Task 1 Types, Initialization and Reference

Basic Type

- Explain the output of the code `type.cpp`. What will happen to call `output_type( 31l )` ?
- Try to design a function `cmp_equal(a , b)` which takes two floating numbers and return true if $|a-b| < 10^{-6}$ and false otherwise. Do you need to design two versions for `double` and `float` ?
- Try to design a series of functions `cmp_less(a , b)` which takes two integers(unsigned or signed) and returns true if $a < b$ and false otherwise.

Initialization

- Quadratic

  You need to implement `quadratic.cpp` that takes real numbers a, b, c as input and calculate the solution of $ax^2 + bx + c = 0$. You need to meet the following requirements:

  - When there is no solution to the equation, output one line `No solution found!`.
  - If there are two solutions x_1, x_2, you should output **only one** solution x_1 if $|x_1 - x_2| < 10^{-6}$ (You can use `cmp_equal` in Basic Type). Otherwise, you should output the smaller solution and the other one in one line and separate them by a single space.
  - Try to implement these things with functions and learn how to use `std::pair` to pass the results to the `main` function. You can also use `auto` for initialization to make it easier. You should at least learn `std::pair` before class.

- Assume that there is a function that solves the equation:

  ```
  std::pair<double,double> solve_quadratic( double a , double b , double c );
  ```

  How can you get the solutions `double x1,x2` use

  - Direct initialization
  - Uniform initialization
  - Structured binding

  Show your code in a few lines. You don't need to write full codes for these three situations.

Reference

- Implement a function `swap(a , b)` which swaps two integer values.
- Assume that you are implementing a queue and write a function to get the front value of the queue:

```
using T = int;
const int MAX_LEN = 300;
T data[MAX_LEN]; int head , tail;
T& front( ) {
  return data[head];
}
```

What are the advantages for `int&` to be the return type of `front()` compared with `T` or `const T&`?