CS120L – C Programming Language Labs
Fall 2014
Week 7: Loops, Conditional Expressions & Simple Problem Sovling

In this lab, you are not allowed to use any libraries other than `stdio.h`. 20 marks will be deducted if you use other libraries.

Task 1

Description: Print out a pattern of stars (*) with the height given by the user. The following example shows how the pattern for `height = 5` should look like:

```
***********
 *         *
  *       *
   *     *
    * * *
     * *
    * * *
   *     *
  *       *
 *         *
***********
```

For `height=0`, we would have:

```
*
*
```

For `height=1`, we would have:

```
***
 *
***
```

For `height=2`, we would have:

```
*****
 * *
  *
 * *
*****
```

Input: A single integer value, which is the user-provided non-negative height.
Output: The pattern as shown above.

Task 2

Write a program that can determine whether or not a 3-digit integer has the cube property. An integer has this property if the cube of its digits add up to the number itself. For example, 153 has this property:

$153 = 1^3 + 5^3 + 3^3 = 1 + 125 + 27 = 153$

The output of the program should look like this:

```
153 has the cube property. (1 + 125 + 27)
370 has the cube property. (27 + 343 + 0)
…
```

The program should accept a number from the user and display whether it has the cube property (Only accept 3 digit numbers). The program should run until 4 numbers with the cube property have been found.

Pseudo-code:

```
/*Prompt user for number*/
/*Determine if number is valid input*/
/* Extract the three digits */
/*    If the sum of the cubes of the digits */
/*          equals the number, print the number satisfies cubes property*/
/*    Else print the number does not satisfy cubes property*/
/*Repeat until four cubes have been found*/
```

Implementation

- The task should be implemented as a separate functions in a file named: `q2.c`.
- We are providing you the `q2_template.c` file where this function is called. The `main` function should not be updated.
- After you have implemented the function, you need to rename the file to `q2.c`.

Command line for compiling

```
/usr/bin/gcc -Wall -Wextra -ansi -pedantic q1(2).c -o q1(2).exe
```

Grading scheme

- 100 marks in total (50 marks each question).
- 10 marks for successful build.
- 10 marks for successful execution and correct output (2 marks * 5 hidden test cases).

  Note that your program should only print out the required output, ending immediately with an end-of-line character `\n`, and not any redundant character.

- 20 marks for correct implementation.
- 10 marks for good programming style. For the complete guide, please see the Assignment Guideline on CS120 page.

  In general, you should have clear variable names, reasonable comments to explain your code, and consistent indentation. Please use **2-blank spaces** instead of tab spaces.

Output formatting

You should use the given sample input and output files to check your output format before submission. After compiling your program into **a.exe**, you can generate your output using:

```
./a.exe < q1.in > result.out
```

Then compare it with the sample output by typing:

```
diff result.out q2.out
```

You need to make sure that `diff` does not print out anything. Otherwise, your output is considered wrong and you will lose all the 10 marks for output correctness.

Submission

Please name your source code for question 1 as **q1.c**, question 2 as **q2.c**, put them in a folder named **cs120<session>_<your Digipen login id>_<labnumber>**, in which <**labnumber**> is 7 this week and <session> is either a or b, and zip them in **cs120<session>_<your Digipen login id>_<labnumber>.zip** for submission.

Wrong submission file/folder name will cause 10 marks deducted.

Note that from this lab onwards, the file and folder names must be **lowercase**.

The deadline of submission is 17th October 23:59 and late submission will receive zero mark.