Data Structure

Chapter one

Program and Data Structure

- **Program:** An implementation of an algorithm in some programming language
- **Data:** Set of possible values for variables, characters, numbers, pictures, videos or mixed.

EX :int ,float,char

Processing Type:

1. Insert
2. Delete
3. Marge
4. Sort
5. Electronic copy
6. Security
7. Edit and retrieve

Data structure

- A collection of data elements whose organization is characterized by accessing(Mechanism for organizing information)
- operation that are used to store and retrieve the individual of elements

Types of data structure:

1. *LINEAR STRUTURES*
   - ARRAY
   - RECORD
   - STACK
   - QUEUE
   - CIRCULAR QUEUE
   - LINKED list
   - LINKED STACK
2. *NON-LINEAR STRUCTURES*

- GRAPHS
- TREE STRUCTURE

How can choose the principle data structure

For each set of data there are more than one method to organization depend on:

1. Data size
2. Required storage space
3. The time required to retrieve the data
4. Programing language manner

Array

- Fundamental data structure
- Homogeneous collection of values (all the same type)
- Store values sequentially in memory
- Associate INDEX with each value
- Use array name and index to quickly access k_{th} for any k

Memory address	Index	Content
100	0	h
101	1	e
102	2	l
103	3	L
104	4	o

One-Dimensional Array

- General array declaration statement:

Data-type array-name [number-of-items];

Ex: int mm [10];

- The number-of-items must be specified before declaring the array.

  const int SIZE = 100;

  float arr [SIZE];

- Individual elements of the array can be accessed by specifying the name of the array and the *index* of element

  arr [3]

Warning: indices assume values from 0 to *number-of-items -1*

- *Important*: this is essentially "call by reference":

  a) The name of the array *arr* stores the address of the first element of the array *arr [0]* (i.e., *&arr [0]*).

  b) Every other element of the array can be accessed by using its index as an offset from the first element.

arr[0]	arr[1]	arr[2]	arr[3]	arr[4]

*Physical Location (X [I]) = Base address + (I*Size)*

When (I) is a logical address.

Ex: -Let int x [50];

What is the address of the element x [33] if the base address (BA=970)

Sol: Physical Location (X [I]) = Base address + (I*Size)

= 970 + (33 * 2)

= 970 + 66

= 1036

When we have two domination arrays there are two methods to fine the physical address:

(Row_wise method) and (column _wise method)