California State University, Sacramento
College of Engineering and Computer Science

Computer Science 35: Introduction to Computer Architecture

Fall 2021 – Lab 6 – *The Great Hall*

Overview

Hogwarts School of Witchcraft and Wizardry has many wonderous things… and one of them is the feasts. Oh, the feasts!

And the desserts! Oh my, the desserts are incredible! Hogwarts offers a wide selection of sugary snacks – from ice cream sundaes to banana-split sundaes to delicious shakes.

Life is good!

Department of Student Health

The Ministry of Magic, naturally, wants to prevent the students from getting sick – the result of eating too much candy, ice cream, butter beer, etc…

So, the wizards & witches in the Department of Student Health (Officially: *The Department for the Overly-Enthusiastic Promotion of Student Health and Ridiculously-Fervent Regulation of Sugary Intake*) are getting involved. They want to prevent the students from having too much sugar. This much to the chagrin of both the School and the students.

As a result, they have ordered that all dessert plates must be enchanted. The plates will limit the number of portions a student can have.

$$\textbf{Allowed Portions} = \frac{\textbf{Total Allowed Sugar}}{\textbf{Total Sugar per Portion}}$$

Your Task

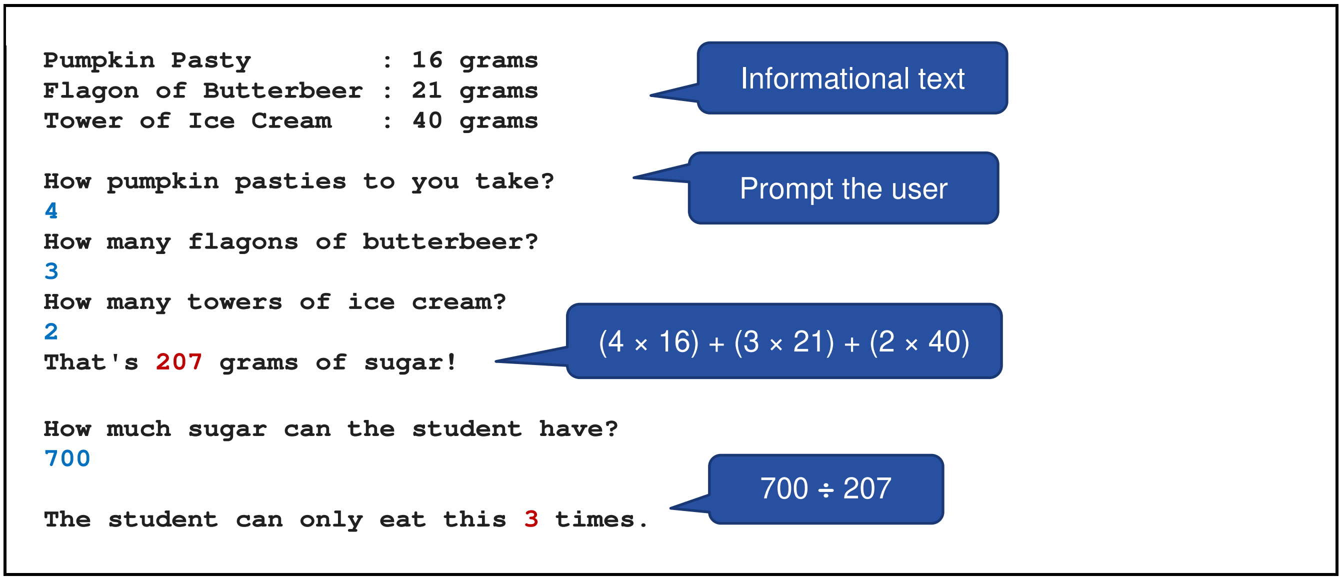
You will write a program that:

- calculates how much sugar a portion will contain
- determine how many of these the student will be allowed to eat

Once the total sugar is calculated, you need to calculate how many the student will be allowed to eat. Input the total daily sugar allowed and use the equation above.

Sample Run

The following is a sample run of the program. 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. *Please feel free to change the wording of the text.*



```
Pumpkin Pasty        : 16 grams
Flagon of Butterbeer : 21 grams
Tower of Ice Cream   : 40 grams

How pumpkin pasties to you take?
4
How many flagons of butterbeer?
3
How many towers of ice cream?
2
That's 207 grams of sugar!

How much sugar can the student have?
700

The student can only eat this 3 times.
```

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. Store each inputted value using direct storage
2. Prompt the user for each of the 3 desserts the student ate.
3. Compute the total sugar.
4. Output this value to the screen with some helpful text.
5. Prompt the user for the student's daily sugar limit and input the value.
6. Compute the number of rounds they can order

Hints

- Start off by getting the first calculation to work and print the correct value.
- For multiply, I strongly recommend using the two-operand version
- Now work on each of the requirements below one at a time. You will turn in the final program, but **incremental design is best for labs**.

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.

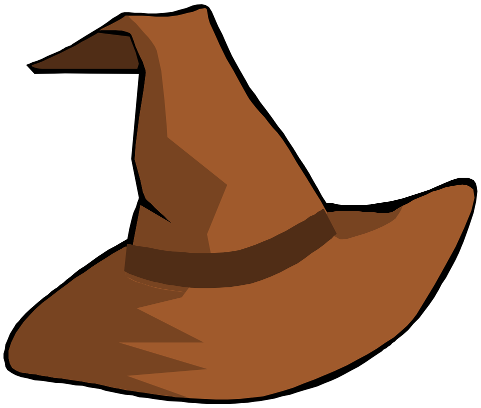
```
alpine
```

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.