Programming in C

Chapter -1

Introduction to Computer Programming

by, Santa Basnet

Introduction to PROGRAMMING

A computer, what is ?

A computer is a device that computes something over a sequence of finite steps.

Introduction to PROGRAMMING

A computer, what is ?

A general purpose computer allows you to alter the sequences of finite steps to solve more than one problems.

Introduction to PROGRAMMING

An Electronic computer

- An electronic computer have at-least the following components to solve the problems.
  1) CPU [ALU and CU]
  2) Memory (Main, Auxiliary)
  3) Input and Output

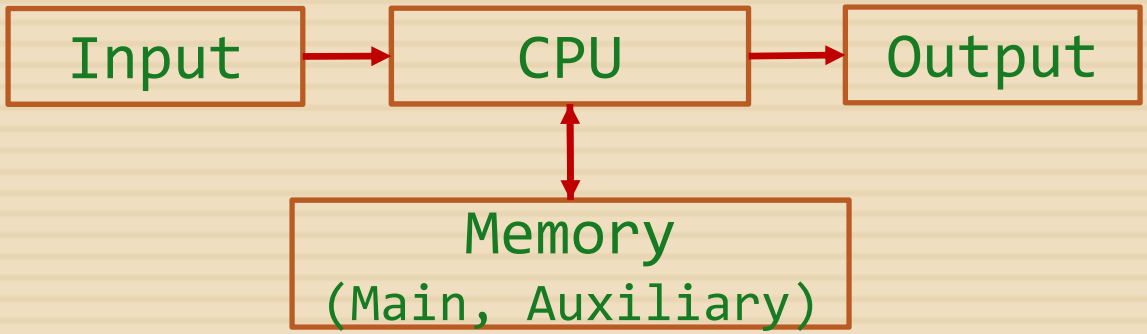


Fig. Block view of a computer.

Introduction to PROGRAMMING

Programming History

- Early and 40's, the physical machine and the code
- 50's, Assembly and Macros
- 60's, Beginning of structured style language
- 70's, Structured/Procedural language
- 80's, Object oriented, Query and Logic programming
- 90's, Scripting language and Internet
- 00's, Functional and OO language together

Introduction to PROGRAMMING

Programming History

```
    MOV     C,A
    MVI     A,00
L:  ADD     B
    DCR     C
    JNZ     L
    STA     2010
    HLT
```

```
&INIT   MACRO
        INCRX   &AMT
&CNT    SET     &AMT
        WHILE   (&CNT GT 0)
        INCRX   X
CNT     SET     &CNT - 1
        ENDW
        MEND
```

```
[ op   |  rs  |  rt  |  rd  |shamt| funct]
  0       1       1       6       0      32      decimal
000000  00001  00010  00110  00000  100000      binary
```

```
int **x;
int size = 4, i, j;
int memsize = 4;
x = (int*)malloc(memsize*(size));
for(i = 0; i < size; i++)
{
    *(x+i) = (int*)malloc(memsize*(size));
    for(j = 0; j < size; j++){
        *(*(x+i)+j) = i+j;
    }
}
```

```
friend void readInput(Complex &C1, Complex &C2);
friend istream& operator>>(istream& s, Complex& c)
friend ostream& operator<<(ostream& s, Complex& c)
void operator++(int){
   this->real ++; this->imaginary ++;
```

```
00000001
00000010
00000100
00001000
00010000
00100000
01000000
```

```
$this->dbms = $dbms;
if ($thisConnection) {
    $this->host = $thisConnection->host;
    $this->database = $thisConnection->database;
}
$this->fn = $fn;
$this->msg = $errmsg;
```

```
template <typename Collection,typename Predicate>
Collection filterNot(Collection col,Predicate predicate ) {
auto returnIterator = remove_if(col.begin(),col.end(),predicate);
col.erase(returnIterator,std::end(col));
return col;
```

