



Assignment 9: Advanced Array Concepts

Assignment Description:

This assignment will cover advanced concepts of arrays. Please review your class notes, slides and textbook to ensure you understand the necessary concepts.

YOU MAY NOT USE `cstring` OR `string.h` FOR THIS LAB!

You may use `scanf("%s",charArray)`, `gets` and `printf`.

Program 1 (12.5 pts):

You will write a function called `myStringLen`. It will take in a single char array. You can assume adequate space has been allocated to it. You will iterate through the array and count how many characters are preset (white space included) until you reach a null character (`'\0'`). You will return the number of characters found (not including the null character).

Use “`gets`” to populate a char array in main and call this function. Print the size in main.

Example output (Prompt does not have to be exact):

```
Enter a string or sentence:  
The quick red fox jumped over the lazy brown dog!  
The number of characters in your string/phrase is 49
```

Submit a screen shot of your output and the source code (.c file)!

Code output screenshots:

```
❖ make -s  
❖ ./main  
Enter a string or sentence:  
Testing my string counter function! :)  
The number of characters in your string/phrase is 38  
❖ █
```

```
❖ make -s  
❖ ./main  
Enter a string or sentence:  
Hello COP-2271 Class  
The number of characters in your string/phrase is 20  
❖ █
```

```
❖ make -s  
❖ ./main  
Enter a string or sentence:  
My function works :))  
The number of characters in your string/phrase is 21  
❖ █
```



Program 2 (12.5 pts):

You will write a function called `myStringCmp`. It will take two character arrays. You may assume that there is enough space allocated for each array. Iterate through the arrays and compare each character.

- If the characters are the same until you reach a null character `'\0'` you will return a 0.
- If the first char array would come before the second in the dictionary you will return a negative number.
- Otherwise, you will return a positive number.

```
Enter 2 words:  
Apple Zebra  
Apple comes before Zebra in the dictionary.
```

```
Enter 2 words:  
Bat Bat  
Bat and Bat are the same word.
```

```
Enter 2 words:  
Jolly Jelly  
Jolly comes after Jelly in the dictionary.
```

Submit a screen shot of your output and the source code (.c file)!

Code output screenshots:

```
❖ make -s  
❖ ./main  
Enter 2 words:  
research research  
research and research are the same word.❖
```

```
❖ make -s  
❖ ./main  
Enter 2 words:  
stop go  
go comes before stop in the dictionary.❖
```

```
❖ make -s  
❖ ./main  
Enter 2 words:  
apple banana  
apple comes before banana in the dictionary.❖
```



Program 3 (12.5 pts):

Create a function called `printArray` that will take the float array and size as parameters. The function will then print out all the elements of the float array.

Create another function called `mySwap`. This function will take in a float array, the size of the float array, and the two indexes you wish to switch. The function will then swap the values at the two indexes specified.

In Main, you will create a float array of size ten and populate it with random numbers between 0 and 1 (Use a for loop). You will then print the array, call `mySwap`, pass in 0 and 9 for the indexes, and then print the array a second time.

- *Hint: use `RAND_MAX` instead of `INT_MAX`.*

Submit a screen shot of your output and the source code (.c file)!

```
Before Swap!
0.087710
0.924924
0.115207
0.922544
0.414075
0.553972
0.725669
0.631367
0.567492
0.552843
After Swap!
0.552843
0.924924
0.115207
0.922544
0.414075
0.553972
0.725669
0.631367
0.567492
0.087710
```

Code Output Screenshots:

<pre>❯ make -s ❯ ./main Before Swap! 0.640561 0.615728 0.128002 0.756654 0.080934 0.022448 0.953801 0.910014 0.451983 0.246716 After Swap! 0.246716 0.615728 0.128002 0.756654 0.080934 0.022448 0.953801 0.910014 0.451983 0.640561 ❯</pre>	<pre>❯ make -s ❯ ./main Before Swap! 0.718376 0.139842 0.194840 0.423312 0.425137 0.434475 0.197909 0.496980 0.198399 0.580933 After Swap! 0.580933 0.139842 0.194840 0.423312 0.425137 0.434475 0.197909 0.496980 0.198399 0.718376 ❯</pre>	<pre>❯ make -s ❯ ./main Before Swap! 0.395190 0.793679 0.367531 0.066936 0.144820 0.398170 0.446358 0.463674 0.448497 0.368256 After Swap! 0.368256 0.793679 0.367531 0.066936 0.144820 0.398170 0.446358 0.463674 0.448497 0.395190 ❯</pre>	<pre>❯ make -s ❯ ./main Before Swap! 0.271179 0.990546 0.408065 0.868918 0.156077 0.711546 0.862120 0.389965 0.032177 0.638938 After Swap! 0.638938 0.990546 0.408065 0.868918 0.156077 0.711546 0.862120 0.389965 0.032177 0.271179 ❯</pre>
--	--	--	--

Name Domenic Iorfida

Date 11/30/2022



FLORIDA POLYTECHNIC
UNIVERSITY

Program 4 (12.5 pts):

You do not need to create a function for this program. You will create a 2D array of chars called shopping list. The first dimension of the array will be size 10, and the second dimension of the array will be size 80.

Ask the user what items they need to purchase at the store. Record each entry into the consecutive element of the 2D array.

Use `scanf("%s"...)` for the input.

If the user inputs "quit", or if the user enters 10 items, then stop asking the user for items and print out the shopping list in a nicely formatted way.

Hint: Remember myStrCmp you wrote earlier!

Hint 2: You should read the string into a temporary char array, check to make sure it is not "quit" then do a deep copy into the final list (Do not forget to add \0 as the last letter).

```
Enter an item or type quit:
Apple
Enter an item or type quit:
Milk
Enter an item or type quit:
Orange
Enter an item or type quit:
Eggs
Enter an item or type quit:
Yogurt
Enter an item or type quit:
Juice
Enter an item or type quit:
Bread
Enter an item or type quit:
Donuts
Enter an item or type quit:
Soda
Enter an item or type quit:
Pizza
0. Apple
1. Milk
2. Orange
3. Eggs
4. Yogurt
5. Juice
6. Bread
7. Donuts
8. Soda
9. Pizza
```

Submit a screen shot of your output and the source code (.c file)!

Code Output Screenshots:

```
❖ make -s
❖ ./main
Enter an item or type quit:
apples
Enter an item or type quit:
bananas
Enter an item or type quit:
quit
0. apples
1. bananas
❖ █
```

```
❖ make -s
❖ ./main
Enter an item or type quit:
pens
Enter an item or type quit:
pencils
Enter an item or type quit:
calculator
Enter an item or type quit:
ruler
Enter an item or type quit:
quit
0. pens
1. pencils
2. calculator
3. ruler
❖ █
```

Name Domenic Iorfida

Date 11/30/2022



FLORIDA POLYTECHNIC
UNIVERSITY

```
❖ make -s
❖ ./main
Enter an item or type quit:
apple
Enter an item or type quit:
banana
Enter an item or type quit:
turkey
Enter an item or type quit:
ham
Enter an item or type quit:
chicken
Enter an item or type quit:
pork
Enter an item or type quit:
steak
Enter an item or type quit:
ham
Enter an item or type quit:
ice
Enter an item or type quit:
veins
0. apple
1. banana
2. turkey
3. ham
4. chicken
5. pork❖
6. steak
7. ham
8. ice❖
9. veins
❖
```