Lecture Summary

Data types

Recall that a *data type* is set of values and set of operations on those values. It determines the possible values that can be stored in a variable and the operations that can be performed on that variable. It also determines the amount of memory required to store the value and the way the value is stored in the memory. Some commonly used data types in C++ are following[1]:

- **int** - stores integers (whole numbers), without decimals, such as 123 or -123. The `int` data type stores values that are within a range of -2,147,483,648 to 2,147,483,647 (32-bit signed integer).

  Other useful data types for integers are: `short` (stores 16-bit integers), `long` (stores 32-bit integers), `long long` (stores 64-bit integers).

- **double** - stores floating point numbers, with decimals, such as 19.99 or -19.99. The `double` data type is a double-precision 64-bit floating point.

  Other useful data types for floating point numbers are: `float` (stores 32-bit floating point numbers), `long double` (stores 80-bit floating point numbers).

- **bool** - stores values with two states: `true` or `false`

- **char** - stores single characters, such as `'a'` or `'B'`. The `char` values are surrounded by single quotes.

  The `char` is a type of integer, so it also stores numbers, which are interpreted as characters according to the ASCII encoding. For example, the ASCII value for the character `'a'` is 97 and the ASCII value for the character `'b'` is 98.

- **std::string** - stores text, such as `"Hello World"`. The `std::string` values are surrounded by double quotes.

[1] The sizes of these data types may vary from one compiler to another.

Some functions from the C++ library needed for the lab

function	description	example
`fmin()`	returns the minimum of two numbers	`int m = fmin(3, 5);` `// 3 is stored in m`
`fmax()`	returns the maximum of two numbers	`int m = fmax(3, 5);` `// 5 is stored in m`
`abs()`	returns the absolute value of a number	`int x = abs(-10);` `// 10 is stored in x`
`pow()`	returns the first argument raised to the power of the second argument	`double y = pow(2, 3)`
`sqrt()`	returns the square root of a number	`double s = sqrt(16);`
`ceil()`	returns the smallest integer that is greater than or equal to the argument	`int z = ceil(9.2);`
`floor()`	returns the largest integer that is less than or equal to the argument	`int z = floor(9.2);`
`lround()`	returns the long integer that is closest to the argument	`long z = lround(9.2);`
`rand()`	returns a random number	`int r = rand();`
`srand()`	sets the seed for the random number generator	`srand(time(0));`

To use the above functions (except the last two), you need to include the following header file in your program:

```
#include <cmath>
```

For the last two functions (`rand()` and `srand()`), you need to include the following header file in your program:

```
#include <cstdlib>
```

Lab Questions

1. Write a program `stats3.cpp` that prints three uniform random values between `0` and `RAND_MAX`, their average value, and their minimum and maximum value.

   Note:

   - Use `rand()` and `srand()` from the `<cstdlib>` library.
   - Remember average of three numbers could be a fraction, so use `double` for the average value.

2. Write a program `three_sort.cpp` that takes three `int` values from the user and prints them in ascending order.

   Note: Do not use `if` statement (we haven't covered it in lectures yet). Use `fmin()` and `fmax()` from the `<cmath>` library.

3. The *International Standard Book Number (ISBN)* is a 10-digit code that uniquely specifies a book. The rightmost digit is a checksum digit which can be uniquely determined from the other 9 digits from the condition that $d_1 + 2d_2 + 3d_3 \cdots + 10d_{10}$ must be a multiple of 11 (here d_i denotes the i-th digit from the right). The checksum digit d_1 can be any value from 0 to 10: the ISBN convention is to use the value `X` to denote 10.

   Example:
   The checksum digit d_1 corresponding to $d_{10}d_9d_8d_7d_6d_5d_4d_3d_2 = 020131452$ is 5 since it is the only value of d_1 between 0 and 10 for which $d_1 + 2*2 + 3*5 + 4*4 + 5*1 + 6*3 + 7*1 + 8*0 + 9*2 + 10*0$ is a multiple of 11.

   Write a program `isbn.cpp` that takes a integer from user, computes the checksum, and prints the 10-digit ISBN number $d_{10}d_9d_8d_7d_6d_5d_4d_3d_2d_1$ (when $d_1 = 10$, you may print as it is instead of printing `X`)

   Hint:
   If an `int` variable `x` contains a 9-digit integer, then you can access its individual digits as follows:

   ```
   int d2 = x % 10;
   int d3 = (x % 100) / 10;
   int d4 = (x % 1000) / 100;
   ```

   and so on.