Introduction to PROGRAMMING

Text Editor and Files

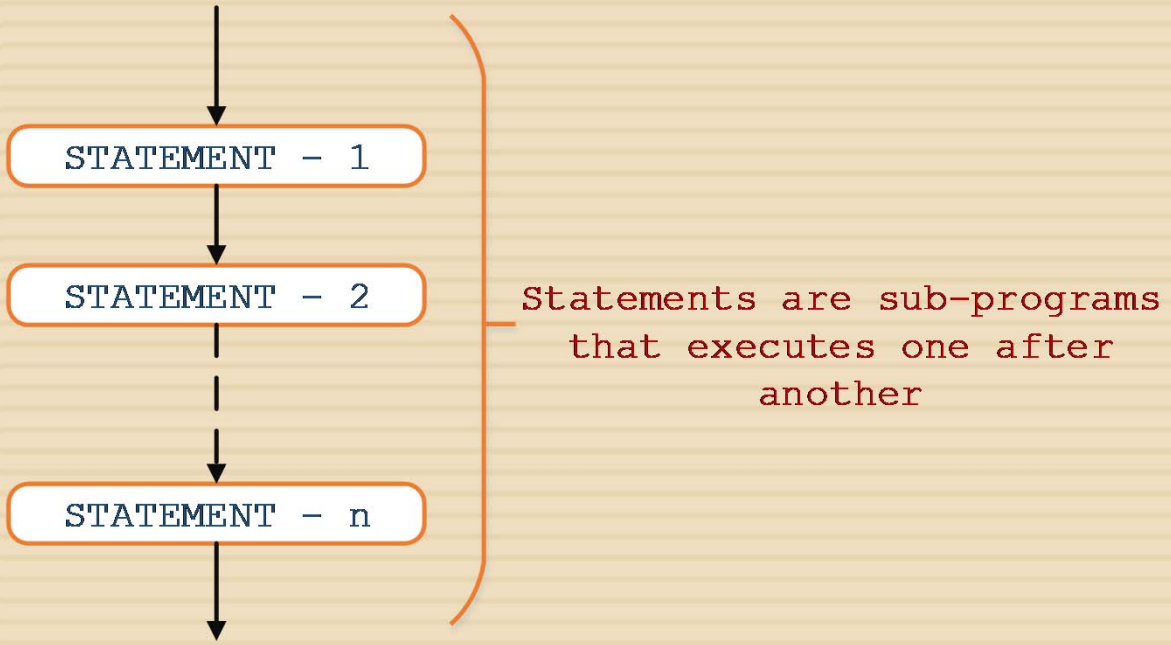
- With support to Operating system, we store all data in various file formats. Example: photo, video, music, text etc.
- We store high level source programs in text format and edit/save using text editors.
- Some text editors: notepad++, gedit, Vi, Emacs etc.
- An IDE provides facilities in source code editing, build automation and debugging with code completion feature for a source program. Example: Visual Studio, NetBeans, Code::Blocks etc.

Introduction to PROGRAMMING

Structured Programming

- We express the programming problem/writing software through 3 basic structures:

