CS1050 – Prelab 5

Spring 2019

Concepts to Practice

- Arrays
- Passing arrays to functions
- Symbolic constants

Description

For the prelab assignment, you need to implement a program that works in two phases. In Phase 1, your program will read up to 10 non-negative integer values into an array. You will know that the user has entered the last value when the user enters -1. Note that you will need to check to make sure the user does not enter more than 10 values. You should define a symbolic constant called ARRAY_SIZE and use this constant rather than ever passing a hard-coded 10 to any function or any array declaration.

In Phase 2, you will present a menu of possible operations that the user may request be performed on the array. Each operation will be implemented in its own function and will correspond to a menu option. After each operation is performed, the menu should again be displayed so the user can select another operation. When the user enters -1, the program exits.

The main() function in your program should:

1. Print a message welcoming the user to the Array Operations Program.
2. Call the FillArray() function to allow the user to enter values to be stored in your array.
3. Call the DisplayMenu() function to allow the user to select a function to perform on the array.
4. If the selected menu item is legal and not -1, call the function that corresponds to the selected number and print the result returned by that function.
5. If the selected menu item is not -1, continue with step #3 so the user may enter additional operations.
6. Print out a message thanking the user for using the program.

Functions You Must Write

You may write any functions you wish to implement this program, in **addition** to the following functions. However, you **must** implement the following functions:

- **int FillArray(int array[], int size)** – This function takes an array that has size elements. The function prompts the user to enter values until the user enters -1, or all size elements have been filled. The function returns the number of elements entered.
- **int DisplayMenu()** – This function displays a list of the following functions: Sum, Product. It returns the number corresponding to the selected function.
- **int Sum(int array[], int size)** – This sums all of the elements in the given array, and returns the total.
- **int Product(int array[], int size)** – This multiplies all of the elements in the given array, and returns the product.
- **int main(void)** – Of course, you need to write a main() ☺.

Hint

Remember that arrays are passed to functions sort of similar to “pass by reference”. So, if I pass an array called “myarray” to a function by using its name, this is the same as passing the address of the first element of that array (&myarray[0]). This is important, because it means that a function receiving an array as an argument can make changes to the values contained in that array.

Sample Output (Run #1)

```
JimR@SkullCanyon:~/CS1050/CS1050_Spring2019/Lab5$ compile prelab5.c
JimR@SkullCanyon:~/CS1050/CS1050_Spring2019/Lab5$ ./a.out
Welcome to the Array Operations Program!

Enter value for array element 0:
3
Enter value for array element 1:
5
Enter value for array element 2:
7
Enter value for array element 3:
9
Enter value for array element 4:
-1

Select a function:
1. Sum
2. Product
Enter -1 to exit
1
Sum = 24

Select a function:
1. Sum
2. Product
Enter -1 to exit
2
Product = 945

Select a function:
1. Sum
2. Product
Enter -1 to exit
1
Sum = 24

Select a function:
1. Sum
2. Product
Enter -1 to exit
2
Product = 945

Select a function:
1. Sum
2. Product
Enter -1 to exit
2
Product = 945

Select a function:
1. Sum
2. Product
Enter -1 to exit
-1

Thanks for using the Array Operations Program!
```

Sample Output (Run #2)

```
JimR@SkullCanyon:~/CS1050/CS1050_Spring2019/Lab5$ vi prelab5.c
JimR@SkullCanyon:~/CS1050/CS1050_Spring2019/Lab5$ compile prelab5.c
JimR@SkullCanyon:~/CS1050/CS1050_Spring2019/Lab5$ ./a.out
Welcome to the Array Operations Program!

Enter value for array element 0:
1
Enter value for array element 1:
2
Enter value for array element 2:
3
Enter value for array element 3:
4
Enter value for array element 4:
5
Enter value for array element 5:
6
Enter value for array element 6:
7
Enter value for array element 7:
8
Enter value for array element 8:
9
Enter value for array element 9:
2

Select a function:
1. Sum
2. Product
Enter -1 to exit
1
Sum = 47

Select a function:
1. Sum
2. Product
Enter -1 to exit
2
Product = 725760

Select a function:
1. Sum
2. Product
Enter -1 to exit
2
Product = 725760

Select a function:
1. Sum
2. Product
Enter -1 to exit
-1

Thanks for using the Array Operations Program!
```