CS 31 Solutions Week 3

This worksheet is entirely **optional**, and meant for extra practice. Some problems will be more challenging than others and are designed to have you apply your knowledge beyond the examples presented in lecture, discussion or projects. Although exams are online this quarter, it is still in your best interest to practice these problems by hand and not rely on a compiler.

Solutions are written in red. The solutions for **programming problems** are not absolute, it is okay if your code looks different; this is just one way to solve the specific problem.

If you have any questions or concerns please contact your LA or go to any of the LA office hours.

Concepts: Loops, If Statements, Cin, Variables, Doubles, Ints

Reading Problems

1. This code snippet tries to print all prime numbers between 3 (inclusive) and a given input *n* (exclusive). Find the 3 bugs contained in the code and fix them.

```
int n;
cin >> n;
for (int candidate = 3; candidate < n; ++candidate) {
        bool isPrime = true;
        for (int x = 2; x < n; x++) {
                if (candidate % x = 0){
                        isPrime = false;
                }
        }

        if (isPrime) {
                cout << n << " ";
        }
}
```

```
int n;
cin >> n;
// check the candidates: the integers in the interval of [3, n)
for (int candidate = 3; candidate < n; ++candidate) {
    bool isPrime = true;  // a flag for whether the candidate is
prime or not
    for (int x = 2; x < candidate; x++) {
        // if any number in between 2 (inclusive) and the
candidate
        // (exclusive) is the factor of the candidate,
        // then the candidate is not a prime.

        if (candidate % x == 0){
            isPrime = false;
        }
    }

    if (isPrime) {
        cout << candidate << " ";
    }
}
```

Programming Problems

1. Write a program that takes in a number as an int and outputs the sum of all of the digits in that number.

Sample Output:
Enter a number: 184
The sum of the digits in your number is 13!

```
#include <iostream>
using namespace std;

int main() {
    cout << "Enter a number ";
    int num;
    cin >> num;
    int sum = 0;
    while (num > 0) {
        sum += num % 10;
        num /= 10;
    }
          cout << "The sum of the digits in your number is " <<
sum << "!" << endl;
}
```

2. Write a program that takes in N numbers and writes their mean.

Sample output:

```
How many numbers do you want to average? 5
Number: 4
Number: 2
Number: 8
Number: 9
Number: 7
The average is  6!
```

```
#include <iostream>
using namespace std;

int main() {

      cout << "How many numbers do you want to average?";
      int n;
      cin >> n;
      double num;
      double total = 0;

      for (int i = 0; i < n; i++) {
            cout << "Number: ";
            cin >> num;
            total += num;
      }
      cout << "The average is " << total/n << "!" << endl;
      return 0;
}
```

3. Write a program that reads in an integer N and prints an NxN box where the (i,j)th character is as follows:

'.' if j > i
i + j otherwise

Where i is the row number and j is the column number (starting at 0, not 1). For Example, if the input is 4, it should print:

```
      0  .  .  .
```

```
    1 2 . .
    2 3 4 .
    3 4 5 6
```

```
int main() {
    int n;
    cout << "Enter a positive integer: ";
    cin >> n;
    for (int i = 0; i < n; i++) {
        for (int j = 0; j < n; j++) {
            if (j > i)
                cout << ". ";
            else
                cout << i + j << " ";
        }
        cout << endl;
    }
}
```

4. Write a program that reads in an integer and prints whether that number is a perfect number. A perfect number is defined as a number that is equal to the sum of all positive factors excluding itself.

Example:

4 != 1 + 2	=> Print “Not perfect.”
5 != 1	=> Print “Not perfect.”
6 = 1 + 2 + 3	=> Print “Perfect.”
12 != 1 + 2 + 3 + 4 + 6	=> Print “Not perfect.”
28 = 1 + 2 + 4 + 7 + 14	=> Print “Perfect.”

```
#include <iostream>
using namespace std;

int main() {
    int perfect;
    cout << "Enter a number: ";
    cin >> perfect;
    int sum = 0;                // store the sum of the factors

    for (int i = 1; i < perfect; i++) {
        if (perfect % i == 0)           // Found a factor
```

```
                sum += i;
        }

        if (sum == perfect)
            cout << "Perfect." << endl;
        else
            cout << "Not perfect." << endl;
        return 0;
}
```

5. Write a program that takes in an integer N where N > 0, and outputs all its factors, each one separated by a comma. There should be no comma before the first number of after the last one.

Sample input:
12
Sample output:
1,2,3,4,6,12

```
int main() {
    int n;
    cout << "Enter a number: ";
    cin >> n;

    cout << "1"; // 1 is always a factor
    for (int i = 2; i <= n; i++) {
        if (n % i == 0)
            cout << "," << i;
    }
    cout << endl;
}
```

6. Write a program that given an input integer N, finds an integer x such that 2^x <= N < 2^{x+1}. The program should ask for user input and print the integer x it finds. If there exists no such x, it should print “error” and return 1.

Sample Input:
200 => Should output 7, since 2^7 = 128 <= 200 < 2^8 = 256.
20 => Should output 4, since 2^4 = 16 <= 20 < 2^5 = 32.
8 => Should output 3, since 2^3 = 8 <= 8 < 2^4 = 16.

```
#include <iostream>
using namespace std;

int main() {
    int number;
    cout << "Enter a number: ";
    cin >> number;

    if (number < 1) {     // check if the number is smaller than 2^0
        cout << "error" << endl;
        return 1;
    }

    int pow = 0, powOf2 = 1;
    while (powOf2 <= number) {
        pow++;
        powOf2 *= 2;
    }
    cout << pow - 1 << endl;   // pow - 1 bc while loop terminates
                          // once powOf2 > number, but we want
                          // powOf2 to still be less than
                          // number
    return 0;
}
```

7. The Fibonacci series consists of the integers 0, 1, 1, 2, 3, 5, 8, … . With the initial values $n_1 = 0$ and $n_2 = 1$ it is possible to find the next number, because the next number is related to the preceding two by the formula $x_{n+1} = x_n + x_{n-1}$. For example, 1+1 = 2, the next number in the series. Based on this information, write a program that receives an integer n as an input and prints the nth Fibonacci number. What is the 10th one?
Your program should also check whether the integer provided is valid. If the user inputs zero or a negative number, the program should print *Error: The input must be positive* and nothing more.
If you haven't done so already, try to write the program using a do-while loop.

10th Fibonacci number: 34

```
int main() {
        int n;
        cin >> n;
        if (n <= 0){
             cout << "Error: The input must be positive" <<endl;
             return 1;
        }
        if (n == 1)
             cout << 0 << endl;
        else if (n == 2)
             cout << 1 << endl;
        else {
             int xnext;
             int x_last1 = 1;
             int x_last2 = 0;
             for (int i = 2; i < n; i++) {
                  xnext = x_last1 + x_last2;
                  x_last2 = x_last1;
                  x_last1 = xnext;
             }
             cout << xnext;
     }
}
```