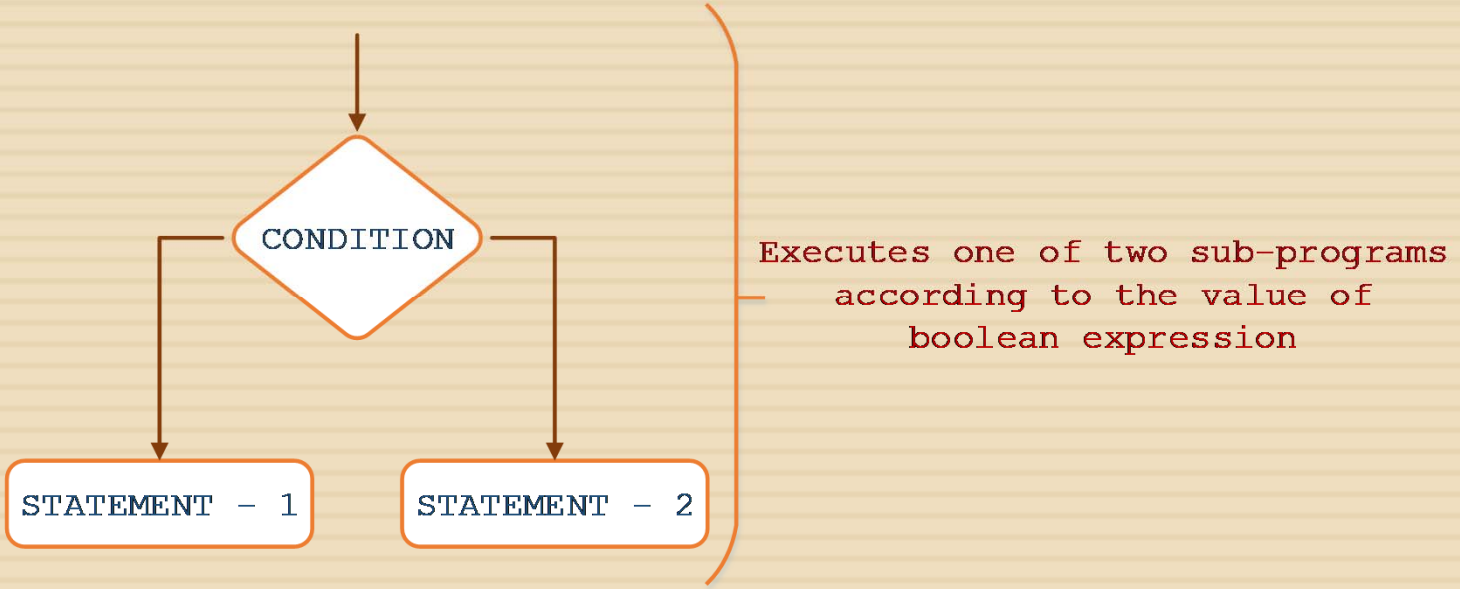
1) Sequences



Introduction to PROGRAMMING

Structured Programming

- We express the programming problem/writing software through 3 basic structures:
  2) Selections

