Programming and Operating Systems

CS211: Getting started with C

01 March 2021

This session is designed to help your practice your programming skills, and to get working with a suitable compiler/IDE. Ideally, you should install `Code::Blocks` (if you are using Windows or Linux) or `xcode` if you are a Mac user.
However, for this will, it will suffice to use an online compiler, such as

- `https://www.onlinegdb.com/online_c_compiler`
- `http://www.tutorialspoint.com/compile_c_online.php`
- `http://cpp.sh/`
- `https://www.codechef.com/ide`

1. Download the `04Fibinacci.c` example from
   http://www.maths.nuigalway.ie/~niall/CS211/Week03/
   As explained in the notes from Week 3, it computes the first ten terms in the Fibonacci Sequence: $f_0 = 1, f_1 = 1$, and

   $$f_{k+1} = f_k + f_{k-1} \text{ and for } k = 1, 2, \ldots.$$

   Read the code and make sure you understand it all. Compile it and run it.
2. Modify the code in the following way:
   (a) `Fib` is declared to be an `int` array of length 100.
   (b) The user is prompted to enter a value for n that is between 2 and 100.
   (c) The program computes and outputs the first n terms in the sequence, with f_n stored in `Fib[n]`.
3. You should find that we can't actually compute f_{100} successfully: it is too large to store as an `int`. By experimenting with n, find the largest value of n for which f_n can be computed.
4. Modify your code so that it preforms some basic input checking: the entered value of n should be no less than 2 and no more than the largest value of n that you found in Q3.

5. Unless you change the code, you should find that results are not nicely tabulated. E.g, the first 12 lines look like this:

```
Fib[0] = 1
Fib[1] = 1
...
Fib[5] = 8
Fib[6] = 13
Fib[7] = 21
Fib[8] = 34
Fib[9] = 55
Fib[10] = 89
Fib[11] = 144
```

Add a width field to the format specifiers so that the output looks like

```
Fib[ 0] =   1
Fib[ 1] =   1
...
Fib[ 5] =   8
Fib[ 6] =  13
Fib[ 7] =  21
Fib[ 8] =  34
Fib[ 9] =  55
Fib[10] =  89
Fib[11] = 144
```

6. There are many other sequences in that one could study, for example the *Collatz Sequence* which is defined as follows: Set C_0 be any positive integer you choose, and

$$C_{k+1} = \begin{cases} C_k/2 & \text{if } C_k \text{ is even} \\ 3C_k + 1 & \text{if } C_k \text{ is odd.} \end{cases}$$

For example, if $C_0 = 10$, then the sequence is

$$\{10, 5, 16, 8, 4, 2, 1, 4, 2, 1, 4, 2, 1, \dots\}.$$

There is a famous conjecture that claims, not matter what you choose for C_0, eventually it will fall into a "4–2–1" cycle.
Write a programme that prompts the user to enter C_0 and n, which may be at most 500. It then computes the first n terms in the sequence. It should output the computed terms, and then also report the minimum number of steps required for one first terms to be 1, 2 or 4.

You **do not** have to submit anything this week. But try to make your code as good as possible. In particular, it should have comments at the start explaining what it does and how. If you would like some feedback, send your code to Niall.Madden@NUIGalway.ie