Array Variables

Submitted by Andy Lindsay on Thu, 03/21/2013 - 16:04

original source: http://learn.parallax.com/propeller-c-start-simple/array-variables

Lesson edited to work with **Dev-C++** IDE by Jeff La Favre 10/22/15

*(Updated 2013-08-08 for SimpleIDE 0.9.4 and its Learn folder's Simple Libraries and Examples) [SimpleIDE is the IDE for use with the robot. This lesson is edited so that we can use **Dev-C++** as the IDE, which does not require a robot – J. La Favre]*

A variable *array* is a group of variables stored under the same name but with different index values. In the array declaration and initialization example below, you can think of the index value as the position of a specific variable in the list.

```
  int p[] = {1, 2, 3, 5, 7, 11};
```

Each array element has a name (which is **p** in this example, the same as the array name) and an index [between brackets] that makes it possible to select an element.

This declaration sets aside six variables, each storing a prime number. This array has been initialized so that the **p[0]** array element stores 1. Likewise, the **p[1]** element stores 2. It continues for the rest of the array elements: **p[2]** stores 3, **p[3]** stores 5, **p[4]** stores 7, and **p[5]** stores 11.

- Start **Dev-C++**.
- Open the **File** menu and select **New**. Then select **Source File**.
- Click the mouse in the text window of **Dev-C++** and use the keyboard to enter the following text: **#include <stdio.h>**
- Open the **File** menu and select **Save**, which opens a **Save As** dialog box.
- In the dialog box, open the drop-down labeled **Save as type** and select **c source files(*.c)**. In the **file name** slot enter this name for the file: **array variables.** At the top of the dialog box there is a **Save in** slot, which determines

where the file will be saved. Make sure you know the location where you are saving your file so that you can find it later. Now click the **Save** button to save your program file.

- Copy the text in the box below and paste it into the text window of **Dev-C++** under the first line of text you have already entered. Alternatively you can type in the code, without comments, using the keyboard.
- Click the **Save** button to save the code you just pasted or entered with the keyboard.
- Now examine the program code, and try to predict what will be displayed when the program runs.
- Run the program by opening the **Execute** menu and selecting **Compile and Run**. If there are no errors in the program, a new program window will open. Compare the actual output to your predicted output.

```
int main()
{
 int p[] = {1, 2, 3, 5, 7, 11};  //create an array variable of 6 elements named p and assign values of 1, 2, 3, 5, 7, 11
 printf("p[0] = %d\n", p[0]);  //displays the value stored in p array element 0
 printf("p[3] = %d\n", p[3]);  //displays the value stored in p array element 3
}
```

How it Works

`int p[] = {1, 2, 3, 5, 7, 11}`

initializes an array named `p` with a list of numbers between braces `{ }`. Keep in mind that `p[0]` stores `1`, `p[1]` stores `2`, `p[2]` stores `3`, `p[3]` stores `5`, and so on. So, **`printf("p[0] = %d\n", p[0])`** displays the contents of `p[0]`, which is `1`. Likewise, **`printf("p[3] = %d\n", p[3])`** displays the contents of `p[3]`, which is `5`.

Did You Know?

- You can declare an array with six elements, but without initializing the values, like this: `int p[6];`
- You can assign and re-assign values to array elements at various points in your program like this: `p[4] = 7;`

Try This

Here is an example that shows how the value of an array element can be reassigned. Take a look at the second to last statement in the `main` function. It's `p[3] = 101`. After that, `printf("p[3] = %d\n", p[3])` will allow you to verify that the new value `p[3]` stores is 101.

- Open the **File** menu and select **Save As** and save your program with the name **Array Variable Reassignment** (make sure to save as **c source files(*.c)** file type).
- Modify the program using the example on the next page. (Just add the last two lines.)
- Click the **Save** button to save your modifications.
- Run the program by opening the **Execute** menu and selecting **Compile and Run**. Verify that the program starts with `p[3]` being 5, and that it reassigns it to 101.

```
int main()

{

  int p[] = {1, 2, 3, 5, 7, 11};

  printf("p[0] = %d\n", p[0]);

  printf("p[3] = %d\n", p[3]);

printf("Next the code will change the value of p[3] to 101 and display new value \n", p[3]);

 p[3] = 101; //reassign the value of p[3] to 101

printf("p[3] = %d\n", p[3]);

}
```

Your Turn

- Expand Array Variable Reassignment.c by adding commands to the end of the program that puts 11 in `p[0]`, 7 in `p[1]`, 5 in `p[2]`, 3 in `p[3]`, 2 in `p[4]`, and 1 in `p[5]`.
- Expand your program further so that it displays the values after they have been updated.