

Data Structure

هيكل البيانات

المحاضرة الاولى

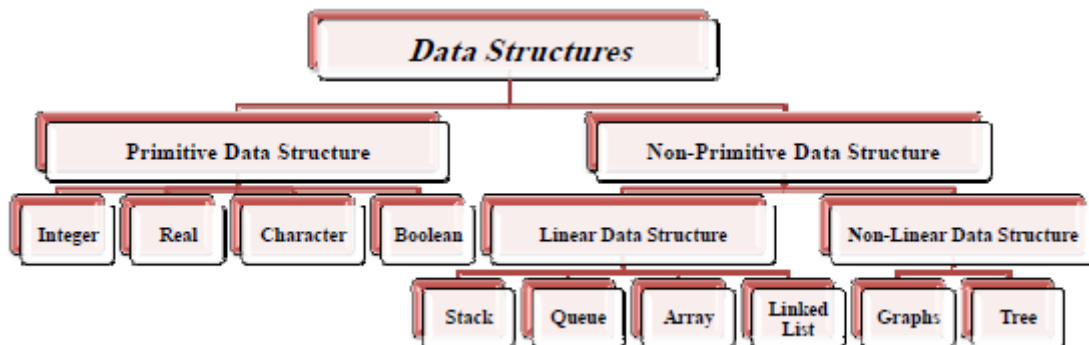
An Introduction to Data Structure

Data Structure Definition:

- In computer science, a data structure is a particular way of storing and organizing data in a computer's memory (or sometimes on a disk) so that it can be used efficiently.

Classification of Data Structures:

- Data Structure are generally classified into primitive and nonprimitive data structure. The primitive data type consist of characters that cannot be divided such as (integer, real, character and Boolean). They are also called simple data type. The figure shows the classification of data structures.



Non-Primitive Data Structure

Linear Data Structure:

- A data structure is said to be linear if its elements form a sequence, or, in other words, a linear list. There are two basic ways of representing such linear structures in memory. these linear structures are called **Arrays** and **Linked List**.

The operations that normally performs on any linear structure, whether it be an **array** or a **linked list**, include the following:

1. **Traversing:** means to visit all the elements of the array in an operation is called traversing.
2. **Sorting:** Re-arrangement of values in a list in a specific order
3. **Searching:** The process of finding the location of a particular element in a list is called searching.
4. **Insertion:** Adding a new element to the list.

5. **Deletion:** Removing an element from the list.
6. **Merging:** Combining two lists into a single list

Linear Data Structure:

1- Array:

Array classified in C++ in three category:

1. One – Dimensional Arrays.
2. Two – Dimension Arrays.
3. Multidimensional Arrays.

One – Dimensional Arrays

- A one dimensional array is used when it is necessary to Keep a large number of items in memory and reference all the items in uniform manner.
- **Int N[100];**
- That is mean we reserves **100** successive memory locations and each location is large enough to conation single integer. In C++, the array element indices are 0-99.



- The address of the first location is called the **base address** of the array and is denoted by base (**BA**) and the rest of the array elements come after this address.
- Computer does not need to keep track of the address of every array element, but need to track only the address of the first element of the array **Base Address (BA)** and to reach to any array element and the compiler use the following formula to do so.

$$\text{Loc (N [I])} = \text{BA} + (\text{I}) \times \text{Size}$$

- **Loc N[I]** : The location of the element **I**, **BA**: Fixed base address, **Size**: A fixed constant, is also known as size of the data type.

Example : Consider an one dimension array (N) with size 10 and the base address equal (3002) and each element of the array occupy 1 byte. find the address the element number six.

$$\text{Loc (N [I])} = \text{BA} + (\text{I}) \times \text{Size}$$

$$\text{then, Loc (N [5])} = 3002 + (5) \times 1$$

$$\text{Loc (N [5])} = 3007.$$

Logical address	Physical address	Memory
N[0]	3002	
N[1]	3003	
N[2]	3004	
N[3]	3005	
N[4]	3006	
N[5]	3007	
:	:	
N[9]	3011	
Programmer view	Compiler view	

الصيغة العامة للإعلان عن المصفوفة ذات البعد الواحد:

data type array name [index] ;

حيث:-

data type : نوع بيانات المصفوفة

array name : اسم المصفوفة ويراعى فيه شروط تسمية المتغيرات.

index : دليل المصفوفة (عدد عناصرها) وهي عبارة عن قيمة صحيحة يمكن أن تكون محدده (ثابتة) أو مدخلة من قبل المستخدم (متغيرة.)

