COMP1200-C - Assign 04
Due midnight – Wednesday – February 15
Submit `assign04a.c, assign04b.c` and `devPlan04(.txt or .pdf)` **via Blackboard**

This assignment continues to demonstrate how to approach solving a large problem by solving one smaller part at a time. By solving a smaller part correctly before adding the next, one can keep the number of statements and errors that may result from them to a minimum. The addition requirements demonstrate how an existing problem may change in scope and thus the solution program must be modified. By saving the solution files with an incremental file name, additional versions can easily be saved using subsequent names providing good backup files.

NOTE: Your submitted file(s) MUST be spelled and cased as instructed.

PART a:

***Before you start writing your program**:*

Using the information in these instructions and previous devPlans, fill in the Software Development Plan (`devPlan04.txt`) to plan your solution to the assign04a assignment problem. If you do not use Notepad to edit the `devPlan04.txt` file, save it as a `.pdf` for submission.

Save your assign03.c as assign04a.c

Read all instructions before beginning your work.

***Program*: assign04a.c**

A growing number of people are becoming more health conscious watching their diet and exercising more. Statistics are available to help them monitor their progress toward becoming a healthier person. The body mass index (BMI) is a statistical measure which compares a person's weight and height. Though it does not actually measure the percentage of body fat, it is used to estimate a healthy body weight. The following equation is used to calculate body mass index. Note: The mass is the body weight in kilograms and height is in meters. [http://en.wikipedia.org/wiki/Body_mass_index]

When relying on user input, data validation is important to ensure that correct information is enter. Use a `do_while` loop to validate each user input. This requirement needs to be added to this program for each value entered: height (59-78), weight (90-350), gender (M,m,F,f), and target BMI (18.5-30.0). Each range is inclusive.

***Additional Problem Constants*:**
None.
***Additional Problem Inputs*:**
None
Additional Problem Outputs:
None
Additional Other variables:
None

- ☐ INPUT
  - Until good weight, prompt the user to enter a value for *weight in pounds*
  - Until good height, prompt the user to enter a value for *height in inches*.
- ☐ BMI
  - Compute conversions.
  - Calculate.
  - Display *BMI*.
  - Display *BMI classification*
- ☐ Target BMI and weight
  - Until good BMI, prompt the user to enter a value for *BMI*.
  - Compute *weight*.
  - Compute conversions.
  - Display *weight*.
- ☐ IBW
  - Until good gender, prompt user to enter the *gender* (F/f or M/m).
  - Compute *IBW* for given *height* and *gender*.
  - Compute conversions
  - Display *IBW*.

New commands
`do…while`

Sample Input/Output:

```
Enter the height in inches(59-78): 50
Enter the height in inches(59-78): 80
Enter the height in inches(59-78): 72
Enter the weight in pounds(90-350): 85
Enter the weight in pounds(90-350): 400
Enter the weight in pounds(90-350): 165

The BMI is: 22.38
BMI Classification: Normal

Enter the target BMI(18.5-30.0): 18
Enter the target BMI(18.5-30.0): 31
Enter the target BMI(18.5-30.0): 25

The target weight is: 184.33

Is the person a female or male? Enter F or M: x
Is the person a female or male? Enter F or M: m

The ideal weight is 171.08 pounds.
```

PART b:

Before you start writing your program*:*

Add the assign04b requirements to your devPlan04 Development Plan to plan your solution to the assign04b assignment problem. Save your assign04a.c as assign04b.c and add the new requirements.

***Program*: assign04b.c**

Once you have all the previous requirements implemented correctly for one person, you are ready to add a loop to run the program for a given number of people using a `for` loop.

***Additional Problem Constants*:**
None.
***Additional Problem Inputs*:**
number of people
Additional Problem Outputs:
None
Additional Other variables:
counter

New commands
`for`
nested loops

Algorithm:

- ☐ Prompt user for the number of people
- ☐ for each person enter and compute stats
    - INPUT
        - While not a good weight, prompt the user to enter a value for *weight in pounds*
        - While not a good height, prompt the user to enter a value for *height in inches*.
    - BMI
        - Compute conversions.
        - Calculate.
        - Display *BMI*.
        - Display *BMI classification*
    - Target BMI and weight
        - Until good BMI, prompt the user to enter a value for *BMI*.
        - Compute *weight*.
        - Compute conversions.
        - Display *weight*.
    - IBW
        - While not a good gender, prompt user to enter the *gender* (F/f or M/m).
        - Compute *IBW* for given *height* and *gender*.
        - Compute conversions
        - Display *IBW*.

Sample Input/Output:

```
Enter the number of people: 2

Enter the height in inches(59-78): 60
Enter the weight in pounds(90-350): 145

The BMI is: 28.32
BMI Classification: Overweight

Enter the target BMI(18.5-30.0): 25
The target weight is: 128.01

Is the person a female or male? Enter F or M: F
The ideal weight is 100.31 pounds.

Enter the height in inches(59-78): 72
Enter the weight in pounds(90-350): 165

The BMI is: 22.38
BMI Classification: Normal

Enter the target BMI(18.5-30.0): 25
The target weight is: 184.33

Is the person a female or male? Enter F or M: M
The ideal weight is 171.08 pounds.
```

General Instructions:

- ☐ Insert comments at the top and throughout each file
  - o Include the follow comments at the beginning of this (and ALL) files.
    - // your name
    - // assignment number
    - // date you completed the assignment
    - // **statement(s) about collaboration**
    - // a short narrative about what the file does
  - o Use the algorithm as comments throughout each file
- ☐ Use descriptive variable names.
- ☐ Use Sample Input/Output as a guide.
- ☐ Use descriptive labeling with output.

-7 points per file for absence of any of these required comments at the top

- ☐ Divide you solution program code into sections as noted in the algorithm. Use section comments as well as the algorithm step comments.
- ☐ Assign01b.c used **`scanf`** for a character; notice that there is a space in front of **`%c`**.

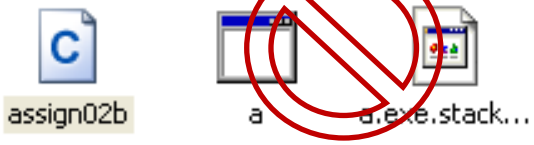
```
scanf(" %c", &gender);
```

- ☐ Indent all blocks. Use CSD.

Submit via Blackboard:

`assign04a.c`	C program file
`assign04b.c`	C program file
`devPlan04.txt` or `.pdf`	Development plan

If you do not use Notepad to edit the `devPlan04.txt` file, save it as a .pdf for submission.



NOTE: Your submitted file(s) MUST be spelled and cased as instructed.