 5f  75 73 65 64 00 70 72 69  |O_stdin_used.pri|
```

Algorithms and Programs

An Algorithm

- Set of *finite steps* that *give the solution* to a problem

- A procedure for solving a problem consists of
  - actions to be executed, and
  - order in which these actions are to be executed

A Computer Program

- It is the algorithm to solve a problem, expressed in a programming language

Characteristics of an Algorithm

- Input
  - may accept zero or more inputs
- Output
  - should produce at least one output
- Precise
  - each step should be clear and precise. No ambiguity
- Finiteness
  - should end after a fixed time. No infinite loop
- Effectiveness
  - steps must be simple and can be done exactly and in a finite length of time, by person using pencil and paper

Algorithms

- Swapping 2 numbers

```
Algorithm swap()
var temp, a, b : Integer;
begin
    readln(a);
    readln(b);
    writeln('The current values of a and b are', a, b);
    temp := a;
    a := b;
    b := temp;
    writeln('The current values of a and b are', a, b);
end
```

Algorithms

- Recap
  - Solution to a given problem
  - Finite
  - Step by step
  - Un-ambiguous

Algorithm <name of algorithm> (<parameter list>):<return type>

var <variable list>:<type>;{similar to parameter list}

begin

{<assertions>}

<instruction block with ; termination>

end

Algorithms

- I/O statement
  - readln(<variable name>)
  - writeln('<output string')

- Example
  - ***readln(****a****);***
  - ***writeln('****This is a line to be printed****');***

Flow Charts

- A flowchart
  - A graphical representation of an algorithm or of a portion of an algorithm

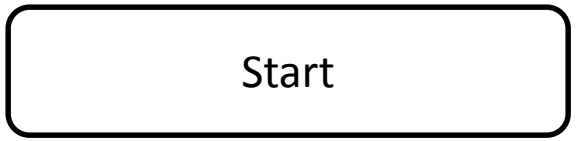
- Flowcharts are drawn using certain special-purpose symbols such as rectangles, diamonds, ovals, and small circles

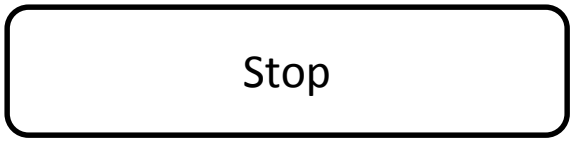
- Symbols are connected by arrows called flowlines

Flow Charts - Symbols

- Oval symbol
- Terminator symbol
  - All programs in C start executing from the first processing statement in main function
  - Flowcharts express the start of a program and termination of the program using a terminator symbol

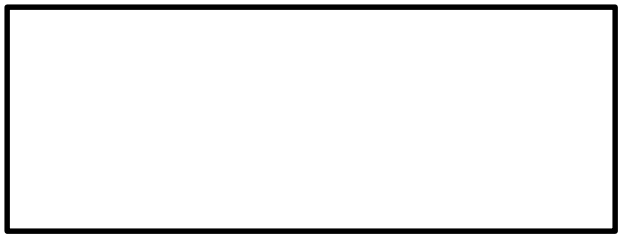
- Examples



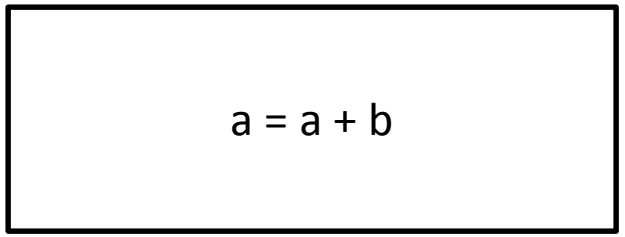


Flow Charts – Symbols contd.

- Rectangle symbol or action symbol
- Processing Statement



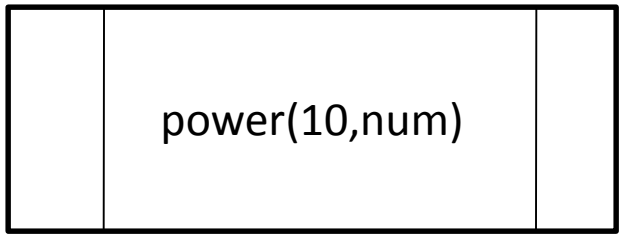
- Example



Flow Charts – Symbols contd.

