

CS 100 Project Six – Spring 2019

Project Overview: In this project, you will collect the height and weight from a group of subjects in order to calculate BMI (Body Mass Index). The collected data from each subject will be saved in the following structure.

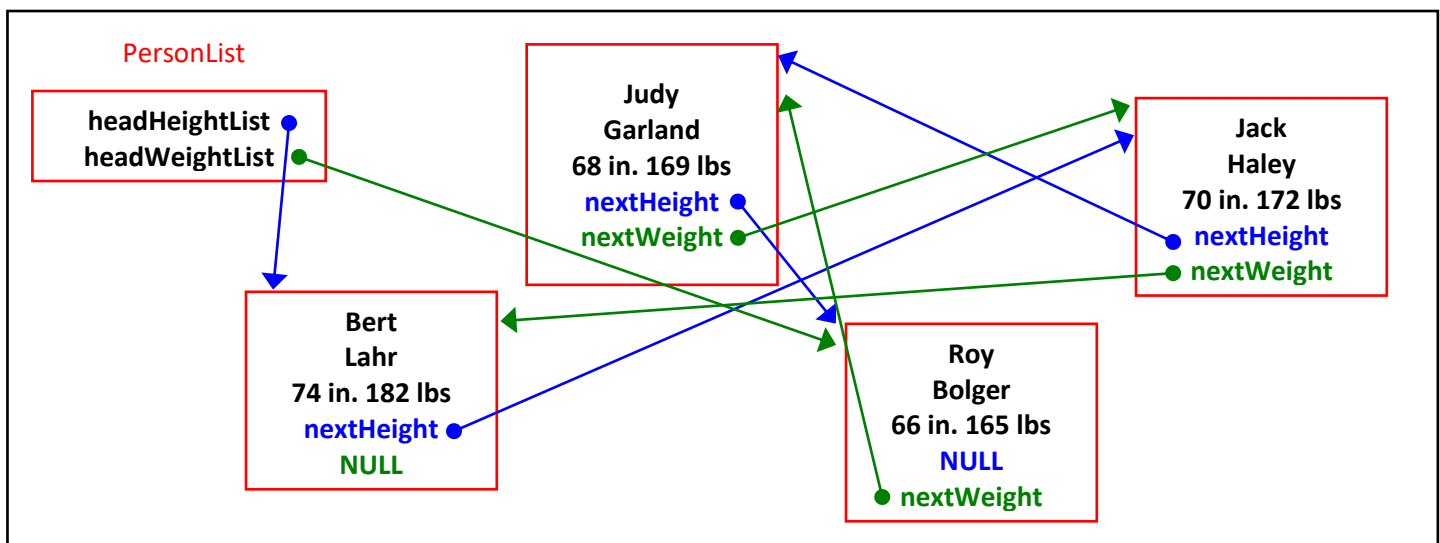
```
typedef struct _person {
    char *first;           // first name of the person
    char *last;            // last name of the person
    int height;            // height in inches
    int weight;            // weight in pounds
    struct _person *nextHeight; // next person in list, ordered by height
    struct _person *nextWeight; // next person in list, ordered by weight
} Person;
```

In addition to the data, there are two “next” fields (**nextHeight** and **nextWeight**) in a node to put the node into two linked lists. One linked list is ordered by the height in descending order using **nextHeight** whereas the other linked list is ordered by the weight in ascending order using **nextWeight**. Details on how to break a tie will be described later. To represent these two lists, a second structure is defined below to store the two heads of these two lists.

```
typedef struct _plist {
    Person *headHeightList;
    Person *headWeightList;
} PersonList;
```

The following example shows 4 nodes representing 4 persons. Each node (or person) will be in two linked lists. The linked list in blue is ordered by the height in descending order and the linked list in green is ordered by the weight in ascending order. Specifically

- The blue list ordered by the height: Bert Lahr → Jack Haley → Judy Garland → Roy Bolger
- The green list ordered by the weight: Roy Bolger → Judy Garland → Jack Haley → Bert Lahr



This project consists of the following three files.

- **main.c** – the user interface code. This prompts the user for an action and performs that action.
- **bmi.h** – definitions of the two structures and related function signatures.
- **bmi.c** – implementation of the functions listed in **bmi.h**.

You can download a working version of **main.c** and **bmi.h** and a skeleton (empty) **bmi.c** file. You must complete all the functions in **bmi.c**. Note that the downloaded **bmi.c** file has code for **add** that does a simple “add at front” for both links. This will allow you to work on some of the basic functions (**size**, **printByHeight**, **printByWeight**) before changing it to the more complicated ordered **add**.

What You Need To Do

- Create a directory **project6**. Download **main.c**, **bmi.h**, and **bmi.c** to that directory.
- Please compile the project using `gcc -Wall -std=c99 main.c bmi.c`
- The file **bmi.c** shall contain the actual code for the nine functions that are used by the main function. Right now there is no code in each function except **add**. Complete these nine functions.

