

CYBR 401
Homework 2
Shell Project – Part 1
Parsing

I. Overview

In this project you will be creating an interactive shell for your OS. This project will be broken into three parts, each building on the previous work. At the end, you will be able to run limited OS commands from within your operating system construct.

II. Part 1 Description

The first portion of this project will see you build a simple command line parser. For any shell program, command line arguments must be effectively parsed before system calls can be made. We will be solving this problem in this portion of the project.

For this assignment, create a C program that will take command line variable input. The first command line argument will be a delimiter. The second will be a string. Your program will split up the incoming string based on the incoming delimiter. At each split, you will calculate the length of the new string.

III. Requirements

- **The splitting and length calculations need to be done in a SINGLE function.** This function will be used in future projects. NOTE: you can write helper functions such as `getLength(char *)`, just be sure to copy those with your parsing function into future starter code.
- **You must use `malloc()` to allocate memory and copy data to the allocated space.** No static string inputs are accepted. You **must null terminate all `malloc()` strings.**
- **You must check for appropriate command line argument numbers** and handle any errors that could occur with too much or too little input.
- **You may only use `stdlib.h` and `stdio.h` as libraries.** You may *not* use built in Tokenizing or String functions that are included in C.

IV Rubric

Criteria	Pts
Uses command line args to get string and delimiter	/ 10
Splitting and Length included in a single function	/ 10
Uses <code>malloc()</code> and checks argument number	/ 10
Only uses <code>stdlib.h</code> and <code>stdio.h</code> functions	/ 10
Output is correct and matches examples	/ 20
Logic is correct and works properly	/ 40

V. Examples

E1:

INPUT

Delimiter: ,

String: Bill, John, Phil, Anne, Jane, Sarah

OUTPUT

Bill Length 4
John Length 4
Phil Length 4
Anne Length 4
Jane Length 4
Sarah Length 5

```
aspanier@comeback-kid:~/Desktop/C Grading$ ./a.out "," "Bill,John,Phil,Anne,Jane,Sarah"
Bill Length 4
John Length 4
Phil Length 4
Anne Length 4
Jane Length 4
Sarah Length 5
```

E2:

INPUT

Delimiter: :

String: a:bbbA:c

OUTPUT

a Length 1
bbbA Length 4
c Length 1

```
aspanier@comeback-kid:~/Desktop/C Grading$ ./a.out ":" "a:bbbA:c"
a Length 1
bbbA Length 4
c Length 1
```