Task 1

You should start by setting up our development environment the way you would like. Get Visual Studio, CLion or XCode installed, and write everyone's first C++ program; print "Hello World!" to the screen!

Task 2

Now lets try some IO. Here's a question you did in CS-UY 1114 for homework, now do it in C++:

Write a program that will read and evaluate a mathematical expression. The expression is of the form: oprand1 op operand2, where operand1 and operand2 are positive integers and op is an operator, which is either +, -, * or /.

For example: 24 + 65 and 276 * 2 are legal expressions.

Assumption: There is a single space between each operand and the operator.

Sample output (2 different executions):

```
Enter a mathematical expression: 5 + 10
5 + 10 = 15

Enter a mathematical expression: 81 / 9
81 / 9 = 9
```

Task 3

Can you work with files and functions?

We would like to design a password generator with easy to remember words. Since letters, alone, aren't secure enough, we should include numbers in there somewhere, either at the beginning or the end. You will write a function that reads the file provided, chooses one word, at random, and generates up to 3 random numbers to append or prepend onto the word.

Your function should allow for the user to choose if the numbers are appended, prepended, or, randomly placed (which should be the default). This will be decided by an integer passed into your function. A negative integer means prepend, positive means append and zero will mean you should choose (prepend or append) randomly.

We have provided a file with world of length between 5 and 7 characters. You should use words from this file.

Task 4

Design a system to keep track of employee data (up to 50 employees). The system should keep track of an employee name and ID number. This data will be stored in a file (user selectable) with the id number, hourly pay rate, and the employee's full name (example):

```
17 5.25 Daniel Katz
18 6.75 John F. Jones
```

Additionally we would like to be able to print payroll information from data in a different file. The data is the id number and a number of hours that they worked (example):

```
17 40
18 20
18 20
```

As you see we can have the same number listed twice in which case that person should be paid the sum of the numbers (John Jones did 40 hours work, but it's listed as 20+20).

After reading in both files (and storing the data in a single vector or array of structs) print the payroll information in order from the person paid the most to the person paid the least.