CS-102 Design Principles: Project 1

Due date: Monday, January 30 at 11:00pm

Projects will generally consist of a single task, but because we are just getting started, we thought it best to give you two smaller tasks. We will provide a github link for the assignment on Discord after you have had lab this week. Until then, feel free to start coding! Later, you can make certain small adjustments to your file names and such in order to submit the assignment. See Discord for starter code for each part.

---

1 The two projects

1. **The golden ratio.** The *Fibonacci sequence* is

$$1, 1, 2, 3, 5, 8, 13, 21, 34, 55, \ldots,$$

an infinitely long sequence that begins with a pair of 1s and where subsequent entries are the sum of the previous two. It is named for Leonardo of Pisa (whose nickname was Fibonacci), an Italian mathematician who used the sequence in his book *Liber Abaci*[1], published in 1202. The sequence was also, in fact, considered by poets, centuries earlier, who wrote in the Sanskrit language.

Although the sequence seems simple, it has yielded many lovely results to those who study it. One such fact about the sequence emerges from its consecutive entries. If we calculate the ratio of each pair of adjacent numbers, using the second of the pair as the numerator and the first as the denominator, we get

$$\begin{aligned} 1 \,/\, 1 &= 1.0 \\ 2 \,/\, 1 &= 2.0 \\ 3 \,/\, 2 &= 1.5 \\ 5 \,/\, 3 &= 1.666666667 \\ 8 \,/\, 5 &= 1.6 \\ 13 \,/\, 8 &= 1.625 \\ 21 \,/\, 13 &= 1.615384615 \\ 34 \,/\, 21 &= 1.619047619 \\ 55 \,/\, 34 &= 1.617647059, \text{ and so on} \end{aligned}$$

The resulting sequence of ratios can be shown[2] to approach the number

[1] translation: *The Book of Calculation*

[2] see ϕ, π, e *and* i (MAA Spectrum Series, 2017) by David Perkins for more details

$$\frac{1+\sqrt{5}}{2} = 1.61803398874989484\ldots,$$

known as *the golden ratio* and denoted by φ. In this part of the project, you will write a program that asks the user for a *tolerance* (a small positive number close to zero) and then returns the first ratio that comes within that tolerance of the golden ratio.

Input: A tolerance t such that $0.0000000001 < t \leq 0.5$.

Output: A pair of Fibonacci numbers x and y such that the ratio x/y is the first such ratio that is within t of φ; the numbers should be printed on one line separated by a space.

A run of your program should follow this simple format:

```
Enter a tolerance: 0.001
55 34
```

This is the correct output because the difference between 55/34 (see table above) and φ is approximately 0.00038, which is less than the given tolerance 0.001. The previous ratio 34/21 is not within this tolerance.

You should write your solution in `golden_ratio.cpp`, which is provided on Discord. Some starter code is already provided for you, including:

`const double GOLDEN_RATIO` = 1.618033988749895

so that everyone is using the same approximation to φ.

If you want to use the absolute value function, use `#include <cmath>`, whereupon `abs(x)` will yield the absolute value of `x`.

2. **Pascal's Triangle.** Computer scientists often make use of a set of numbers called *binomial coefficients* that can be arranged like so:

This arrangement was studied by French mathematician Blaise Pascal in the mid-1600s. There is a formula for each entry in the triangle; to understand it, you must first understand that if j is a positive integer, then $j!$ is read aloud as "j factorial" and it equals

$$j! = 1 \times 2 \times 3 \times \cdots \times j$$

For example, $4! = 1 \times 2 \times 3 \times 4 = 24$. We define $0! = 1$.

With this in mind, entry k in row n (in each case, start counting at zero) is found by

$$\frac{n!}{k! \times (n-k)!}$$

For example, entry 2 in row 6 equals

$$\frac{6!}{2! \times (6-2)!} = \frac{720}{2 \times 24} = 15$$

Write a program that asks the user for a single positive integer, and outputs the corresponding row of Pascal's Triangle.

Input: A positive integer n between 0 and 20 (inclusive).

Output: Row n of Pascal's Triangle, all on one line with a single space between numbers. There should *not* be a space after the final number.

A run of your program should follow this simple format:

```
Enter n: 7
1 7 21 35 35 21 7 1
```

You should write your solution in `pascals_triangle.cpp`, which you can find on Discord.

Note (in the starter code) that we are requiring you to write two functions, and that we provided the full declaration for one of them. You should complete the other declaration.

2 Your grade

Your final grade will come from:

80%	passing the tests
20%	style

There are 12 tests for this project. Projects passing zero tests will receive at most 70% of the project grade (assuming perfect style). Passing tests increases the grade; however, this assumes that your code isn't written to circumvent testing (for example, by "hard coding" answers to the tests), which is a violation of the honor code.

The elements of style that we will assess on Project 1 are:

- ☐ proper indentation
- ☐ correct placement of curly braces
- ☐ reasonable variable names
- ☐ completed file and function header comments

The file header comments should look like this:

```
/**
 * Creates a random array of integers 0-9 and queries the
 * user for one such integer; the program searches the array
 * and reports if the user's integer is found.
 * @file search.cpp
 * @author(s) Ayesha Rascoe
 * @date September 12, 2022
 */
```

Your main function can simply have a header comment like this:

```
// Controls the flow of the program:
int main() {
    // some code here
}
```

Your function header comments should look like this (if your function does not take parameters, simply leave out the @param lines, and if your function is void, simply leave out the @return line):

```
/**
 * Returns true if the given integer pick is found
 * in the array numbers.
 * @param numbers: the array to search.
 * @param pick: the integer to search for in the given array.
 * @return true if pick is found in the array, and
 *         false otherwise.
 */
bool is_found_in_array(int numbers[NUM_NUMBERS], int pick) {
    for (int i = 0; i < NUM_NUMBERS; i++) {
        if (numbers[i] == pick) {
            return true;
        }
    }
    return false;
}
```

In the function above, you see examples of proper indentation, correct placement of curly braces, and reasonable variable names.