Program 1: Exploring data types

Write a C++ program to explore different data types and answer the questions.
You will use this program:

```
// ********************************************
// COSC 501 LAB #1
// YOUR NAME: Ahmed Hamza
// DUE-DATE
// PROGRAM-NAME: Lab1_1
// A simple description of the program
//*********************************************
```

Lab questions: (5 points each)

1. Why the results are different between int division in part A and float division in part C?

The results are different because of the data types being used:

- In part A, you are performing integer division (**i/j**), which results in an integer quotient. When you divide two integers, the fractional part is truncated, and only the integer part is retained. For example, **7 / 3** results in **2** because it's the largest integer that fits in the result.

- In part C, you are performing float division (**f1/f2**), which results in a floating-point number. Floating-point numbers can represent fractional parts, so the result includes the decimal portion. For example, **7.0 / 3.0** results in **2.33333...** because it preserves the fractional part.

2. Explain the result of the operations in Part B

In Part B, you are exploring the behavior of the increment operators (**i++** and **++i**):

- **i++** is the post-increment operator. It first uses the current value of **i** in an expression and then increments **i**. So, when you print **i++**, you see the current value of **i** (before the increment), and then **i** is incremented.
- **++i** is the pre-increment operator. It increments **i** first and then uses the updated value in an expression. So, when you print **++i**, you see the value of **i** after it has been incremented.

3. What is the integer value of a character?

The integer value of a character is its ASCII (or Unicode) value. In C++, you can get the integer value of a character by explicitly casting it to an integer using **int(char_variable)**. Each character is associated with a unique integer code that represents it. For example, the ASCII code for the character '7' is 55, '3' is 51, and 'A' is 65.

4. Is there any correlation between the memory size and the maximum value of an object of that type? Explain your answer.

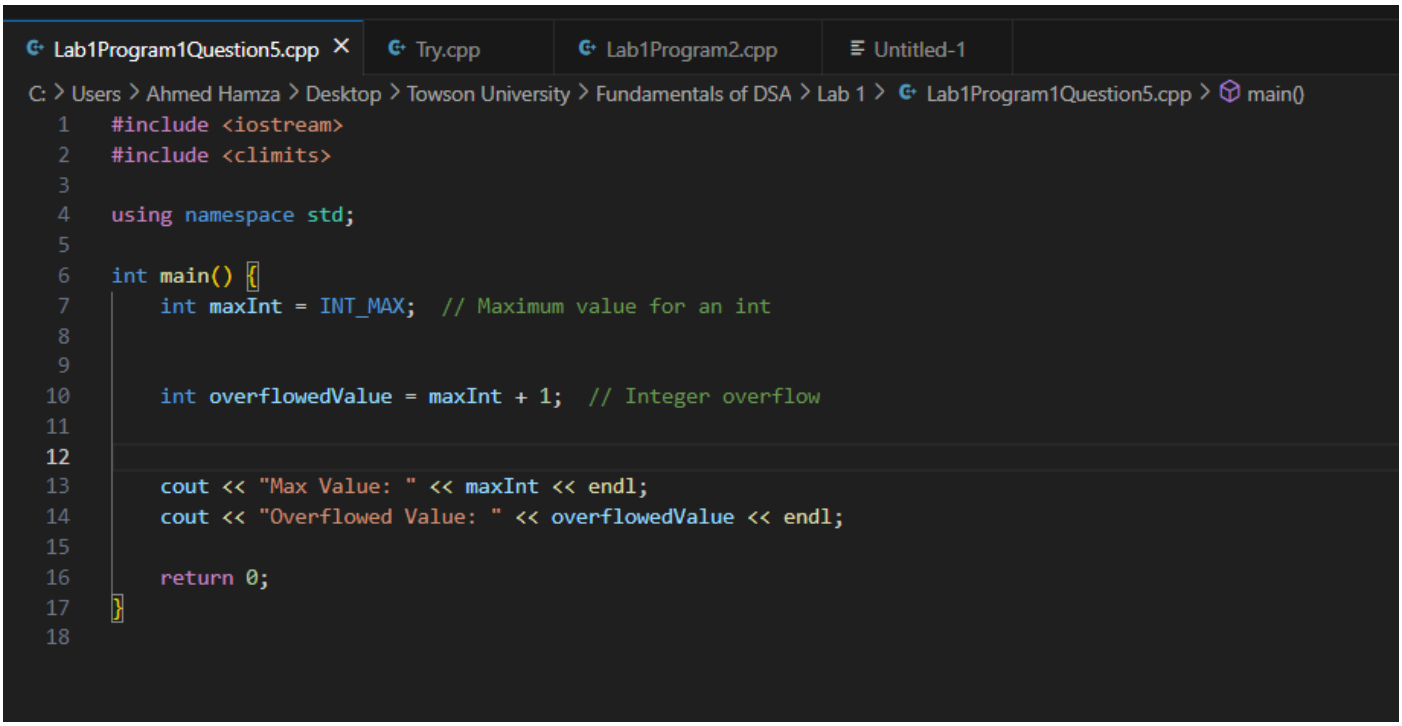
Yes, there is a correlation between the memory size and the maximum value of an object of that type, especially for integer types. The memory size determines the range of values an object of a particular type can hold. This is because the number of bits allocated for the data type affects the maximum and minimum representable values.
For example:

- An **int** typically uses 32 bits on most systems, allowing it to represent values from approximately -2 billion to +2 billion (assuming two's complement representation).
- A **long int** usually uses 64 bits, which provides a larger range for storing values compared to an **int**.
- A **float** uses 32 bits for representing floating-point numbers, which provides a certain range and precision.
- A **double** uses 64 bits for representing floating-point numbers, offering a larger range and higher precision compared to **float**.
- For character types like **char**, the memory size is typically 1 byte, and the integer value is the ASCII code of the character.

5. What is an integer overflow? Give an example.

An integer overflow occurs when the result of an arithmetic operation exceeds the maximum representable value for that data type. When an overflow occurs, the result wraps around to a minimum or negative value or exhibits unexpected behavior.

Example:



In this example, **maxInt** represents the maximum value that can be stored in an **int**. When we add 1 to it, integer overflow occurs, and the result wraps around to the minimum value for an **int**, which is **INT_MIN**. This behavior can lead to unexpected results and bugs in your program if not handled correctly.