California State University, Sacramento
College of Engineering and Computer Science

Computer Science 35: Introduction to Computer Architecture

Spring 2016 – Lab 5 – *Mr. Meeseeks*

Overview

Seriously? Did your instructor actually make a lab based on Mr. Meeseeks?

Ok... Fine…

Mr. Meeseeks is a helpful and optimistic creature that is spawned by using a "magical" Meeseeks Box. Once created, he will do ***anything*** to complete a simple task for you. Once the task is done… ***poof***… he is gone.

While incredibly helpful, Mr. Meeseeks tends to have some strange personality quirks. First, he tends to repeat your task before he does it. Second, he also tends to shout everything he says.

Your Task

Your program will scan (read) a string from the keyboard that represents the task. You will then print it back to the screen in uppercase – to simulate Mr. Meeseeks shouting.

Essentially, you are going to store the string in memory and then convert it to uppercase using a loop.

Your solution doesn't to look exactly like the example below. But, make sure to fulfill all the requirements. You can use this output to test if your program is correct.

Examples

Make sure to check if each character is a lowercase letter. The user's input is in **red**. The capitalized outputted string is in **blue**.

```
I'M MR. MEESEEKS. LOOK AT ME!
do this lab

DO THIS LAB? CAN DO!
```

```
I'M MR. MEESEEKS. LOOK AT ME!
Study for the final!

STUDY FOR THE FINAL!? CAN DO!
```

```
I'M MR. MEESEEKS. LOOK AT ME!
Find the cardinality of {}

FIND THE CARDINALITY OF {}? CAN DO!
```

Reading and Storing Text

To read text from the keyboard, please read about the ScanCString subroutine in the CSC35 Library. You will need to create a buffer large enough to hold the phrase.

The example below creates a buffer (space) of 30 characters.

```
Task:
   .space 30
```

Tips

- Read the string into a buffer. Use the **`.space`** directive. Please see below on how to use ScanCString.
- First try to print the scanned string to the screen. This will test if you inputted it correctly.
- Read the documentation LengthCString. Your lab will not work if you don't use it.
- Read each character in the string into an **8-bit register!**
- Check if each character is a lowercase letter. If so, convert it to uppercase. Look at an ASCII Chart. Alternatively, you can also check if each character **isn't** a lowercase letter.

Requirements

You must think of a solution on your own. The requirements are as follows:

1. Print a Mr. Meeseeks greeting to the screen
2. Scan a string from the keyboard using the library call.
3. Convert it to uppercase.
4. Display the sentence to the screen with some industrious text like "CAN DO!", "NO PROBLEM!", "YESSIREE!" etc….
5. Do **not** create a subroutine. You will not receive credit if you do.

Submitting Your Lab

Run Alpine by typing the following and, then, enter your username and password.

```
alpine
```

Please send an e-mail to yourself (on your Outlook, Google account) to check if Alpine is working. To submit your lab, send the source file (not a.out or the object file) to:

```
dcook@csus.edu
```

UNIX Commands

Editing

Action	Command	Notes
Edit File	**nano** *filename*	"Nano" is an easy to use text editor.
E-Mail	**alpine**	"Alpine" is text-based e-mail application. You will e-mail your assignments it.
Assemble File	**as -o** *objectfile asmfile*	Don't mix up the *objectfile* and *asmfile* fields. It will destroy your program!
Link File	**ld -o** *exefile objectfiles*	Link and create an executable file from one (or more) object files

Folder Navigation

Action	Command	Description
Change current folder	**cd** *foldername*	"Changes Directory"
Go to parent folder	**cd ..**	Think of it as the "back button".
Show current folder	**pwd**	Gives a file path
List files	**ls**	Lists the files in current directory.

File Organization

Action	Command	Description
Create folder	**mkdir** *foldername*	Folders are called directories in UNIX.
Copy file	**cp** *oldfile newfile*	Make a copy of an existing file
Move file	**mv** *filename foldername*	Moves a file to a destination folder
Rename file	**mv** *oldname newname*	Note: same command as "move".
Delete file	**rm** *filename*	Remove (delete) a file. There is **no** undo.