day-10-mini-project

September 6, 2023

1 Day 10 Mini Project II

1.1 Part I

Write a C++ class `DoubleListD` implementing a doubly-linked list of doubles. Your class should implement the insertion (`<<`) operator to append values to the list, and it should implement a method `int get_length()` which returns the number of items in the list. Implement a `void print()` method to print the contents of the list. Ensure that the destructor properly cleans up the list of values, including any memory the class allocates.

Implement a method `void sort()` which sorts the list in ascending order using the bubblesort algorithm.

Delegate the comparison of two doubles in the list to a private method/external function with the signature `bool greater_than(double lhs, double rhs)` which returns `true` if `lhs` is greater than `rhs` and `false` otherwise.

Your class might look something like this:

```
class DoubleListD {
public:
    DoubleListD() = default;

    ~DoubleListD() {
      // implement the destructor
    };

    int get_length() {
        // return the length
    };

    void sort() {
        // sort the list using bubblesort.
    }

    void operator<<(DoubleListD &list, double d) {
        // append d to list
    }

private:
    struct dlist *mhead;
    struct dlist *mtail;
    int mlength;
};
```

Test your class in a separate file `test-list.cpp` which builds a list from user input and prints it.

1.2 Part II

Extend the `sort` method so that it accepts a parameter `compare` which is a pointer to a function with the signature `bool (double lhs, double rhs)`. This parameter allows the user to inject a comparison to be used during sorting, i.e. this parameter should be used inside `sort` to decide whether to swap adjacent values when sorting the list.

Use this `compare` parameter and function pointers or non-capturing lambdas to sort the list in descending order, or by absolute value, or by difference from some fixed value, or by some other criterion you can think of.

Again, test you class with user input in `test-list-2.cpp`.