Exercise 1b - If- and else-statements Functions

During the lesson we went through how if-else-statements work. Let us write our first C++ program which makes use of them.

The if-statement needs a condition to validate (something to check whether it is true or not) but the else-statement should not have anything. The action that each statement should perform (printing out something) can be placed inside “{“ and “}” blocks, or if it is just one single line it can be placed without the “{“ and “}”

```
int number_of_volvos = 5;
int number_of_teslas = 6;

if (number_of_volvos < number_of_teslas)
{
   std::cout << "You should buy more Volvos!";
}
else
{
   std::cout << "You have enough Volvos for now.";
}
```

Since above program only uses one single line in the statements it can also be printed as:

```
int number_of_volvos = 5;
int number_of_teslas = 6;

if (number_of_volvos < number_of_teslas)
   std::cout << "You should buy more Volvos!";
else
   std::cout << "You have enough Volvos for now.";
```

But it is always recommended to use the “{“ “}”

If you run the code above, the if-statement will be true since 5 is less than 6.

However, if you would increase the number of Volvos to 6 or greater, then the if-statement will be false and the it will print out whatever is in the else-statement instead.

Exercise 1b

Write the code above and change the values of number_of_volvos and number_of_teslas so that you will get to see both print statements being printed out.