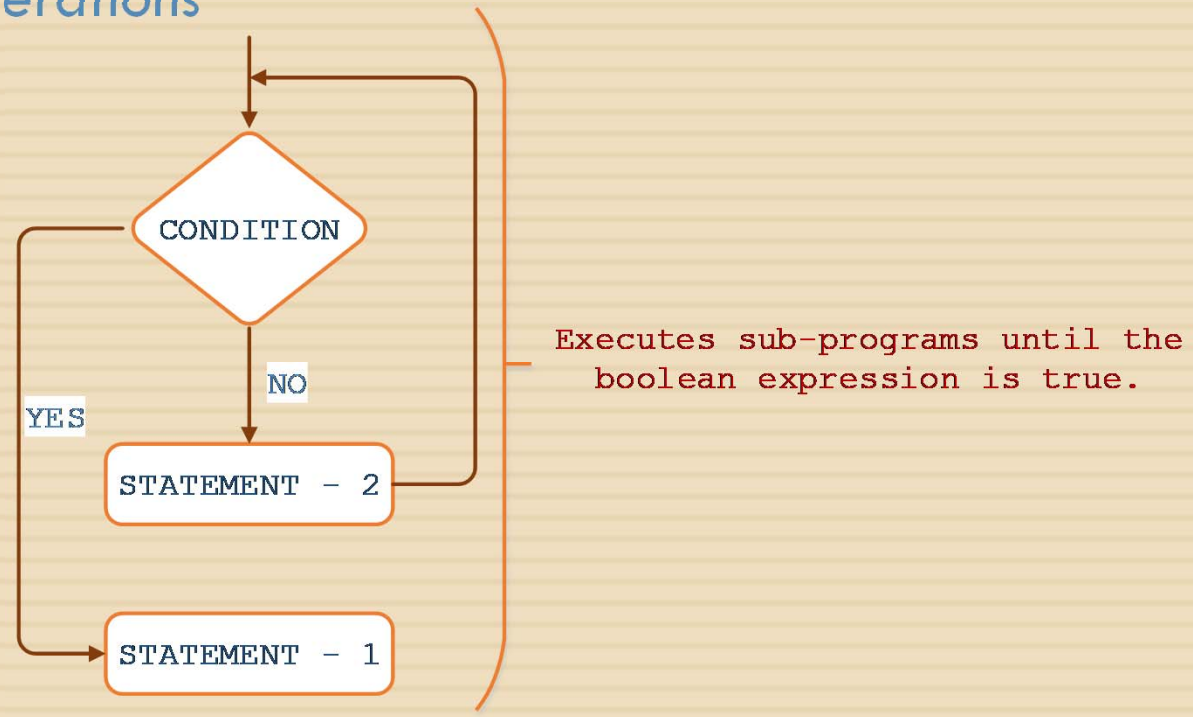


Introduction to PROGRAMMING

Structured Programming

- We express the programming problem/writing software through 3 basic structures:

3) Iterations



Introduction to PROGRAMMING

Flow Chart and Algorithm

- An Algorithm is a finite set of rules that defines a sequence (step by step) of operations that a computer need to perform.
- We write a computer program after defining an algorithm for it, otherwise we will be mislaid while writing a program.

Example:

```
Algorithm to find Largest Number in a list.
  /*Input: a list of numbers.
    Output: The largest number in the list.*/
if size_of_list = 0 then return null.
Assign Largest <- List[0]
for each Element in List, do
        if Element is_greater_than Largest then
                Assign Largest <- Element
print Largest
```

Introduction to PROGRAMMING

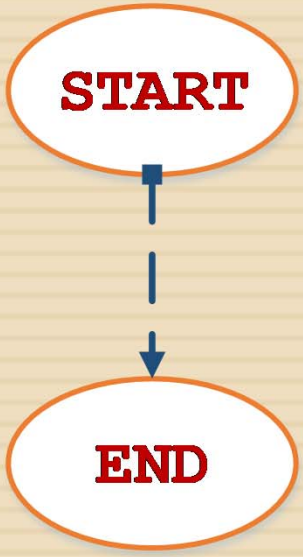
Flowchart and Algorithm

- A flowchart is a pictorial representation of the algorithm.
- A flowchart comprises different symbols that have specific meaning in representing the control flow of a computer program.
- A flowchart is a solution model to a problem.

Introduction to PROGRAMMING

Flowchart and Algorithm

- A flowchart Symbol: START and END

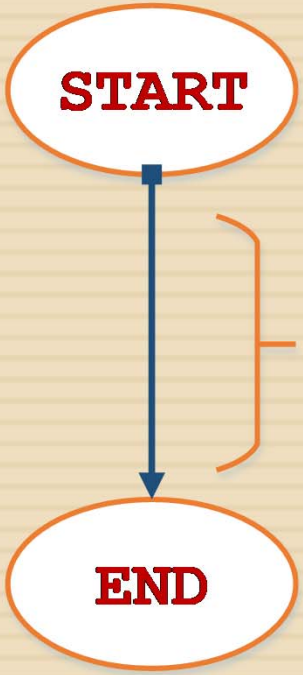


A circle or oval is used to denote the program START and END.

