

Name:

Sec:

06/13/2006

Lab 2.1 Arrays

An array is a structured data type that is a collection of a fixed number of components wherein all of the components are arranged in a list. Arrays are declared using square brackets with the number of components shown inside the brackets, as in `int num[5];`. To access a component of the array, you also use the square bracket with an integer inside the brackets indicating the position from the beginning of the array.

Working with arrays almost always requires iteration. Like all variables, an array and its size must be declared when the program is compiled. An individual array component is treated the same as any individual variable. Operations are the same for individual components as an operation on any variable.

For example, the following code would assign the fourth variable of an array to the variable `num`:

```
int array[5] = {5, 8, 9, 6, 4};
int num;
num = array[3];
```

Objectives

In this lab, you process individual array components. You will also practice using functions, and see how to pass an array as a parameter to a function.

Instructions

1. You will write a small program that prompts the user to enter 10 numbers. This program should read the numbers into an array (using a loop). Do these steps first. Create an array of 10 integers. Write a loop to get 10 integers from the user using `cin`. Do these steps in your `main()` function.
2. Arrays may be passed in to functions just as a simple single data type may be. To create a function that accepts an array as a parameter, you write something like this for the function definition:

```
void arrayfunction(int arrayparam[], int arraysizesize)
{
}
```

The previous is an example of a function that has the name “arrayfunction”. The function takes 2 parameters, an array of integers, and a single integer. The first parameter is actually an array, this is indicated by having the [] brackets after the name of the parameter. This says an array of integers is passed as a parameter to the function, but we don't know the size of the array (e.g. the number of items in the array). The second parameter is common when creating a function that takes an array as a parameter, since inside of the function we will need to know how many items are actually in the array. Therefore we usually also need to pass in the size of the array to the function.

3. Write a function for your program that takes the array of 10 integers entered by the user and finds the smallest number in the array. The smallest number should be returned as the result of the function. Your function prototype should therefore look something like:

```
// Given a list of integers, return the smallest integer that is in the
// list as a result of calling this function.
int smallest(int list[], int size);
```

That is to say the function called `smallest` takes an integer array and the arrays size as its 2 parameters. It returns an `int` as the result, which will be the value of the smallest item in the `list[]` array that is passed to the function.

4. Write another function that accepts an array as a parameter. This function should find the average of the items in the array passed in, and return the average value as its result. To find the average you need to 1) sum up all of the integer values in the array and then 2) divide the sum by the number of items (which should be the second parameter that is passed in to your function, e.g. the size of the array).

Lab 2 Finished

You have now completed Lab 2. If your program compiles and runs correctly and you have successfully uploaded your source file to the eCollege online submission site, then you are done. At this point, you might want to review any concepts about arrays and functions that are still fuzzy to you, as well as begin to look at multi-dimensional arrays for next class.