CSE 102 Programming Assignment 5

DUE

December 19, 2023, 23:55

Description

- This is an individual assignment. Please do not collaborate
- If you think that this document does not clearly describes the assignment, ask questions before its too late.

For this assignment, you are expected to write an interactive program which, according to the user input, creates dynamic responses and do simple calculations.

A compound object will be defined interactively and you program will keep track of a simple properties and return a result based on the composition of the object.

Example:

Suppose that you want to calculate the total calories in a meal. For this example, your compound object is a meal. A meal will have different components(ingredients). Some of the components will have sub-components. So we can describe the meal with a hierarchical structure. We can visualize it as a tree. Be careful, this assignment is not about trees. You are not going to construct trees.

In this example, the meal is an hamburger. Lets see what are the ingredients(components):

```
- Hamburger
   - Patties
      - SoybeanOil (10g)
      - GroundBeef (170g)
   - Buns
      - FlaxSeeds (5g)
      - WheatFlour (50g)
      - Sugar (5g)
   - Tomatoes (40g)
   - Pickles (35g)
   - Mayonnaise
      - Eggs (5g)
      - Butter
         - Milk (2g)
         - ButterFat (10g)
```

Your program will ask about these ingredients and calculate the total calories in this meal. Below is a simple dialog:

```
> What is the meal?:
> Hamburger
> What is Hamburger?:
> Patties, Buns, Tomatoes(40g), Pickles(35g), Mayonnaise, Ketchup
> What is Patties?:
> SoybeanOil(10g), GroundBeef(170g)
> List the amount macro nutrients in SoybeanOil:
> 0.0 0.0 99.0             //carb percentage, protein percentage, fat percentage
> List the amount macro nutrients in GroundBeef:
> 3.0 50.0 30.0
> What is Buns?:
> FlaxSeeds(5g), WheatFlour(50g), Sugar(5g)
> List the amount macro nutrients in FlaxSeeds:
> 20.0 40.0 40.0
> List the amount macro nutrients in WheatFlour:
> 90.0 10.0 0.0
```

```
> List the amount macro nutrients in Sugar:
> 100.0 0.0 0.0
> List the amount macro nutrients in Tomatoes:
> 25.0 5.0 0.0
> List the amount macro nutrients in Pickles:
> 5.0 3.0, 0.0
> What is Mayonnaise?:
> Eggs(5g), Butter
> List the amount macro nutrients in Eggs:
> 0.0 40.0 50.0
> What is Butter?:
> Milk(2g), ButterFat(10g)
> List the amount macro nutrients in Milk:
> 10.0 5.0 3.0
> List the amount macro nutrients in ButterFat:
> 0.0 0.0 100.0
> Total calories is A in Bg Hamburger. %C calories from carbs, %D from proteins, %E from fats.
```

- A: Total calories
- B: Total weight
- C: percentage of calories coming from carbohydrates
- D: percentage of calories coming from proteins
- E: percentage of calories coming from fats

Calculate and replace `A`, `B`, `C`, `D`, `E` with actual values.

In order to calculate calories, define the following at the beginning of the program.

```
#define CARB_CAL_1G   4          /* Number of calories in 1g carbohydrate */
#define PROTEIN_CAL_1G   4       /* Number of calories in 1g protein */
#define FAT_CAL_1G   9           /* Number of calories in 1g fat */
#define WATER_CAL_1G   0         /* Number of calories in 1g water */
```

Remarks

- You can assume that there won't be more than `20` components at each level.
- You can assume that the length of a component name cannot exceed `30` characters.
- You will calculate the result on-the-fly. You don't need a tree.
- Percentages will be given in the following order:
  - carb percentage, protein percentage, fat percentage
- **If the percentages of the macro nutrients do not add up to `100`, you can assume that the rest is water.**
- There may be spaces in between the terms of the user input. Below are valid inputs:

```
Eggs(5g),                    Butter
        5.0         3.0       0.0
```

- If there is only one component, there won't a comma(`,`) following it. Example:

```
Hamburger
```

- The leaf nodes(components) will have numbers next to them. that number is the amount in grams. Example: This component is a leaf component. There won't be a sub-component of it. It is 2g of Milk.

```
Milk(2g)
```

- **You are not allowed to use global variables.**
- Do not submit your code without testing it with several different scenarios. `Hamburger` and its structure are presented as an example only. Your program should run if the user tries to describe another meal. For example: A salad, manti etc...
- You can use c structs, unions, arrays, c strings, pointers, dynamic memory allocation, etc...
- Write comments in your code.

- If your code does not compile you will get `0`
- Do not share your code with your classmates.
- Do not print anything other than the expected output.
- You cannot use anything which is not covered in class.
- Do not submit any of the files you used for testing.

Turn in:

- Source code of a complete C program. Name of the file should be in this format: `<full_name>_PA5.c`.
- Example: `Lucius_Vorenus_PA5.c`. Please do not use any Turkish special characters.
- You don't need to use an IDE for this assignment. Your code will be compiled and run in a command window.
- Your code will be compiled and tested on a Linux machine(Ubuntu). GCC will be used.
- Make sure you don't get compile errors when you issue this command : `gcc <full_name>_PA5.c`.
- A script will be used in order to check the correctness of your results. So, be careful not to violate the expected output format.
- Provide comments unless you are not interested in partial credit. (If I cannot easily understand your design, you may loose points.)
- You may not get full credit if your implementation contradicts with the statements in this document.

Late Submission

- Late submission is NOT accepted.

Grading (Tentative)

- `Max Grade` : `100`.

All of the followings are possible deductions from `Max Grade`.

- No submission: `-100`.
- Compile errors: `-100`.
- Irrelevant code: `-100`.
- Major parts are missing: `-100`.
- Unnecessarily long code: `-30`.
- Using language elements and libraries which are not allowed: `-100`.
- Not caring about the structure and efficiency: `-30`. (avoid using hard-coded values).
- Significant number of compiler warnings: `-10`.
- Not commented enough: `-5`. (Comments are in English).
- Source code encoding is not `UTF-8` and characters are not properly displayed: `-5`. (You can use 'Visual Studio Code', 'Sublime Text', 'Atom' etc... Check the character encoding of your text editor and set it to `UTF-8`).
- Fails at asking question: `-30`.
- Fails at parsing leaf nodes: `-30`.
- The result is wrong: `-20`.
- Output format is wrong: `-30`. (Be careful with spacing)
- Infinite loop or recursion: `-90`.
- Prints anything extra: `-30`.
- Unwanted chars and spaces in the output: `-30`.
- Submission includes files other than the expected: `-10`.
- Submission does not follow the file naming convention: `-10`.
- Sharing or inheriting code: `-200`.
- IF YOU DON'T FOLLOW THE FILE NAMING CONVENTIONS YOU WILL GET `0`.

Note: Some of these items are not independent. So, you cannot expect isolation of many of them. For example, if you cannot read input file correctly, you will fail to produce the correct output file. Partial grading is not guaranteed.