Code Assignment 4
Array Operations

ECE 131 – Programming Fundamentals

Instructor: M. Wolverton

Items Due

- **`arrayUtil.h`**
- **`arrayUtil.c`**
- **`arrayOps1.c`**
- **`arrayOps2.c`**

Instructions

Complete the coding directive in your C language IDE of choice (Eclipse, NetBeans) or by command line compiling. Once you have verified that the program works as intended, login to the file server at *cec-code-lab.aps.edu* and create a folder with the assignment name (e.g. Code 3). Locate all **.c** and **.h** source files written by you (e.g. hlgame.h, hlgame.c, main.c) and upload them to into the assignment folder. Code will be downloaded and archived for grading on the assignment due date.

Array-Utilities

Write a library for manipulating int arrays containing a few essential functions. Then write a couple of short program that include the library and demonstrate it.

Array-Ops-1

Demonstrate the int array functions in a program written in main1.c that works as described below.

Create an array of 25 int values.
Print the array (before any initialization - these are the values already in the memory now allocated the the array).
Set all values to zero.
Print the array.
Set all values to a random number between 0 (inclusive) and 1000 (exclusive)
Print the array.
Sort the array from largest to smallest.
Print the array.

Note: Please print some kind of separator or a status statement between each array print. Otherwise it will be difficult to figure out where one array print stops and the next starts.

Array-Ops-2

Demonstrate the char array functions in a program written in main2.c that works as described below.

Create an array of 12 char values.
Print the array (before any initialization - these are the values already in the memory now allocated the the array).
Set all values to ‘ ‘ (space).
Print the array.
Repeat 5 times {
    Generate a random password in the array.
    Print the array.
}
Set the array values to `{h,e,l,l,o, ,w,o,r,l,d,!}`. Use your library to store the the number of instances of ‘`l`’ in a variable. (it should be 3).
Print out the array and the variable holding the number of instances of ‘`l`’.

Note: As before, please print some kind of separator or a status statement between each array print. Otherwise it will be difficult to figure out where one array print stops and the next starts.

Code Structure

- Create a library consisting of a source file (`arrayUtil.c`) and a header file (`arrayUtil.h`). Include `arrayUtil.h` in your main source code files for the project, `main1.c` and `main2.c`.
- Prototype the following functions in `arrayUtil.h`, with full implementation in `arrayUtil.c` .

int Array Utilities

`void printIntArray(int array[], int size)`

Description: Print each array value on a new line using printf.

`void setAllIntArray(int array[], int size, int value)`

Description: Sets each array value to equal parameter value.

`void randIntArray(int array[], int size, int min, int max)`

Description: Sets each array value to a random number between min (inclusive) and max (exclusive).

`void sortIntArray(int array[], int size)`

Description: Reorders the array from smallest to largest value.

char Array Utilities

`void printCharArray(char array[], int size)`

Description: Print each character side-by-side on the same line followed by a line return.

`void setAllCharArray(char array[], int size, char c)`

Description: Sets each array value to equal parameter value.

`void randPassWord(char array[], int size)`

Description: fills the array with a random character a-z, A-Z, or 0-9. No other characters should be possible.

`int charCountInArray(char array[], int size, char c)`

Description: Returns the number of instances in the array of the character in c.

Sample Output

main1

```
----uninitialized array values-----
[0]: 0
[1]: 0
[2]: 15775231
                        ...
[24]: -1106640944
-----array set to zero-----
[0]: 0
[1]: 0
[2]: 0
                        ...
[24]: 0
-----array randomized-----
[0]: 815
[1]: 928
[2]: 516
                        ...
[24]: 752
-----array sorted-----
[0]: 65
[1]: 129
[2]: 153
                        ...
[24]: 974
```

main2

```
----uninitialized array values-----
05F????5+V?
-----array set to space-----

-----random passwords-----
5N8F0ji7o0s0q9R5Smu9uPL2v
Vpbu4M737OW57BXAmGYy20u07
U011f5oY92lBi9A67Z9d0k8iH
70P1H4SDxS2t45MUY2BOu3WiA
Y29AioW09B90J4Gvm7nLN3pnB
-----array values set to “hello world”-----
hello world!
there are 3 instances of 'l' in the array
```