Introduction to PROGRAMMING

Flowchart and Algorithm

- A flowchart Symbol: ARROW

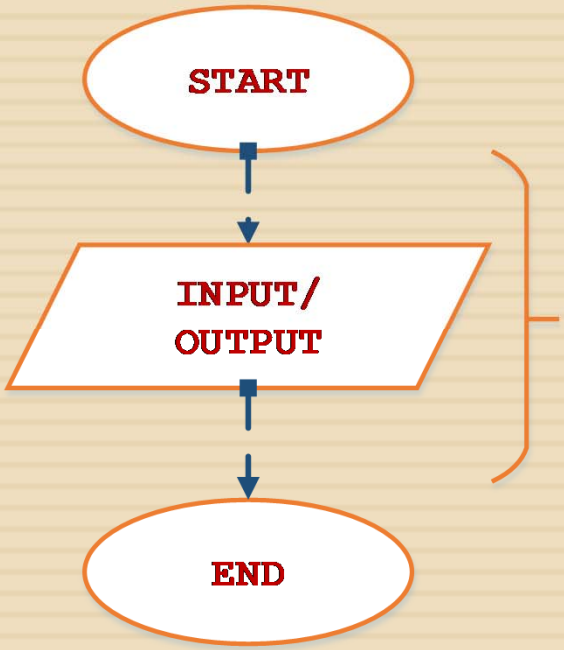


An arrow denotes the direction of program flow.

Introduction to PROGRAMMING

Flowchart and Algorithm

- A flowchart Symbol: PARALLELOGRAM



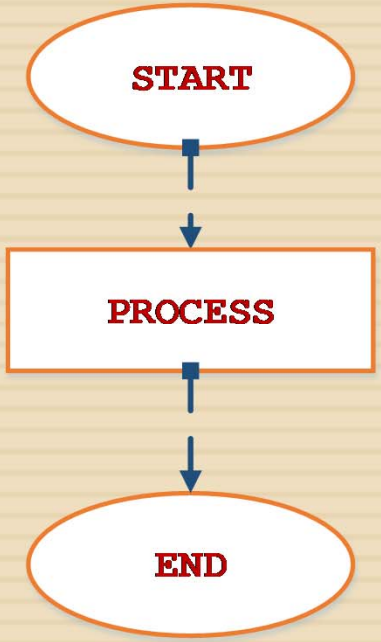
A parallelogram is used for input and output of the program.

Example: Get student's age,
Print weight etc.

Introduction to PROGRAMMING

Flowchart and Algorithm

- A flowchart Symbol: RECTANGLE



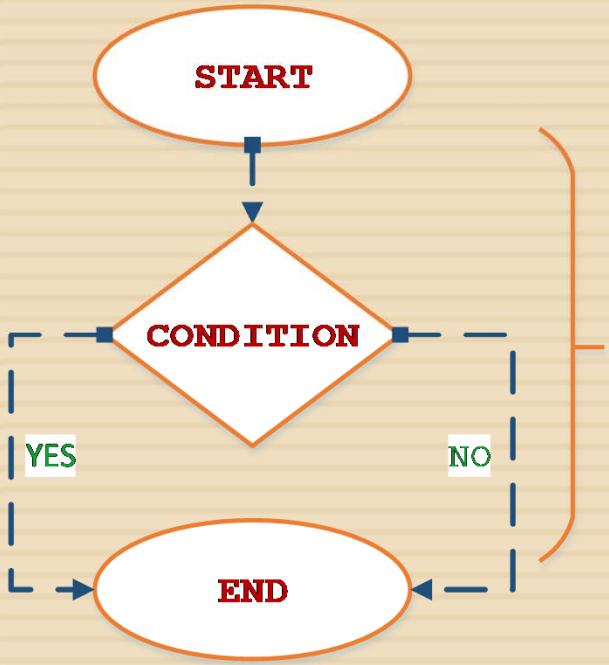
A rectangle is used to represent the processing step of the program.

Example: Add 1 to age,
Initialize 0 to weight etc.

Introduction to PROGRAMMING

Flowchart and Algorithm

- A flowchart Symbol: DIAMOND



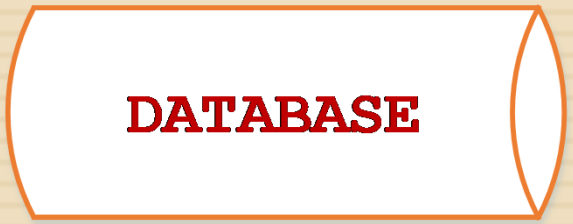
A diamond is used to represent the conditional step of the program that allows to choose either one of the control flow.

```
Example: IF age > 25 then
             goto ADULT
         ELSE
             goto TEEN.
```

Introduction to PROGRAMMING

Flowchart and Algorithm

- A flowchart Symbol: SUB-PROCESS, DOCUMENTS AND DATABASE.