1. **size** – return the number of persons in the list.
2. **printByHeight** – prints the persons in the list ordered by the height.
3. **printByWeight** – prints the persons in the list ordered by the weight.
4. **search** – takes a name (including first and last) and returns the address of the node representing such a person or NULL if there is no person by that name.
5. **updateName** – updates the name for the individual specified.
6. **add** – creates a new node to represent a new person and adds the new node to both ordered linked lists. To break the tie, remember to put a tall person before a short one and a light person before a heavy one. If two persons have the same height and weight, put a new person after an existing one. We recommend you write two helper functions to compare two persons (a **new** person and an **existing** person) to determine the order of these two persons in the two sorted linked lists. Note you can create only one new node in this function. If you create two new nodes and add them to the two lists respectively, a severe penalty will be assessed. Do not add a person if the name already exists. Instead print out an error message to indicate that. See the details on when to print error messages below.
7. **removePerson** – removes the specified person from the list.
8. **updateHeight** – updates the height of the individual specified. See the **add** function on how to break a tie. The person with newly updated height (or weight in **updateWeight**) can be viewed as a **new** person when comparing two persons.
9. **updateWeight** – updates the weight of the individual specified.

- We recommend building this project one function at a time. We will grade each of them individually, so you can get partial credit even if you don’t complete them all. We recommend the following order when completing these functions, but you can complete them in any order that you want.
 - **size**, **printByHeight**, and **printByWeight** (using the original “add-at-front” add)
 - **add** (ordered add)
 - **search** and **updateName**
 - **removePerson**
 - **updateHeight** and **updateWeight**
- You can assume all the input to these functions will be valid, but print out an error message when encountering the following situations.
 - **add** for a name that already exists (both first name and last name match)
 - **updateName** and **updateHeight** and **updateWeight** and **removePerson** for a name that does not exist
- Input for this program is entered from standard input. The **main.c** does not completely check the input validity, please input very carefully. You can enter all the commands in a file, say **testData**, as shown at the right, then redirect input from a file using `./a.out < testData` or `./a.out no < testData` (with or without prompts).

testData

```
add
Judy
Garland
68
169
add
Bert
Lahr
74
182
size
add
Roy
Bolger
66
165
add
Jack
Haley
70
172
size
printByHeight
printByWeight
search
Judy
Garland
updateName
Judy
Garland
Jill
Garland
updateHeight
Jill
Garland
71
printByHeight
quit
```

When you are finished and ready to submit your project:

- Compress your **project6** directory into a single (compressed) zip file, **project6.zip**.
- Once you have a compressed zip file named **project6.zip**, submit that file to Blackboard.

Project 6 is due at 5:00pm on Friday, April 19. Late projects are not accepted.

This document including its associated files is for your own personal use only.

You may not post this document or a portion of this document to a site such as chegg.com without prior written authorization.

**A project shall be completed individually, with no sharing of code or solutions.
All submissions will go through MOSS (Measure Of Software Similarity) for similarity check.
The University of Alabama's Code of Academic Conduct will be rigorously enforced.**

Execution of the Program with the Sample testData file

```
Enter a command or help to get a command list: add
    Enter first name: Judy
    Enter last name: Garland
    Enter height: 68
    Enter weight: 169

Enter a command or help to get a command list: add
    Enter first name: Bert
    Enter last name: Lahr
    Enter height: 74
    Enter weight: 182

Enter a command or help to get a command list: size

There are 2 persons in the list

Enter a command or help to get a command list: add
    Enter first name: Roy
    Enter last name: Bolger
    Enter height: 66
    Enter weight: 165

Enter a command or help to get a command list: add
    Enter first name: Jack
    Enter last name: Haley
    Enter height: 70
    Enter weight: 172

Enter a command or help to get a command list: size

There are 4 persons in the list

Enter a command or help to get a command list: printByHeight
Bert Lahr: height=74, weight=182
Jack Haley: height=70, weight=172
Judy Garland: height=68, weight=169
Roy Bolger: height=66, weight=165

Enter a command or help to get a command list: printByWeight
Roy Bolger: height=66, weight=165
Judy Garland: height=68, weight=169
Jack Haley: height=70, weight=172
Bert Lahr: height=74, weight=182

Enter a command or help to get a command list: search
    Enter first name: Judy
    Enter last name: Garland
Judy Garland: height=68, weight=169

Enter a command or help to get a command list: updateName
    Enter first name: Judy
    Enter last name: Garland
    Enter new first name: Jill
    Enter new last name: Garland

Enter a command or help to get a command list: updateHeight
    Enter first name: Jill
    Enter last name: Garland
    Enter new height: 71

Enter a command or help to get a command list: printByWeight
Roy Bolger: height=66, weight=165
Jill Garland: height=71, weight=169
Jack Haley: height=70, weight=172
Bert Lahr: height=74, weight=182

Enter a command or help to get a command list: quit
```