

## CS201 GDB-1 Solution

BC\*\*\*\*\*

[TopicsLearn.com](https://www.topicslearn.com)

To store the number 51,147,483,647,321, we need to select a data type that can accommodate such a large value. Let's examine the available data types and determine the most suitable one:

- **Integer:** The largest integer data type in most programming languages is typically `long long int`, which can hold values up to 9,223,372,036,854,775,807 (assuming a 64-bit system). Unfortunately, this data type is not sufficient to store the given number, as it exceeds the maximum value it can represent.
- **Float:** The `float` data type is used to represent decimal numbers. However, it has a limited precision and cannot accurately represent such a large integer value. Additionally, it has a smaller range compared to integers, so it is not suitable for storing the given number.
- **Char:** The `char` data type is used to represent individual characters. It is not designed to store large numerical values, and its range is typically limited to -128 to 127 or 0 to 255, depending on whether it's signed or unsigned.
- **Double:** The `double` data type is a floating-point type that offers higher precision compared to `float`. However, similar to the `float` type, it has limited range and is not suitable for storing the given number.

Considering the limitations of the above data types, the most appropriate choice for storing the number 51,147,483,647,321 would be a data type that can handle large integers. For this purpose, we can use a "big integer" data type or a library that provides support for arbitrary-precision arithmetic. A common example of such a library is the "BigInteger" class in many programming languages, which allows for the storage and manipulation of arbitrarily large integers. By utilizing this class or a similar library, we can store the given number without losing any precision or exceeding the range of the data type.

In conclusion, since the given number exceeds the capabilities of standard data types like `long long int`, `float`, `char`, and `double`, we would need to use a “big integer” data type or a library that provides support for arbitrary-precision arithmetic to store the number accurately.

[CS302 GDB Solution](#)

[CS201p Assignment Solution](#)

[MGT211 Assignment No 1 Solution](#)

[ENG101 GDB Solution](#)

Powered by [Topics Learn](#)