California State University, Sacramento
College of Engineering and Computer Science

Computer Science 35: Introduction to Computer Architecture

Fall 2021 – Lab 5 – ***Magical Vending Statue***

Overview

Whether you are working a nine-to-five job or running from class to class, there is one machine that is in all our lives – the vending machine.

The student common rooms at ***Hogwarts School of Witchcraft and Wizardry*** are no different. Except, of course, their idea of a "machine" is quite different.

Sitting majestically, a vending contraption graces each the four common rooms. Each "contraption" is uniquely suited for their house, appearing as a peculiarly large statue of their house animal. Ravenclaw has a statue of a raven; Gryffindor has a statue of a lion; Hufflepuff has a statue of a badger; Slytherin has a statue of a snake.

Students feed the coins to the statue and it, in return, "coughs up" *(a nice way of saying it)* the item they seek.

Your Task

You are going to use the odd muggle technology called "computers" to simulate the vending statue. Your program will display a menu of items, input the amount of knuts, input their choice and then, output their selection and their change.

You must a switch statement for this lab.

Have fun!

You can come up with your own theme. You don't have to use mine. Use your imagination!

- Cat items
- Dog items
- Rick and Morty items
- Pokemon items
- etc....

Example

Your solution doesn't have to look exactly like the example below. The user's input is printed in **blue**. The data outputted from your calculations is printed in **red**. You don't have to make the text that color in your program.

```
Oh! Hi there! I'm the Gryffindor Vending Statue.

1. Quill & Ink (25 knuts)
2. One-Day-Only Cauldron (85 knuts)
3. One-Day-Only Wand (120 knuts)
4. Every Flavor Beans (42 knuts)
5. Cancel the order (0 knuts)

How many knuts are you feeding it?
200

Enter your selection:
3

A wand flies out of the statue.

Your change is 80 knuts.
```

Requirements

This activity may only be submitted in Intel Format.

Using AT&T format will result in a zero. Any work from a prior semester will receive a zero.

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

1. Display a menu of items and costs. You must have (at least) five. The last one should cost zero – which will let the user get back their money. Make sure to also print a name for your vending machine.
2. Input how much money they entered
3. Input their selection
4. Output the item they bought to the screen. You must use a Switch Statement (or similar logic). Using a table (which we haven't covered yet) will result in a zero on the lab.
5. Handle an invalid entry by using a "default" case.
6. Output their change to the screen.

Submitting Your Lab

Please set the subject field of your e-mail to be:

CSc 35 – #

...where # is your lecture section number. This will help me sort your work.

To submit your lab, you must run Alpine by typing the following. You might have to re-enter your username and password.

```
alpine
```

To submit your lab, send the assembly file (do not send the a.out or the object file) to:

```
To      : dcook@csus.edu
```

Please give a descriptive subject for your e-mail. For example, the following is a good subject for a student in lecture Section 1.

```
Subject : CSC 35 – 1
```

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** *object source*	Don't mix up the *object* and *source* fields. It will destroy your program!
Link File	**ld -o** *exe object(s)*	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 the current 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.