Introduction to PROGRAMMING

Flowchart and Algorithm

- A flowchart Symbols:

Bringing all together

Class Work:

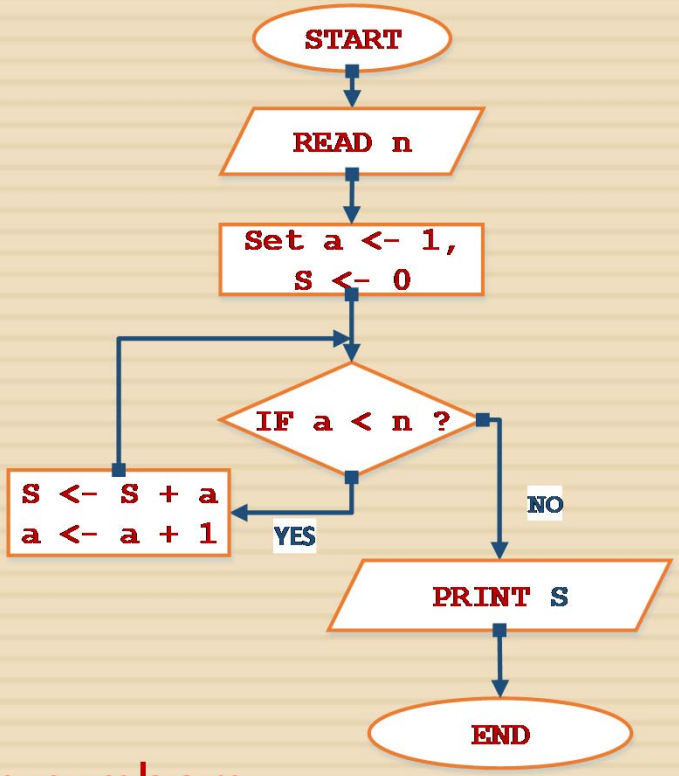
Flowchart to find the sum of first n numbers.

Introduction to PROGRAMMING

Flowchart and Algorithm

- A flowchart Symbols:

Bringing all together



Class Work:

Flowchart to find the sum of first n numbers.

Introduction to PROGRAMMING

Flowchart and Algorithm

- A flowchart Symbols: Bringing all together

Class Work:

Flowchart to find the largest number in a list of n numbers.

Introduction to PROGRAMMING

Program Documentation

- Specification & Analysis: variables, nature of the variables and the qualities.
- Design: Coding principles and relationship to the environment.
- Implementation (technical):
  - Mathematics of the program model.
  - An Algorithm and a flowchart.
  - Writing program code.
  - Debugging and Evaluation.
- User Guide

Introduction to PROGRAMMING

Program Documentation

- An Example: a program that prints the square-root of first N natural numbers.

Specification:

1. N numbers of Integer type.
2. ROOTS are the numbers of real(float) type.
3. COUNT that iterates from 0 to N.

Introduction to PROGRAMMING

Program Documentation

- An Example: a program that prints the square-root of first N natural numbers.

Design: we use structured programming principle to complete this job with two sub-processes.

