Worksheet 2

Problem 1.

This code snippet tries to print all prime numbers between 3 and a given input `n`. Find the three 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 << " ";
  }
}
```

Problem 2.

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

Problem 3.

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

Problem 4.

Write a program that reads in an integer `N` and prints an `N x N` 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
```

Problem 5.

Write a program that reads in an integer and prints whether that number is a perfect number.

(Hint: A perfect number is defined as a number that is equal to the sum of all factors excluding itself)

Problem 6.

Write a program that takes in an integer `N` where `N > 0`, and outputs all its factors, each one separated by a comma.

Problem 7.

Write a program that takes in an input integer `N` and 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`".

Problem 8.

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 equation (1).

$$F_n = F_{n-1} + F_{n-2} \quad (1)$$

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.

Note: We have not done do-while loops in lecture yet, so I will post the solutions later.

Solution to Problem 1.

This code snippet tries to print all prime numbers between 3 and a given input `n`. Find the three 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 < candidate; x++) {
    if (candidate % x == 0){
      isPrime = false;
    }
  }
  if (isPrime) {
    cout << candidate << " ";
  }
}
```

- **Quick note:** Bool is another data type like int and double. It's used to store data types that are either true or false.

I've marked the errors in red above:

1. `n` should be `candidate` since this section tries to see if `candidate` is divisible by any number other than 1 and itself, thus making it composite

2. This is not an assignment statement

3. We are outputting `candidate`, not the original number you typed in

Solution to Problem 2.

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

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

int main()
{
  // This section asks for a number input
  int number;
  cout << "Please type a number: ";
  cin >> number;

  // This section declares everything used in the while-loop
  int sum, remainder, newNumber;
  sum = 0;
  newNumber = number;

  while (newNumber > 0)
  {
    remainder = newNumber % 10;
    sum += remainder; // sum = sum + remainder
    newNumber /= 10;
  }
  cout << "The sum of all digits of the number you entered is " << sum << endl;
}
```

Solution to Problem 3.

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

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

int main()
{
  // N is the amount of numbers
  // n corresponds to the nth number you type in
  // dataNumber is the actual number you type in
  // sum is the sum of "dataNumbers"

  int N;
  cout << "How many numbers do you want to compute the mean of? ";
  cin >> N;

  if (N == 0)
    cout << "Enter a number greater than 0" << endl;

  else
  {
    double dataNumber, sum;
    sum = 0;
    for (int n = 1; n <= N; n++)
    {
      cout << "Number " << n << ": ";
      cin >> dataNumber;

      sum += dataNumber;
    }

    double average;
    average = sum / N;

    cout << "The mean of your " << N << " numbers is " << average << "." << endl;
  }
}
```

Solution to Problem 4.

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

int main()
{
  // N is the number of rows or columns
  // n is the row number
  // k is the number of remaining spaces that must be filled by periods
  // o is the oth period
  // sum is the actual number being shown on the box

  int N;

  cout << "What dimensions would you like the square box to be? ";
  cin >> N;

  for (int n = 1; n <= N; n++)
  {
    for (int sum = n - 1; sum <= 2*n - 2; sum += 1)
      {
        if (sum != 2*N - 2)
        {
          cout << sum << "  ";
        }
        else
          cout << sum << endl;
      }
    int k = (N - n);
    for (int o = 1; o <= k; o +=1)
    {
      if (o != k)
        cout << ".  ";
      else
        cout << "." << endl;
  }}}
```

Solution to Problem 5.

Write a program that reads in an integer and prints whether that number is a perfect number.

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

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

  int sum = 0;
  for (int candidate = 1; candidate < number; candidate +=1)
  {
    bool isMultiple = false;
    int product = 0;
    for (int multiplier = 1; product <= number; multiplier +=1)
    {
      product = multiplier * candidate;
      if (product == number)
        isMultiple = true;
    }
    if (isMultiple)
      sum += candidate;
  }
  if (sum == number)
    cout << "Perfect" << endl;
  else
    cout << "Not perfect" << endl;
}
```

Solution to Problem 6.

Write a program that takes in an integer `N` where `N > 0`, and outputs all its factors, each one separated by a comma.

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

int main()
{
  int N;
  int product;
  cout << "Enter a positive whole number: ";
  cin >> N;

  for (int factorCandidate = 1; factorCandidate <= N; factorCandidate += 1)
  {
    for (int multiplier = 1; multiplier <= N; multiplier += 1)
    {
      product = factorCandidate * multiplier;
      if (product == N)
      {
        if (factorCandidate != N)
          cout << factorCandidate << ", ";
        else
          cout << factorCandidate << endl;
}}}}
```

Solution to Problem 7.

Write a program that takes in an input integer `N` and 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`".

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

int main()
{
  int N;
  int x = 0;

  cout << "Enter a number: ";
  cin >> N;

  // Finds the value of x
  for (int n = 2; n <= N; n  *= 2)
  {
    x += 1;
  }

  int lowerBound = 1;

  if (x == 0)
    {lowerBound = 1;}

  if (x > 0)
    {for (int j = 1; j <= x; j++)
    {
      lowerBound = lowerBound * 2;
    }}

  int upperBound = lowerBound * 2;

  if ((lowerBound <= N) && (upperBound > N))
    cout << x << endl;
  else
    cout << "error" << endl;
}
```