- Predefined Process

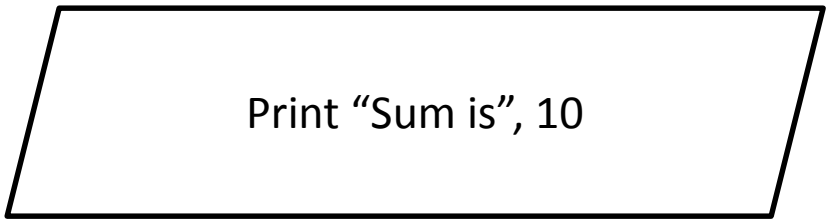
- Example

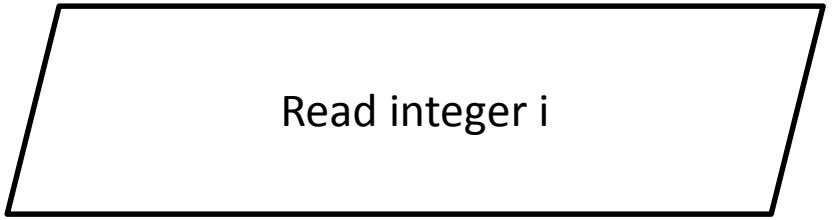


Flow Charts – Symbols contd.

- I/O statement

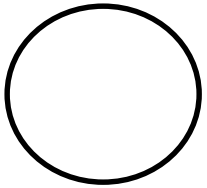
- Examples



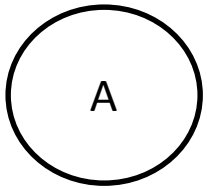


Flow Charts – Symbols contd.

- Connectors

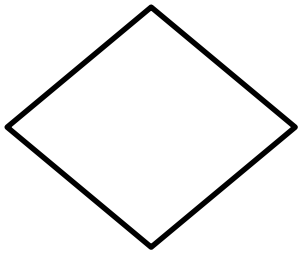


- Examples

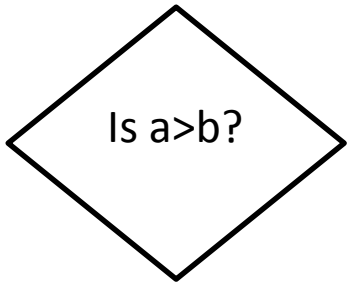


Flow Charts – Symbols contd.

