CS 31 Worksheet 1

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. All exams will be done on paper, so it is 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.

Concepts

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

Reading Problems

1. Circle where the bug occurs and explain what incorrect behavior will happen. What do you think this program will output? Add a fix.

```
cout << "Enter your name: ";
getline( cin , name );

cout << "\nEnter your UID: ";
int UID;
cin >> UID ;

cout << "\nEnter your Major: ";
getline( cin , major );

cout << "\nEnter your residence hall: ";
getline( cin , hall );
```

Bug: It will skip “Enter your Major”, because getline has already consumed a newline character.

A newline is always appended to your input when you select Enter or Return when submitting from a terminal. It is also used in files for moving toward the next line. When the flow of control reaches std::getline(), the newline will be discarded, but the input will cease immediately. The reason this happens is because the default functionality of this function dictates that it should (it attempts to read a line and stops when it finds a newline).

Because this leading newline inhibits the expected functionality of your program, it follows that it must be skipped our ignored somehow. One option is to call cin.ignore(10000, '\n') after the the first extraction. It will discard the next available character so that the newline is no longer intrusive.

2. What is the output of the following code?

```
int a = 10;
int b = 22;
while (a / 2 >= 1) {
  a--;
  cout << a << endl;
  if ((a + b) % 2 == 0) {
    a--;
    cout << a << endl;
    b /= 2;
  }
}
```

9
8
7
6
5
4
3
2
1

3. This code snippet tries to print all prime numbers between 3 and a given input *n*. 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 < candidate; x++) {
          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 the 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;       // or  num = num / 10;
      }
      cout << "The sum of the digits in your number is " << sum <<
      "!" << endl;

}
```

2. Write a program that takes in N numbers to average and outputs the average of the N numbers.

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;
}
```

3. Write a program that takes in N numbers and outputs the sum of the even numbers.

Sample output:
How many numbers are you entering? 4
Enter a number: 2
Enter a number: 9
Enter a number: -3
Enter a number: 6
Result: 8

```
#include <iostream>

using namespace std;

int main() {
    cout << "How many numbers are you entering? ";
    int numbers = 0;
    cin >> numbers;

    int input_num = 0;
    int total = 0;
    while (numbers > 0) {
```

```
        cout << "Enter a number: ";
        cin >> input_num;
        if (input_num % 2 == 0) {
            total += input_num;
        }
        numbers--;
    }
    cout << "Result: " << total << endl;
}
```

4. Write a program that takes in two numbers and a command of type string ("Add", "Subtract", "Multiply", "Divide"). Inputting an invalid command should cause the program to ask for a valid command.

Sample output:
Enter your first number: 3
Enter your second number: 7
Enter your command: Multiply
Result: 21

```
int main() {
    int first = 0;
    int second = 0;
    string command = "";
    bool invalid_command = true;

    cout << "Enter your first number: ";
    cin >> first;
    cout << "Enter your second number: ";
    cin >> second;
    cin.ignore(1000000, '\n');

    while (invalid_command) {
        invalid_command = false;
        cout << "Enter your command: ";
        getline(cin, command);
        if (command == "Add")
            cout << "Result: " << first + second << endl;
        else if (command == "Subtract")
            cout << "Result: " << first - second << endl;
        else if (command == "Multiply")
            cout << "Result: " << first * second << endl;
        else if (command == "Divide" && second != 0)
```

```
            cout << "Result: " << first / second << endl;
        else {
            cout << "Invalid command!" << endl;
            invalid_command = true;
        }
    }
}
```

5. 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;
  }
}
```

6. 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 factors excluding itself.

Example:

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

```
int main()
{
    int perfect;
    cout << "Enter a number: ";
    cin >> perfect;
    int sum = 0;
    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;
}
```

7. Write a program that takes in an integer N where N > 0, and outputs a comma-separated list of all the factors of N.

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;
}
```

8. Write a program that given an input integer, N, finds a positive 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".

Sample Input:

200 => Should output 7, since $2^7 = 128 <= 200 < 2^8 = 256$.

20 => Should output 4, since $2^4 = 16 <= 20 < 2^5 = 32$.

```
int main() {
	int number;
	cout << "Enter a number: ";
	cin >> number;
	if (number < 1) {
		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;
}
```