For this assignment you will be writing 3 programs that work with the Fibonacci series. The first 2 numbers of the series are 1 and 1. To compute the next value you add the last two together. For example, the first 8 Fibonacci numbers are:

1, 1, 2, 3, 5, 8, 13, 21

The Fibonacci series starts to generate very large numbers, very quickly. **You will want to use a long or a double data type to hold the numbers...**

Problem 1 (40 Points): Write a POSIX program (on UNIX) that uses 3 processes to print the Fibonacci series to a file. Your main process will ask the user for a single integer between 1 and 50. This input will be the number of Fibonacci numbers the program will generate. After accepting the input, the main process shall start two child processes that are connected via a single pipe.

Computation Process: This process will compute the first N Fibonacci numbers (where N is the number the user entered). After generating each number, it should send it along the pipe to the printing process to print it. **Note:** you should be able to do this entire program without using an array to store the Fibonacci numbers!

Printing Process: This process first create an output file called fib-N.txt, where N is the number the user originally entered. You do not need to check if the file already exists - you may simply overwrite it if it does. It will then read a number from the pipe and simply write it out to the output file. After all the numbers have been generated, the process should close the output file and exit. *You may use any method you like to make sure the printing process knows when to stop reading.*

The relationship between the 3 processes is shown below:
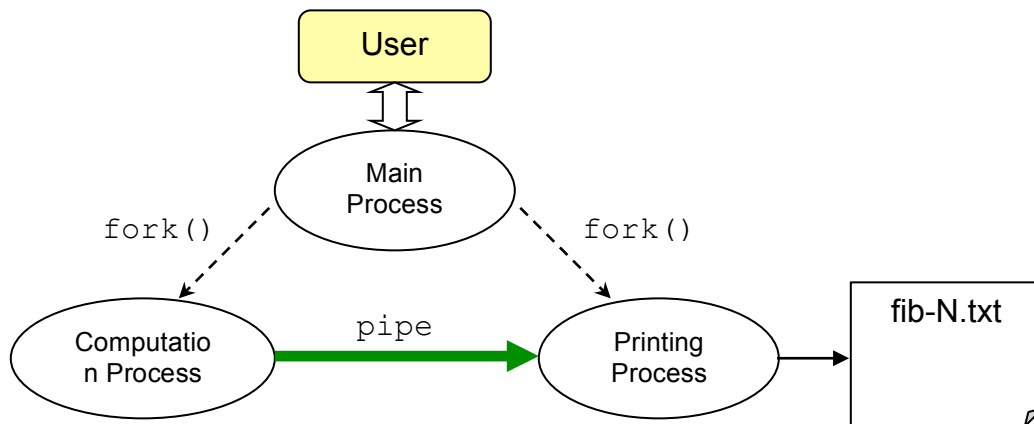
After starting the two additional processes and linking them together with the pipe, your main process should wait for each of its children to terminate. After they are done, the main process should again ask the user for another integer between 1 and 50 and repeat the whole process. If the user enters a -1, your program should quit.

Sample Input and output: (User input in red)
```
Enter an integer between 1 and 50:   4
First 4 fibonacci numbers were written to fib-4.txt
Enter an integer between 1 and 50:   5
First 5 fibonacci numbers were written to fib-5.txt
Enter an integer between 1 and 50:   -1
Goodbye.
```

Contents of Fib-4.txt

1:	1
2:	1
3:	2
4:	3

Contents of Fib-4.txt

1:	1
2:	1
3:	2
4:	3
5:	5

Problem 2 (30 Points): Write a **Win32** program that mimics the functionality of Problem #1 in but uses only 2 threads instead of 3 processes. The main thread should be in a loop that accepts input *n* – either a non-negative integer (*n*) or -1 to terminate the program. Every time the user enters a non-negative integer the program should create a separate thread that computes the Fibonacci series **and** writes (the entire series) to a file called fib-n.txt where n is the number the user has entered. The main thread should wait for the worker thread to complete before asking the user for another integer. Note, there should be no use of pipes in this program…
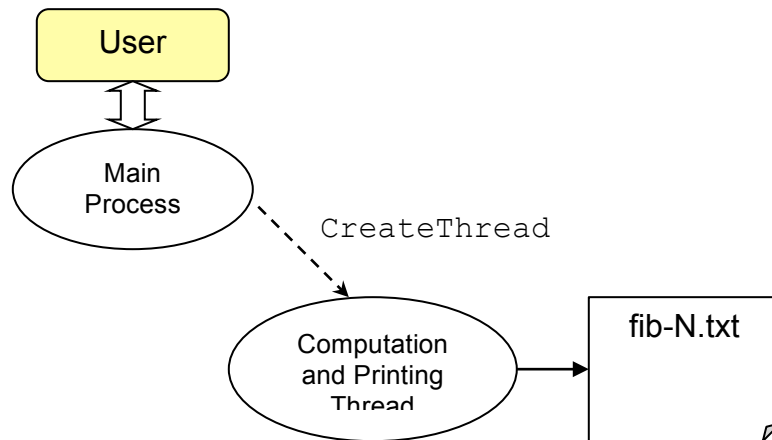
Problem 15 (30 Points): Write the same program as in Problem 2, this time using **POSIX's** *pthread* library. To compile on unix you may need to use the following command (assuming your source code is in prob3.cpp and the executable will be prob3)

```
g++  -o prob3 prob3.cpp -lpthread
```

Problem 2 is to be written and run on Windows – all other problems in this assignment require the POSIX environment (unix/linux).

Submitting your programs: Submit your programs using moodle (attachments). Your submission should include 3 source files: fib1.c, fib2.c and fib3.c (or .cpp for C++).