- Diamond symbol or decision symbol – indicates that a decision is to be made
- Contains an expression, such as a condition, that can be either true or false

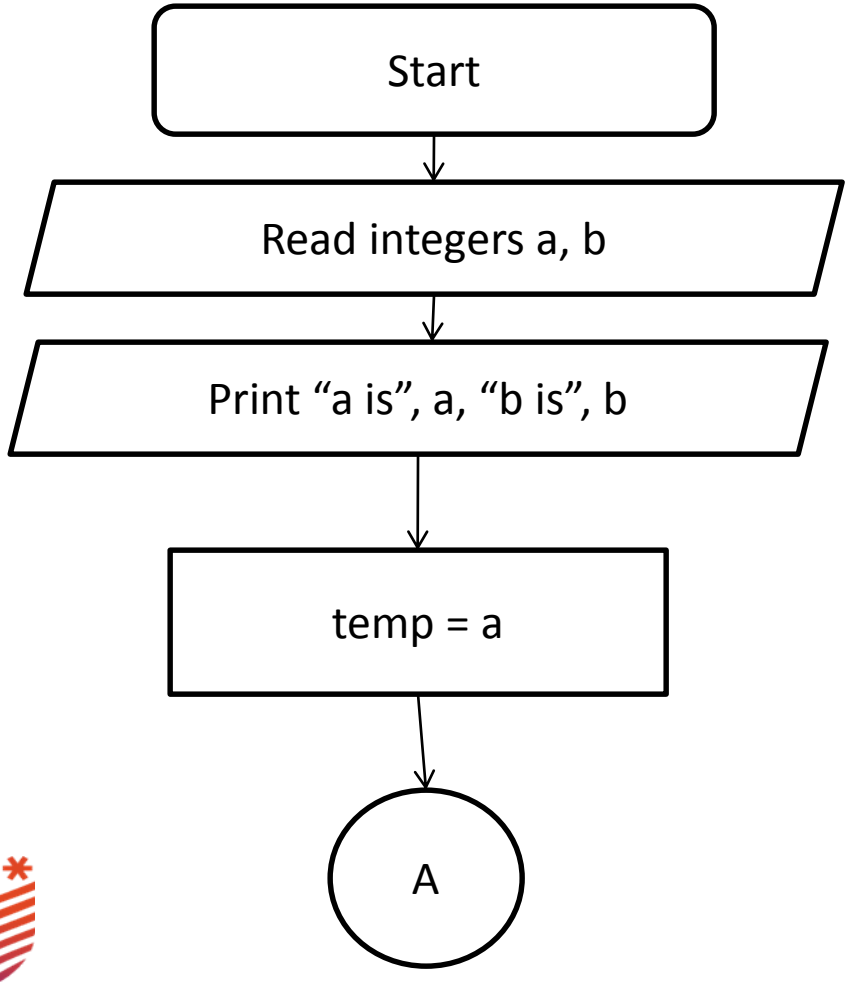


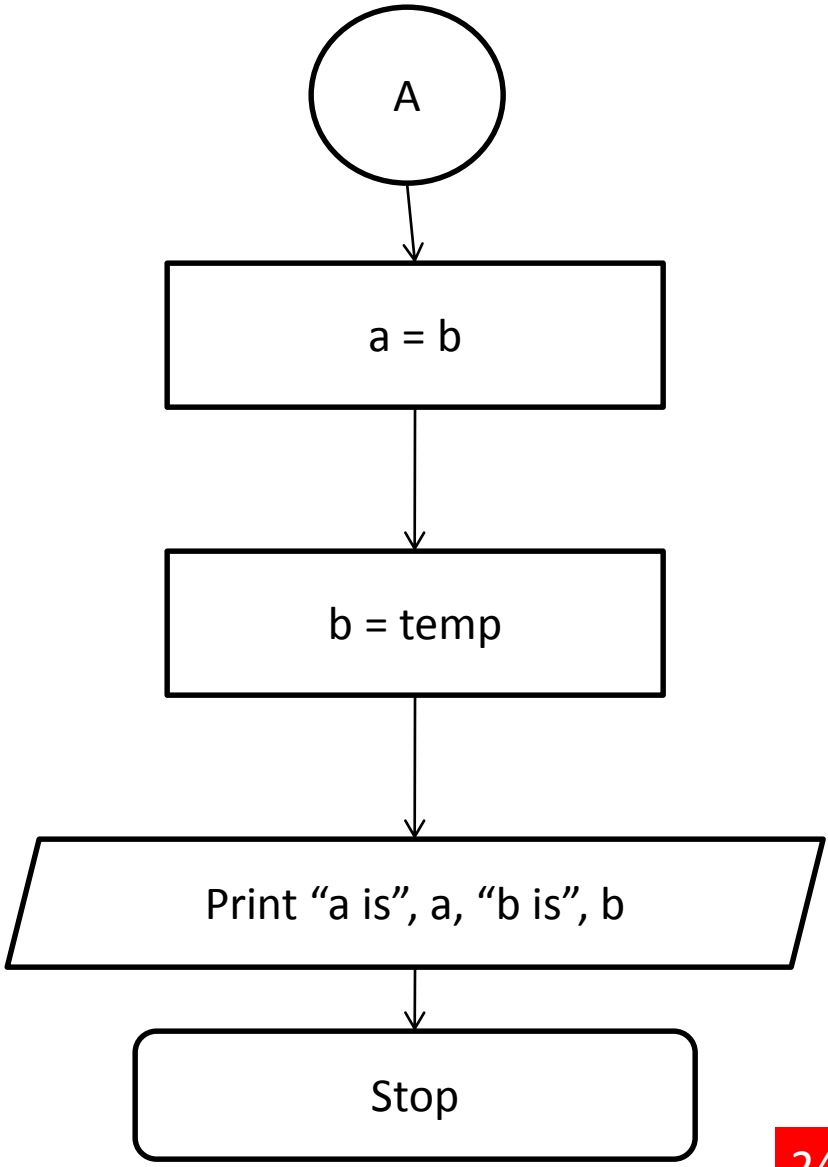
- Examples



Flow Charts

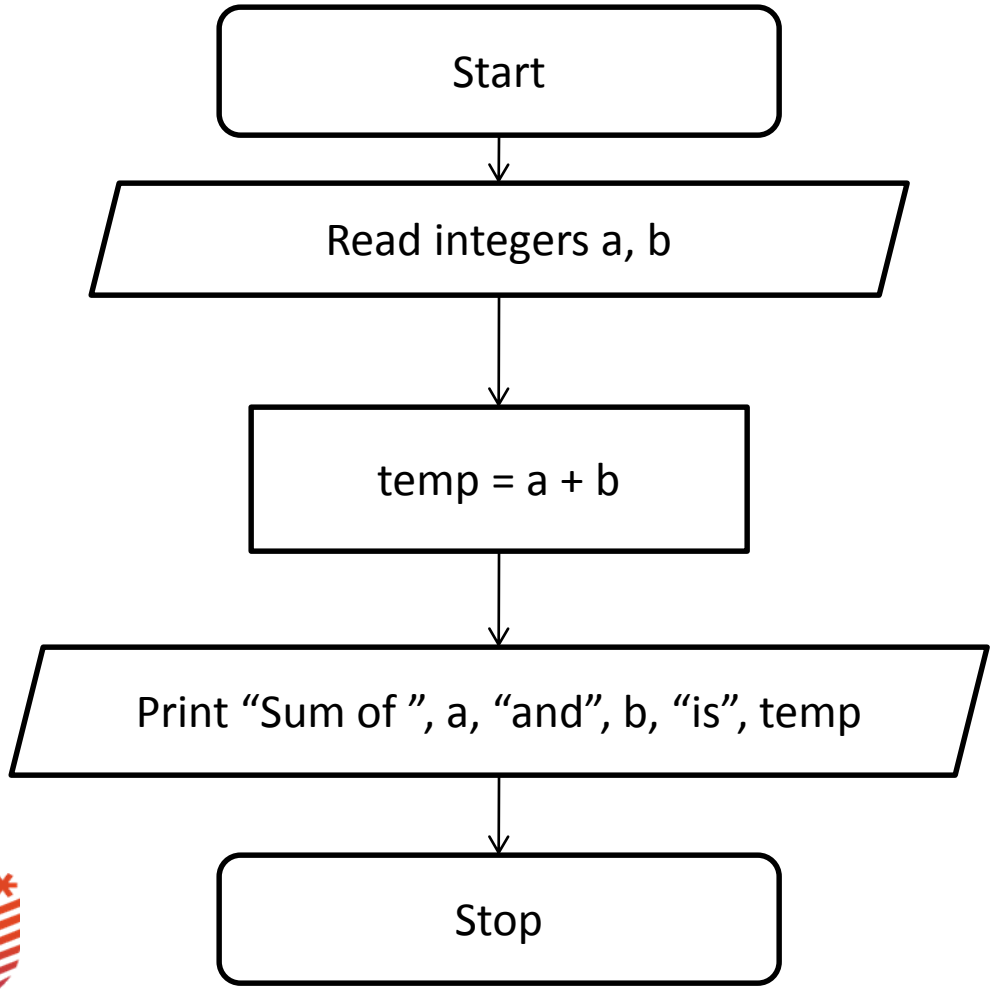
- Swapping 2 numbers





Flow Charts

- adding 2 numbers



Summary

- An algorithm is a set of explicit and unambiguous finite steps which when carried out for a given set of initial conditions, produce the corresponding output and terminate in a finite time
- Computer solutions are called programs
- A Program is a set of explicit and unambiguous instructions expressed in a programming language
- Programming is application of problem solving using a computer
- Flow Charts are graphical representation of Algorithms and clearly show the control flow

ANY QUERIES