Faculty of Information Technology, University of Moratuwa
BSc. (Hons) in Information Technology
BSc. (Hons) in Artificial Intelligence
Fundamentals of Programming IN1101

Level 1 – Semester 1 Lab Tutorial - Arrays

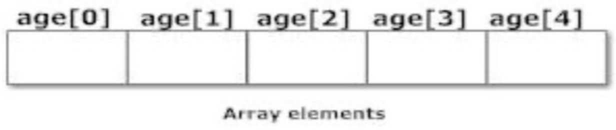
Single dimensional arrays

Introduction
In this lab session, you'll learn about single dimensional array in C.

Arrays
What is it? An array is a data structure that holds a number of related variables. Thus, an array has a size which is the number of variables it can store. All of these variables must be of the same type. (In essence declaring an array allows you to use many variables of the same type without explicitly declaring all of them.)

Each cell of the array can hold one data item. Furthermore, each cell has its own index, to distinguish it from the rest of the cells. In C, these cells are always numbered from 0 to the size of the array minus one.

int age[5];



Declaring Arrays
Here is the generic syntax for an array declaration:
type <aray_name>[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 to define an integer array called numbers of size 10, we would do the following:
int numbers[10];

Note: the expression inside the brackets of an array declaration must evaluate to a constant. This is because the computer needs to know how much space to allocate for the array beforehand.

Initializing Arrays

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

```
int numbers[10]={1,2,3,4,5,6,7,8,9,10};
```

<u>Determine the size of an array.</u>

The array size can be determined using the sizeof() function.

```
int arr[5] = {4, 7, 9, 2, 7};
int size = sizeof(arr) / sizeof(arr[0]);
```

here `sizeof(arr)` is 20 bytes because `arr` holds five integers, and each integer takes 4 bytes, as you know.

`sizeof(arr[0])` is 4 bytes because `arr[0]` location is an integer and integer takes 4 bytes.

So `sizeof(arr)/sizeof(arr[0])` => 20 / 4 => 5.

Example:

```
#include <stdio.h>
int main() {
    int arr[5] = {4, 7, 9, 2, 7};
    int size = sizeof(arr) / sizeof(arr[0]);
    for (int i = 0; i < size; i++) {
        printf("%d ", arr[i]);
    }
}
```

Output: 4 7 9 2 7

Important: Do not use `sizeof()` function for arrays in function parameters.

Consider the following example of what happens if you use the `sizeof()` function for an array in a function parameter.

Example:

```c
#include <stdio.h>
void fun(int myArr[]);
int main() {
    int myArr[5] = {4, 7, 9, 2, 7};
    fun(myArr);
}

void fun(int myArr[]) {
    int size = sizeof(myArr) / sizeof(myArr[0]);
    for (int i = 0; i < size; i++) {
        printf("%d ", myArr[i]);
    }
}
```

Output: 4 7

When the method `fun()` is passed the array parameter `myArr[]` and attempts to determine its size using the `sizeof()` operator, an error is thrown.

Array parameters in C are addressed like pointers. Due to this transformation, the for loop inside fun() only runs twice regardless of the array's size (because the `sizeof(myArr)` is equals to `sizeof(int*)`, which is 8 bytes since it is a pointer and pointers occupy the 8 bytes of memory and int is 4).

`sizeof(myArr)/sizeof(myArr[0])` => `sizeof(int*)/sizeof(myArr[0])` => 8/4 = 2

As a result, `sizeof()` cannot be used to determine the total number of items for arrays in function parameters.

To resolve this issue, the array size must be passed as a separate argument to the function.

Example:

```c
#include <stdio.h>
void fun(int myArr[], int size);
int main() {
    int myArr[5] = {4, 7, 9, 2, 7};
    fun(myArr, 5);
}

void fun(int myArr[], int size) {
    for (int i = 0; i < size; i++) {
        printf("%d ", myArr[i]);
    }
}
```

Output: 4 7 9 2 7

Exercises

1. Write a C program to input n integer numbers to an array and find the highest number among them.
2. Write a C program to store n numbers in an array and find the smallest number among them.
3. Write a C program to input the ages of ‘10’ students to an array and count the number of students who have the age between 17-19 (inclusive of 17 and 19).
4. Write a C program to get the salaries of 10 employees to an array and count the number of employees getting salary in between 50,000 – 75,000 (inclusive of 50,000 and 75,000).
5. Write a C program to find the average of array elements. Use a separate function.
6. Write a C program to sort the values of the array of type int in descending order. Use a separate function.
7. Write a C program to find the correlation coefficient **r** and the regression equation of y on x for n (= 12) pairs of observations of the CPI in the two subjects x and y. The correlation coefficient is r.

X	1	0	3.2	4	1	5	7	0	2	1.1	-1	4.1
Y	3	5	0	-1	0.5	-1	-2	3	4	1	8.1	2

Regression equation of y on x is **y – ybar = byx (x – xbar)**. The regression establishes a linear relationship between x and y given by the data. It helps us in predicting unknown y values for x values input by the user. The correlation coefficient r always lies between -1 and 1 and is: The equation of the correlation coefficient r is as follows.

$$r = \frac{\Sigma xy - \frac{\Sigma x \Sigma y}{n}}{\sqrt{\left(\Sigma x^2 - \frac{(\Sigma x)^2}{n}\right)}\sqrt{\left(\Sigma y^2 - \frac{(\Sigma y)^2}{n}\right)}}$$

The regression coefficient of y on x. it helps us to predict y value given the x value

$$\text{byx} = \frac{\Sigma xy - \frac{\Sigma x \Sigma y}{n}}{\left(\Sigma x^2 - \frac{(\Sigma x)^2}{n}\right)}$$