1. ITERATE up to first N natural numbers.
2. FIND_ROOT to calculate the square root of a number.

Introduction to PROGRAMMING

Program Documentation

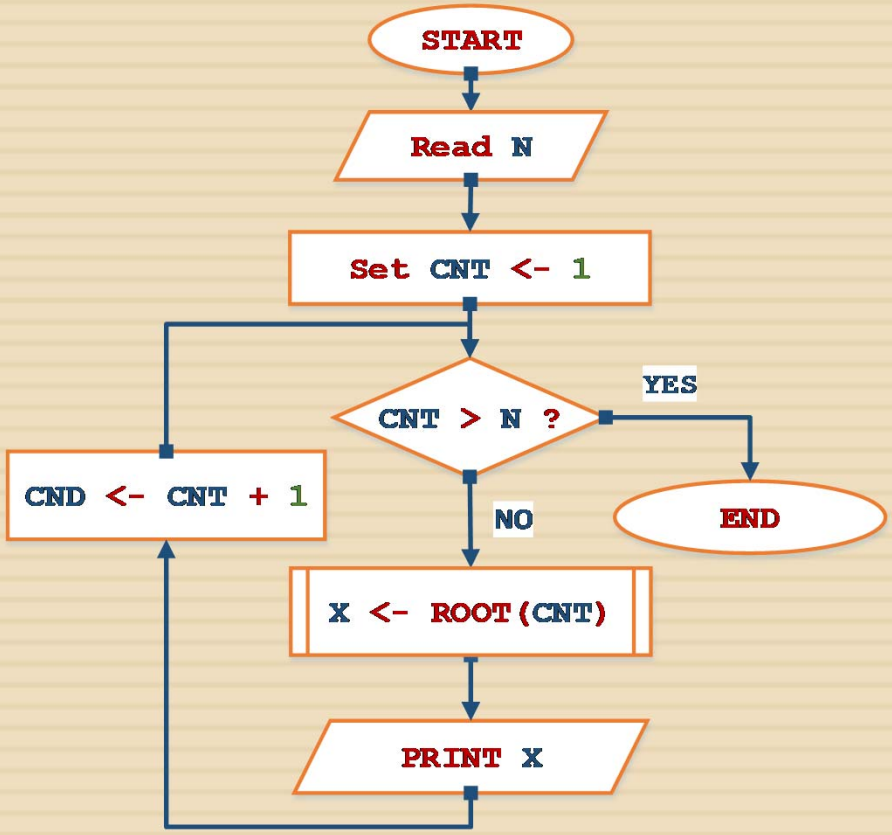
- An Example: a program that prints the square-root of first N natural numbers.

Implementation:

1. Mathematics:

$$X = \sqrt{Y}$$

2. Flowchart and Algorithm.



Introduction to PROGRAMMING

Program Documentation

- An Example: a program that prints the square-root of first N natural numbers.

Implementation:

3. Writing program code:

```
/*Print square root of first n natural numbers*/
#include <stdio.h>
#include <math.h>
float get_root(int x){return(sqrt(x));}
int main()
{
    int n, iter;
    printf("First n = "); scanf("%d", &n);
    for(iter = 1; iter <= n; iter++)
        printf("%4d --> %.2f\n", iter, get_root(iter));
    return(0);
}
```

