Designs 4 (Character strings, and data structures)

Graded course work

Important Notes:

- **Write all the programs, from the designs in this handout**
- **Use the same program names, and variable names that I have specified**
- **Compile, run, and test your programs**
- **Submit a copy of these programs to me for grading**
- **If you don't know how to do any particular part, ask me -- I will show you how**

Program (`height1.c`)

Write a program that will ask for the names of two people, and their heights in inches.

The program will then produce the output below.

Program Output:

```
Please enter the first person's name: John
Please enter the first person's height (inches): 68

Please enter the second person's name: Henry
Please enter the second person's height (inches): 65

John is the tallest, at: 5 foot, 8 inches.
Henry is the shortest, at: 5 foot, 5 inches.

Press any key to continue_
```

My version uses the variables:

```
char first_name[25],
     second_name[25];

int first_height = 0,
    second_height = 0;
```

Program (`to_upper2.c`)

Based on programs: `string2.c`, and `to_upper1.c`, write a program that takes in a lowercase string, and prints out the uppercase version.

Program (`student4.c`)

From the lecture handout, complete the program `student4.c` to read in, and print out data for all four fields of the data structure.

Program (`numbers3.c`)

Based on program `numbers2.c`, write a program that will read in a number ("zero" - "three") in English, and print out the integer value.

Entering the string "quit" will cause the program to end.

REMEMBER:

- To include the `string.h` header file
- That `strcmp()` is used to compare strings

Program output:

```
Please enter a number in English: one
1

Please enter a number in English: zero
0

Please enter a number in English: three
3

Please enter a number in English: quit

Press any key to continue_
```

Program (`area3.c`)

Based on programs: `area1.c`, and `for1.c`, write a program that will calculate the total square footage of a house.

The program will ask the user how many rooms there are in the house, then room by room ask for the width, and length of each room.

My version of the program uses these variables:

```
int loop_counter = 0,
    number_of_rooms = 0;

float room_width = 0,
      room_length = 0,
      square_footage = 0;
```

Program output:

```
How many rooms are in the house: 4

 How wide is room 1 (feet): 12
 How long is room 1 (feet): 18

 How wide is room 2 (feet): 24
 How long is room 2 (feet): 10

 How wide is room 3 (feet): 10
 How long is room 3 (feet): 12

 How wide is room 4 (feet): 40
 How long is room 4 (feet): 20

 The 4 rooms in the house, have a total square footage of: 1376.00

Press any key to continue_
```

Program (`menu2.c`)

This program is a program for taking food orders at MacWesty Burgers. When the program runs, it offers the menu:

```
Menu Choices
1: MacWesty Burger.
2: MacWesty Cheese Burger.
3: MacWesty Fries.
4: MacWesty Onion Rings.
5: MacWesty Shake.
0: Print Bill.
```

If the user enters 1, the program will offer:

```
How many MacWesty Burgers? _
```

When a number is entered for the item, the program will return to the menu.

The program will keep a running total of each item ordered, and when menu choice 0 is entered, the program will show:

```
==================================
Your order is:
2 MacWesty Burgers: $3.98
1 MacWesty Cheese Burgers: $2.10
3 MacWesty Fries: $2.25
1 MacWesty Onion Rings: $1.00
3 MacWesty Shakes: $3.75
For a total price of: $13.08
```

My version of the program is about 100 lines long, and uses these variables, and constants:

```
const float burger_price = 1.99f,
            cheese_burger_price = 2.10f,
            fries_price = 0.75f,
            onion_rings_price = 1.00f,
            shake_price = 1.25f;

float total_price = 0;

int menu_choice = 0,
    number_of_burgers = 0,
    number_of_cheese_burgers = 0,
    number_of_fries = 0,
    number_of_onion_rings = 0,
    number_of_shakes = 0;
```

TIPS:

- Use a `do - while()` loop
- Look at programs: `menu1.c`, `total2.c`

Program (`times3.c`) *Extra Credit*

Extend the programs `times2.c` so that the program will ask the user for the answer on each pass through the loop.

If the user gives the correct answer the program will say so, and ask for the next answer.

If the user gives the wrong answer the program will say so, and re-ask the question.

You should also look at the program `add_test2.c`

Program output:

```
The Multiplication Table Program

Which multiplication table do you want to try? 3
* * * * * * * * * * * * * *
What is:
3 times  1 = 3
Right answer!

* * * * * * * * * * * * * *
What is:
3 times  2 = 4
Wrong answer! -- Try again.

What is:
3 times  2 = 6
Right answer!

* * * * * * * * * * * * * *
What is:
3 times  3 = 6
Wrong answer! -- Try again.
```

My version of the program uses these variables:

```
int times = 0,
    number_of_loops = 0,
    the_answer = 0,
    flag = 0;
```