For all programming projects in this course you can assume that the user (the person using your program) will not try to do so in a way that violates the specifications. For example, if the specifications indicate that the program should expect an int, then you don't have to worry about getting something else, like a double or bool. Some specifications are implicit rather than explicit. For example in Project 2.a the description doesn't tell you which numeric type to use, but you can infer it from the given example. **There are several numeric types available, but for the programming projects you should stick with *int* for integers and *double* for floating-point numbers.**

Remember to not use any material we haven't covered yet (as stated in the syllabus).

Project 2.a

Write a program that asks the user for five numbers and then prints out the average of those numbers. When you run your program, it should match the following format:

```
Please enter five numbers.
-2.4
5.1
6.0
123.8
-19.0
The average of those numbers is:
22.7
```

The file must be named: **average.cpp**. To compile it into an executable file named "average", enter "g++ average.cpp -o average". "g++" is the name of the compiler, "average.cpp" is the name of your program, "-o average" says that you want the executable file to be named "average" (you do not have to match the name of your .cpp file). If you don't use the -o option, then the executable file will be named "a.out".

Project 2.b

Write a program that converts Celsius temperatures to Fahrenheit temperatures. The formula is:

$$F=95C+32$$

where F is the Fahrenheit temperature and C is the Celsius temperature. The program should prompt the user to input a Celsius temperature and should display the corresponding Fahrenheit temperature. It should display **only** the converted temperature without any text (not even a 'F'). When you run your program, it should match the following format:

```
Please enter a Celsius temperature.
-10.5
The equivalent Fahrenheit temperature is:
13.1
```

The file must be named **tempConvert.cpp**.

Project 2.c

Write a program that asks the user for a (integer) number of cents, from 0 to 99, and outputs how many of each type of coin would represent that amount with the fewest total number of coins. When you run your program, it should match the following format:

```
Please enter an amount in cents less than a dollar.
87
Your change will be:
Q: 3
D: 1
N: 0
P: 2
```

Hint: You will find the mod operator (%) and integer division useful.

The file must be named: **change.cpp**.