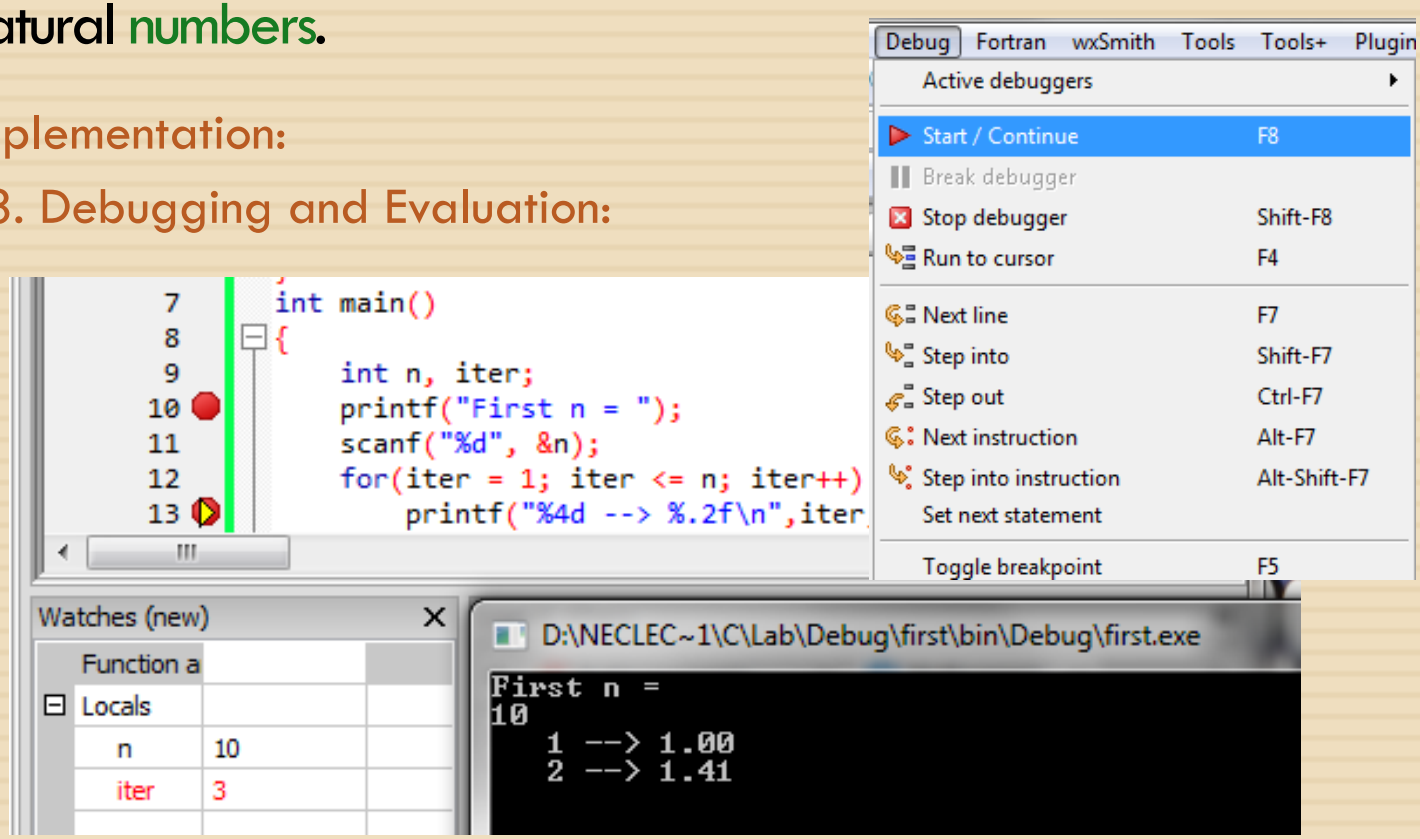
Introduction to PROGRAMMING

Program Documentation

- An Example: a program that prints the square-root of first N natural numbers.

Implementation:

3. Debugging and Evaluation:



Introduction to PROGRAMMING

Program Documentation

- An Example: a program that prints the square-root of first N natural numbers.

Implementation:

3. Debugging and Evaluation:

N	Program Output	Actual Output	Result
1	1.00000	1.00	Accepted
2	1.41421	1.41	Accepted
3	1.73205	1.73	Accepted
4	2.00000	2.00	Accepted
5	2.23607	2.24	Accepted
6	2.44949	2.45	Accepted
7	2.64575	2.65	Accepted
8	2.82843	2.83	Accepted
9	3.00000	3.00	Accepted
10	3.16228	3.16	Accepted

Introduction to PROGRAMMING

Program Documentation

- An Example: a program that prints the square-root of first N natural numbers.

User Guide:

1. Input: Put an **integer number** N when it asks.
2. It will display the list of N square roots. Sample I/O:

```
Input N :  10
Output:
          1 --> 1.000000
          2 --> 1.414214
          3 --> 1.732051
          4 --> 2.000000
          5 --> 2.236068
          6 --> 2.449490
          7 --> 2.645751
          8 --> 2.828427
          9 --> 3.000000
         10 --> 3.162278
```

Thank you.

Questions ?