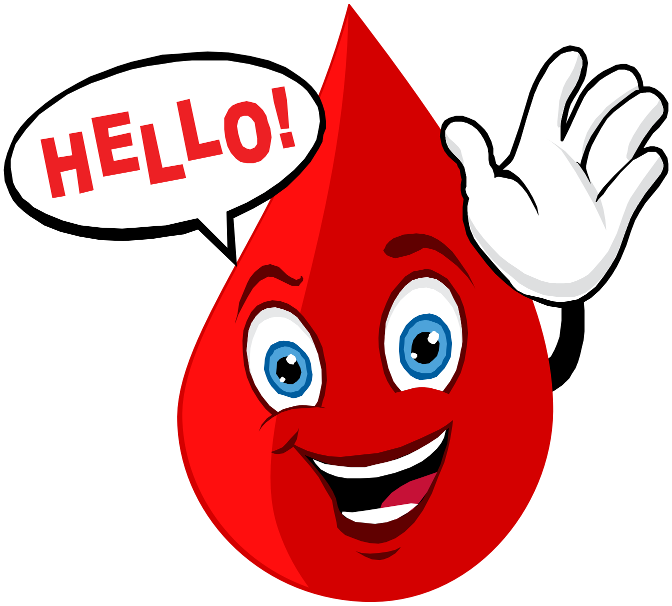
California State University, Sacramento
College of Engineering and Computer Science

Computer Science 35: Introduction to Computer Architecture

Spring 2018 – Lab 3 – *Sugar Math*

Overview



Dr. Acula needs your help! For patients with diabetes, it is important to keep track of – and to maintain – a healthy blood sugar level. If the blood sugar is too high, it can be detrimental to the patient's health and possibly life threatening. Fortunately, blood sugar can be easily controlled using insulin.

But, how much insulin does the patient need? To figure the number of "units", doctors have created some ingenious algorithms and equations. In particular, the doctor needs to know two things:

1. The patient's blood sugar.
2. How much of an effect a "unit" of insulin will have.

You will write a program that reads these two pieces of information and computes the number of "units" needed.

Math of Blood Sugar

A rough way to correct a high blood sugar level, is to compute their Insulin Correction Factor (ICF). This number is calculated by dividing 1800 by the total number of "units" a person takes during a day. The ICF is only important if a patient's blood sugar is high.

$$\text{ICF} = \frac{1800}{\text{total daily dose}}$$

Once that ICF is known, a patient's current blood sugar level can be used to compute the required dosage, in "units", to bring their levels down.

$$\text{dose} = \frac{\text{blood sugar level} - 150}{\text{ICF}}$$

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**. *Please feel free to change the wording of the text.*

```
How many daily units does the patient take?
60

The patient's ICF is: 30           1800 / 60

What is the patient's blood sugar level?
270

The patient needs 4 units.          (270 – 150) / 30
```

Requirements

You must think of a solution on your own. The requirements are as follows:

1. Prompt the user for the total daily units & input the answer

2. Compute the ICF

3. Output the ICF to the screen with some helpful text.

4. Prompt the user for the patient's blood sugar level & input the answer

5. Compute the dose

6. Output the dose to the screen with some helpful text.

Hints

- Start off by getting the first calculation to work and print the correct value.

- 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

Run Alpine by typing the following and, then, enter your username and password.

```
alpine
```

Please send an e-mail to yourself (on your Outlook, Google account) to check if Alpine is working. To submit your lab, send the source file (not a.out or the object file) to:

```
dcook@csus.edu
```

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` *objectfile asmfile*	Don't mix up the *objectfile* and *asmfile* fields. It will destroy your program!
Link File	`ld –o` *exefile objectfiles*	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 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.