Rajarata University of Sri Lanka

Faculty of Applied Sciences

COM 1407 – Computer Programming

Practical – Arrays

Outline

- Declaration and Initialization of arrays.
- Accessing elements in Arrays.
- Populating an array with user elements.

Outcome

- Familiar with Arrays in C programming.
- Get knowledge about general structure of Arrays.
- Write C programs using arrays.

---

1 Declaration and Initialization of Arrays

- Arrays a kind of data structure that can store a fixed-size sequential collection of elements of the same type. An array is used to store a collection of data, but it is often more useful to think of an array as a collection of variables of the same type.
- A specific element in an array is accessed by an index.
- All arrays consist of contiguous memory locations. The lowest address corresponds to the first element and the highest address to the last element.
- There are two types of arrays.

1.1 Single Dimensional Array (1-D Arrays)

- **Syntax**

```
type arrayName [ arraySize ];
```

- The **arraySize** must be an integer constant greater than zero and **type** can be any valid C data type.

```
int balance[10];
```

- When initializing a 1 – D Array there are several methods.
  - ✓ **Method 1** - `<Array name>[Index] = <Intial Value>`

    Eg
    ```
    int float[4];
    a[0]=1.5;    a[1]=2.74;    a[]=3.11;
    a[3]=4.04;
    ```
  - ✓ **Method 2** - `<Data Type><Array Name>[] = {Values}`

    Eg
    ```
    int a[] = {1,2,3,4};
    ```
  - ✓ **Method 3** - `<Data Type><Array Name>[<Array Size>] = {Values}`

    Eg
    ```
    char a[4] = {'s','a','m','s'};
    ```

1.1.1 Accessing elements in a 1-D array

- 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.
  - ✓ **1 – D array →**
```
int A [] = {1,2,3,4};
int a = A[1];
printf("%d",a);
```

1.2 Multi – Dimensional Arrays (2-D Arrays)

- **Syntax**
```
type arrayName [ x ][ y ];
```
- A two-dimensional array can be considered as a table which will have x number of rows and y number of columns.
- When initializing a 2 – D Array there are several methods.
  - ✓ **Method – 1** - `<Array name>[Index][Index] = <Intial Value>`
    Eg -
```
int a[2][2];
a[0][0]=1; a[0][1]=2;
a[1][0]=3; a[1][1]=4;
```
  - ✓ **Method – 2** -
```
<Data Type> <Array Name> [ <size>] [<size>]= {<Initial Values>};
```
    Eg -
```
int a[2][2] = {1,2,3,4};
```

1.2.1 Accessing elements in 2 -D array

- 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.
  - ✓ **2 – D array →**
```
int A[3][4] = {0,1,2,3,4,5,6,7,8,9,10,11};
int a = A[1][2];
printf("%d",a);
```

- Try below 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;    }

 for (j = 0; j < 10; j++ ) {
      printf("Element[%d] = %d\n", j, n[j] );}
 return 0;}
```

2 Input array elements

- You can take input from the user and store it in an array element.
- Try below example.

```
#include <stdio.h>

int main() {
  int values[5];

  printf("Enter 5 integers: ");

   for(int i = 0; i < 5; ++i) {
     scanf("%d", &values[i]);
  }

printf("Displaying integers: ");
for(int i = 0; i < 5; ++i) {
     printf("%d\n", values[i]);
  }
  return 0;
}
```

3 Exercises

a. Write a program to get 5 numbers from the user and store those in an array and print the values in an array.
b. Write a program to enter 5 numbers from keyboard and print summation of those values. Change the program to print those 5 numbers in reverse order.
c. Write a program to enter and print your name by using an array.
d. Write a program to find a minimum and maximum element of an array
e. Write a program to get 5 numbers from the user and save it to an array. Also Find the Maximum element of the array.
f. Enter the given Student Reg.No and Marks into a 2-D Array and find minimum and maximum marks with specific Reg.No

Registration No	Marks
100	75
101	82
102	42
103	44
104	26

g. Using 2 – D array write a program to display a 3*3 unit matrix.

Write a program to input following table to an array.

Registration No	Mathematics	Science	English
100	90	89	93
101	25	45	58
102	45	55	61
103	78	68	33
104	32	44	44

h. Write a program to add two 3*3 matrices. Elements of matrices should be user inputs.