CS3520 – Programming in C++
Fall 2015
Assignment 1

Preliminaries

Assignments should be submitted as a single `.zip` file containing source files and a makefile to Blackboard. Assignments must build and run on the CCIS Linux environment using `g++`. Assignments should include a makefile that builds all the program executables by default, and a `clean` target that removes everything but the source files and makefile. Each program should be in a single source file that builds into an executable of the same name (for example, `fizzbuzz.cpp` builds to the executable `fizzbuzz`). Assignments that are missing a makefile or a have makefile that does not build the programs will receive a 0.

Assignment

Write the following programs in C++. For programs where no input is requested, use appropriate control structures to produce the output, do not simply hardcode the output.

Program 1: fizzbuzz

This program prints out the numbers between 1 and 100 on each line; however, for numbers that are multiples of 3 it prints "Fizz" instead of the number, numbers that are multiples of 5 it prints "Buzz" instead of the number, and for numbers that are multiples of both 3 and 5 it prints "FizzBuzz" instead of the number.

For example, the first 15 lines of output look like:

```
1
2
Fizz
4
Buzz
Fizz
7
8
Fizz
Buzz
11
Fizz
13
14
FizzBuzz
```

Program 2: fib
This program prints out the first 25 Fibonacci numbers. Fibonacci numbers are defined as $F_i = F_{i-1} + F_{i-2}$, with $F_0 = 1$ and $F_1 = 1$. Use a recursive function to compute the Fibonacci numbers.

For example, the first 5 lines of output look like:
```
1
1
2
3
5
```

Program 3: sum
This program reads whitespace-separated integers from standard input until end-of-file, and then prints the sum of the integers. Any inputs that cannot be parsed as integers are treated as if they are 0.

For example, given input like:
```
5
4
3 2
1
```

The program prints:
```
15
```

Program 4: reverse
This program reads in lines of text from the file named `reverse_input.txt` until end of file, and then prints each line in reverse order with the characters in each line reversed. If the input file cannot be opened or read, the program prints `File error` to standard error and exits.

For example, given input like:
```
123
abc
456
```

The program prints:
```
654
cba
321
```

Program 5: frame
This program reads in lines of text from the file named `frame_input.txt` until end of file, and then prints the text that was read in with a square frame of `*` characters around the text, horizontally separated by at least one space. If the input file cannot be opened or read, the program prints `File error` to standard error and exits.

For example, given input like:

```
a
ab
abc
```

The program prints:

```
*******
* a   *
* ab  *
* abc *
*******
```

Grading
Each program is worth 20% of the assignment, broken down as:

- 70% - Functionality: Does the program handle inputs correctly? Does it handle error cases gracefully? Does the program not crash?
- 30% - Style: Is the code well-structured with appropriate functions? Are the variable names suitably descriptive? Does the code have explanatory comments?