أمثلة:

1. int x [50]; إعلان عن مصفوفة حجمها 50 وبياناتها من النوع الصحيح
2. float y [20]; إعلان عن مصفوفة حجمها 20 وبياناتها من النوع الحقيقي
3. char name [15]; إعلان عن مصفوفة حجمها 15 وبياناتها من النوع الحرفي
4. int x [3] = { 5,10,15 }; إعلان عن مصفوفة حجمها 3 وبياناتها من النوع الصحيح مع إعطائها قيم ابتدائية
5. char y [4] = { 'a','b','c','d' } ; إعلان عن مصفوفة حجمها 4 وبياناتها من النوع الحرفي مع إعطائها قيم ابتدائية

Program -1-: Write C++ program to read one-dimensional array consisting of 10 elements, then give the summation for these elements and print it :

```
#include <iostream>
using namespace std;

int main()
{
    int a[10];
    int sum=0, i;
    /* ----READ ARRAY-----*/
    for(i=0;i<10;i++)
    {
        cout<<"A["<<i<<"]=";
        cin>>a[i];
    }

    for(i=0;i<10;i++)
        sum +=a[i];

    cout<< "sum=" << sum <<"\n";
    return 0;
}
```

Program-2: Write a program to read one-dimensional array has 5 real values and order them in ascending form and print them (ترتيب تصاعدي)?

```
#include <iostream>
using namespace std;
const int n=5;

int main( )
{
    float x[n],temp;
    int i,j;
    /*--Read Array Before Sorting--*/
    for(i=0;i<n;i++)
    {
        cout<<"X["<<i<<"]=";
```

```

        cin>>x[i];
    }

    /*-- Sort Array --*/
    for(i=0;i<n-1;i++)
        for(j=i+1;j<n;j++)
        {
            if(x[i]>x[j])
            {
                temp=x[i];
                x[i]=x[j];
                x[j]=temp;
            }
        }

    cout<<" Print array after sorting\n";
    for(i=0;i<n;i++)
        cout<<x[i]<<"\n";
    return 0;
}

```

Two – Dimensional Arrays

- **2D array** is a data structure type that consists of a set of elements that are all of the same type, and all elements are distributed on a set of rows and columns that represent the size of the array.
- **Int N[3][5];** That is mean we reserves (3*5=15) successive memory locations and each location is large enough to contain single integer. The number of Rows or Columns is called the range of the dimension.
- The array N will be represented in the memory by block of (3 × 5) sequential memory location. Programming language will store array N either :
 1. **Column by Column:** called (Column-Major Order) Ex: Fortran, Matlab.
 2. **Row by Row:** called (Row-Major Order) Ex: C, C++, Java.

Column-Major Order :

- **Loc (N [I][J]) = BA + (m × J + I) × Size**
- **Where:**
- **Loc N[I][J] :** the location of the element I,J.
- **BA:** fixed base address.
- **m:** number of row.
- **Size:** a fixed constant, is also known as size of the data type

Example : Column-Major Order

Consider a two dimension array (N) with size (m=3 × n= 5) and the base address equal (300) , each element of the array occupy 1 byte. find the address to element N[1][2]. Suppose the programming store 2D using Column-Major:

$$\text{Loc (N [I][J])} = \text{BA} + (\text{m} \times \text{J} + \text{I}) \times \text{Size}$$

$$\text{Loc (N [1][2])} = 300 + (3 \times 2 + 1) \times 1$$

$$\text{Loc (N [I][J])} = 307$$

Row-Major Order :

- $\text{Loc (N [I][J])} = \text{BA} + (\text{n} \times \text{I} + \text{J}) \times \text{Size}$

- **Where:**

- **Loc N[I][J] :** the location of the element I,J.

- **BA:** fixed base address.

- **n:** number of column.

