

Lab 9**Problem 1**

Write a program that will search its input for an count the number of occurrences of a given character. The character should be given as a command line argument. Suppose that the executable version of the program is in the file count. A sample run of a similar program is below. The user's input in in **bold**.

\$ **./count s**

Counting s's. Enter your input now. Press return to finish.

Sally sells sea shells down by the sea shore.

There were 8 s's in that text.

Problem 2

Perfect-sized arrays.

You will gain experience creating perfectly sized arrays to hold data using malloc and realloc. Your program should contain one memory allocated (malloc) arrays for the following types:

- Integer
- Double
- Character
- String

For this problem, the user should be allowed to enter an unlimited amount of integers, doubles, characters, or strings. Your program should identify which type of input was received, and dynamically allocate space for it in its proper array (realloc). The contents of all four arrays should be displayed to the user after each entry.

Example: (Bolded text is user input)

Enter your input: **1**

String List:

Integer List : 1

Double List:

Character List:

Enter your input: **7.6**

String List:

Integer List : 1

Double List: 7.6

Character List:

Lab 9

Enter your input: **apples**

String List: apples

Integer List : 1

Double List: 7.6

Character List:

Enter your input: **a**

String List: apples

Integer List : 1

Double List: 7.6

Character List: a

Enter your input: **86**

String List: apples

Integer List : 1 86

Double List: 7.6

Character List: a

Hints:

- Use integers to keep track of how many elements of each type you have (use with realloc)
- Before using realloc, you MUST malloc! (Even if you malloc size 0!)
- Strings are arrays of characters! (A single dimension malloc array won't cut it!)
- Remember to use fgets to retrieve user input!
- Use the strt* family of function to parse your input! (strtol, strtod)
- Break this problem into smaller parts tackle them one at a time