

CS120L – C Programming Language Labs
Fall 2014
Week 6: Loops & Conditional Expressions

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: Create a table of [Fibonacci numbers](#) and the corresponding Fibonacci quotients. Recall: the Fibonacci sequence is defined as follows:

$$\begin{aligned} F_0 &= 0 \\ F_1 &= 1 \\ F_N &= F_{N-1} + F_{N-2} \text{ for all } N \geq 2 \end{aligned}$$

So this gives: 0,1,1,2,3,5,8,13,21,34,55,etc.

The **Fibonacci quotient** is defined as:

$$Q_N = F_N / F_{N-1} \text{ for all } N > 1$$

The interesting thing about the Fibonacci quotient is that as N gets very large, Q_N approaches the following [Golden Ratio](#):

$$\varphi = \frac{1 + \sqrt{5}}{2} \approx 1.6180339887498948482045868343656 \dots$$

You will have to use some conditional and looping statements, of course. Pay attention to the formatting. You will need to use the `diff` tool and `redirection` that were shown in the last lab.

Once you've got it working, you should experiment with it to see what happens if you provide bogus input. Can you go larger than 46? What happens? How do you fix that? Etc. Just "play" with it to satisfy your curiosities.

Under the resources folder is an executable that provides the basic functionality described. You need to make sure that your executable has the same behavior as the provided executable.

These are the steps (algorithm) that you need to implement in C code.

1. Declare the necessary variables that you will need. (Hint: use a `double` for the Fibonacci quotient and an `int` for everything else.)
2. Display the prompt and accept the user input into one of the variables.
 - a. You'll need some kind of loop that repeats until the user enters a valid number.
3. Print the column headings.
4. Print the first two values. (These are special and don't require any calculations.)
5. Calculate the other values and print them
 - a. You'll need a loop here as well that runs from 2 to the number that the user provided.
 - b. You need to calculate the Fibonacci number and the Fibonacci quotient. (Watch out for integer division.)

- The below output will be shown if the user enters 12.

Enter the Fibonacci number to calculate.
The number must be an integer between 2 and 46. (Enter 1 to quit):

N	Fibonacci number	Fibonacci quotient
0	0	N/A
1	1	N/A
2	1	1.0000000000000000
3	2	2.0000000000000000
4	3	1.5000000000000000
5	5	1.6666666666666667
6	8	1.6000000000000001
7	13	1.6250000000000000
8	21	1.6153846153846154
9	34	1.6190476190476191
10	55	1.6176470588235294
11	89	1.6181818181818182
12	144	1.6179775280898876

You need to use **left justification** for all three columns. Use the sample outputs to check the field size for each column.

- The program should repeatedly ask for the number until the user enters an appropriate value:

Enter the Fibonacci number to calculate.
The number must be an integer between 2 and 46. (Enter 1 to quit): 0
Enter the Fibonacci number to calculate.
The number must be an integer between 2 and 46. (Enter 1 to quit): -4
Enter the Fibonacci number to calculate.
The number must be an integer between 2 and 46. (Enter 1 to quit): 59
Enter the Fibonacci number to calculate.
The number must be an integer between 2 and 46. (Enter 1 to quit): 1

Task 2

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.

Hint: For any question that asks you to print out a pattern based on a user-input value, the first thing to consider is whether the output contains a single or multiple rows. If the output contains multiple rows, it's likely that you will need to use two nested loops, the outer one for the rows, and the inner one for the columns on each row.

The second thing to notice is the relation between the characters on each row, and the row ID. For example, in this task, when `height = 5`, the pattern has this form:

Row #1 has 0 space, followed by 11 stars.
 Row #2 has 1 space, followed by 9 stars.
 Row #3 has 2 spaces, followed by 7 stars.
 Row #4 has 3 spaces, followed by 5 stars.
 Row #5 has 4 spaces, followed by 3 stars.
 Row #6 has 5 spaces, followed by 1 star.

...

Do you recognize the relation between the row ID and the number of blank spaces?

How about between the row ID and the number of stars on each row? Note that:

Row 1: $11 = 5 \times 2 + 1$

Row 2: $9 = 4 \times 2 + 1$

Row 3: $7 = 3 \times 2 + 1$

Row 4: $5 = 2 \times 2 + 1$

Row 5: $3 = 1 \times 2 + 1$

Row 6: $1 = 0 \times 2 + 1$

Note: Similar to the first task, you should keep reading in inputs until the input is non-negative.

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 6 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 10th October 23:59 and late submission will receive zero mark.