- **Size:** a fixed constant, is also known as size of the data type

Example : Consider a two dimension array (N) with size (m=3 × n= 5) and the base address equal (600) and each element of the array occupy 1 byte. find the address to the element N[2][3]. Suppose the programming store 2D using Row-Major.

$$\text{Loc (N [I][J])} = \text{BA} + (\text{n} \times \text{I} + \text{J}) \times \text{Size}$$

$$\text{Loc (N [2][3])} = 600 + (5 \times 2 + 3) \times 1$$

$$\text{Loc (N [I][J])} = 613$$

الصيغة العامة للإعلان عن المصفوفة ذات البعدين:

data type array name [row size] [column size];

حجم الأعمدة حجم الصفوف اسم المصفوفة نوع المصفوفة

أمثلة:

1. int y[4][3] إعلان عن المصفوفة الصحيحة y كمصفوفة ذات بعدين وتتكون من 4 صفوف و 3 اعمدة

2. y[4][3]=23; إسناد القيمة 23 لعنصر الصف الرابع والعمود الثالث من المصفوفة y

3. int y[4][3]={5,0,-4},{-2,3,1},{4,7,6},{9,8,-1}}; إسناد قيم ابتدائية لعناصر المصفوفة y اثناء التصريح

Program-3: Write a program to read 2 (two-dimensional) integer array of size (2x2) ,then print the elements as binary order and sum these arrays elements in third array and print it?

```

#include <iostream>
using namespace std;
const int n=2;

int main( ) {
    int a[n][n],b[n][n],c[n][n], i, j;

    cout<<"read first array :\n";
    for(i=0;i<n;i++)
    {
        for(j=0;j<n;j++)
            cin>>a[i][j];
        cout<<"\n";
    }
    cout<<"read second array :\n";
    for(i=0;i<n;i++)
    {
        for(j=0;j<n;j++)
            cin>>b[i][j];
        cout<<"\n";
    }
    cout<<"print first array :\n";
    for(i=0;i<n;i++)
    {
        for(j=0;j<n;j++)
            cout<<a[i][j]<<"\t";
        cout<<"\n";
    }
    cout<<"print second array :\n";
    for(i=0;i<n;i++)
    {
        for(j=0;j<n;j++)
            cout<<b[i][j]<<"\t";
        cout<<"\n";
    }
    cout<<" sum of A & B arrays in C :\n";
    for(i=0;i<n;i++)
    {
        for(j=0;j<n;j++)

```

```

    {
        c[i][j]=a[i][j]+b[i][j];
        cout<<c[i][j]<<"\t";
    }
    cout<<"\n";
}
return 0;}

```

Write the same program by use functions:

```

#include <iostream>
using namespace std;
const int n=2;

/*---Functions-----*/
int read_array(int z[n][n])
{
    for(int i=0;i<n;i++)
    {
        for(int j=0;j<n;j++)
            cin>>z[i][j];
        cout<<"\n";
    }
}

//-----
int print_array(int z[n][n])
{
    for(int i=0;i<n;i++)
    {
        for(int j=0;j<n;j++)
            cout<<z[i][j]<<"\t";
        cout<<"\n";
    }
}

//-----

```

```

int main( ) {
    int a[n][n],b[n][n],c[n][n], i, j;

```

```

cout<<"read first array a:\n";
read_array(a);

cout<<"read second array b:\n";
read_array(b);

cout<<"print first array a :\n";
print_array(a);

cout<<"print second array b :\n";
print_array(b);

cout<<" sum of A & B arrays in C :\n";
for(i=0;i<n;i++)
{
    for(j=0;j<n;j++)
    {
        c[i][j]=a[i][j]+b[i][j];
        cout<<c[i][j]<<"\t";
    }
    cout<<"\n";
}
return 0;
}

```

H.W:

- **H.W.1:** Consider Int X[3][4] ,what is the address of the X[2][2] if the base address equal (300) and each element of the array occupy 1 byte. Suppose the programming store 2D using Column-Major.
- **H.W.2:** Consider Int X[5][3] ,what is the address of the X[3][2] if the base address equal (100) and each element of the array occupy 2 byte. Suppose the programming store 2D using Row-Major.
- **H.W.3:** Write C++ program to give the average for array consist of 5 real numbers.
- **H.W.4:**

اكتب برنامجاً لقراءة عناصر مصفوفة $A(4 \times 3)$ ثم عمل ما يلي:

a -طباعة المصفوفة بشكل ثنائي مرتب.

b - طباعة الأرقام الموجبة و الأرقام السالبة و الأرقام الصفرية في المصفوفة

c -طباعة عدد الأرقام الزوجية والفردية.