Array

Prepared by: Lec Tasmiah Tamzid Anannya, CS Dept, AIUB

What is Array?

- Array: a set of ordered data items.
- Array is a kind of data structure that can store a fixed-size sequential collection of elements of the same type.
- You can define a variable called x, which represents not a *single* value, but an entire *set of values*.
- Instead of declaring individual variables, such as number0, number1, ..., and number99, you declare one array variable such as numbers and use numbers[0], numbers[1], and ..., numbers[99] to represent individual variables.

What is Array?

- Each element of the set can then be referenced by means of a number called an *index* number or *subscript.*
- Mathematics: a subscripted variable, x_i, refers to the *i*th element *x* in a set
- C programming: the equivalent notation is x[i]
- A specific element in an array is accessed by an index.

Number[0]	Number[1]	Number[2]	Number[3]	Number[4]	……..

Declaring Arrays

- To declare an array in C, a programmer specifies the type of the elements and the number of elements required by an array as follows −

  type arrayName [arraySize]

- This is called a *single-dimensional* array.
- The **arraySize** must be an integer constant greater than zero and **type** can be any valid C data type.
- For example:

  double myarray[10]

  char name[20]

What happens when an array is declared?

- double myarray[10]
- We can access these 10 elements individually by:

myarray[0]

myarray[1]

myarray[2]

.

.

.

myarray[9]

Remember, index of array start at 0. So, an index of 1 references the second element of the array.

Initializing array

- You can initialize an array in C either one by one or using a single statement as follows −

```
double myarray[5] = {1000.0, 2.0, 3.4, 7.0, 50.0};
```

- Or, ***myarray[0]=1000.0***

	0	1	2	3	4
myarray	1000.0	2.0	3.4	7.0	50.0

Accessing Array Elements

- An element is accessed by indexing the array name.
- This is done by placing the index of the element within square brackets after the name of the array.
- For example-

double a=myarray[0];

- The above statement will take the 1st element from the array and assign the value to variable named ***a***.

Example

```
int main()
{
    int arr[4]; /* arr is an array of 4 integers */
    arr[0] = 5;   //initializing 1st element
    arr[2] = -10; //initializing 2nd element
    arr[1] = 2; //initializing 3rd element
    arr[3] = arr[0]; //initializing 4th element

   cout<<arr[0]<<arr[1]<<arr[2]<<arr[3];

    return 0;
}
```

Example

```
#include <stdio.h>
int main () {

   int n[ 10 ]; /* n is an array of 10 integers */
   int i,j;

   for ( i = 0; i < 10; i++ ) {
      n[i] = i + 100; /* set element at location i to i + 100 */
   }
   /* output each array element's value */
   for (j = 0; j < 10; j++ ) {
      cout<<n[j];
   }
   return 0;
}
```

Taking input of an array

cin>>myarray[i];

- C does not perform bound checking on array indexing.
- It is possible to overrun the end of an array.
- Suppose, an array a is declared having 4 element,

  int a[4];
- The compiler will still let you access the 10th member by a[9].
- Of course, attempting non existent members will have disastrous results.
- So the programmers must be careful.

Practice

- Try to copy an array to another using a loop.
- Declare an array of integer for 5 numbers and calculate the sum of them.

Insert an element to the array

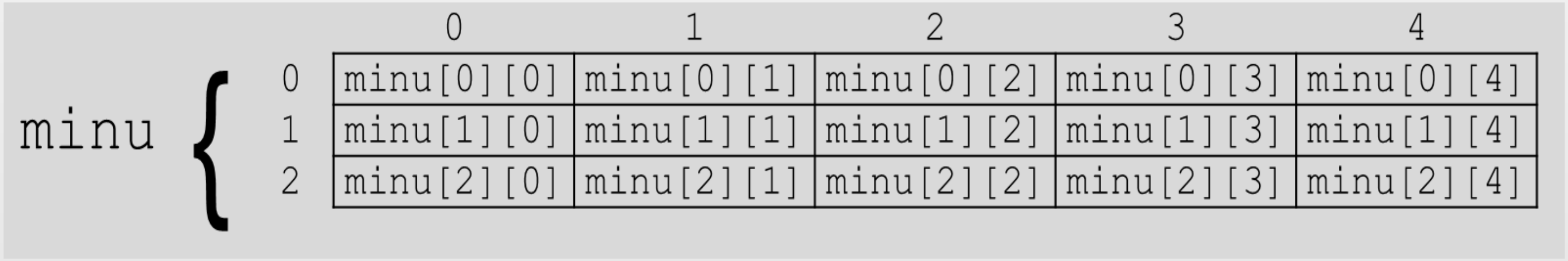
```
int main(){
    int n, pos, value;
    int arr[10];
    cout<<"Size of the array?";
    cin>>n;
    for(int i=0;i<n;i++)                    //taking input in the array
        cin>>arr[i];
    cout<<"position?";                      //In which position the new value will be added?
    cin>>pos;
    cout<<"value";                          //taking input the new value
    cin>>value;
    for(int i=n;i>=pos;i--)
    {
        arr[i+1]=arr[i];
    }
    arr[pos]=value;                           //spacing the new value in the position
    for(int i=0;i<=n;i++)
        cout<<arr[i]<<endl;
}
```

Practice

- Delete an element from the array
- Update an element in the array

2-D Array

- An array of arrays is known as 2D array.
- The two dimensional (2D) array is also known as matrix.
- A matrix can be represented as a table of rows and columns.



minu

	0	1	2	3	4
0	minu[0][0]	minu[0][1]	minu[0][2]	minu[0][3]	minu[0][4]
1	minu[1][0]	minu[1][1]	minu[1][2]	minu[1][3]	minu[1][4]
2	minu[2][0]	minu[2][1]	minu[2][2]	minu[2][3]	minu[2][4]

- The way to declare this array in C++ would be: `int minu [3][5];`

2-D Array

- Assigning values at the time of declaring a two-dimensional array can be any one of the following ways:

```
int minu[3][5] =
{1,2,3,4,5,2,4,6,8,10,3,6,9,12,15};
int minu[3][5] =
{{1,2,3,4,5},{2,4,6,8,10},{3,6,9,12,15}};
int minu[3][5] = {
  {1,2,3,4,5},
  {2,4,6,8,10},
  {3,6,9,12,15}
 };
```

Example

```
int main()
{
    int a[2][3];                    //declaring an 2-d array with row=2 and col=3

//taking input
    for(int i=0;i<2;i++)        //the outer loop will execute upto row number
    {
        for(int j=0;j<3;j++)    //the inner loop will execute upto col number
            cin>>a[i][j];
    }
//printing output
     for(int i=0;i<2;i++)
    {
        for(int j=0;j<3;j++)
            cout<<a[i][j]<<" ";

        cout<<endl;
    }
}
```