Lab Assignment 9

Lab Grading Policy: Attendance 40%, Score 60%

In case you have difficulty in finishing the exercises on time, you should upload them by **Thursday noon** with a penalty of 20% on your score. No late submission is permitted after that. We will in general post the reference solutions **by Friday**.

Exercise 1 (30%, `erase` in a `list`): A prime (質數) is an integer greater than 1 and divisible only by itself and 1. The Greek mathematician Erathosthenes gave a very simple algorithm for finding all prime numbers less than some integer N.

It works like this:

> Begin with a list of integers 2 through N. The number 2 is the first prime. The multiples of 2 (for example 4, 6, 8 etc.) are not primes and should be removed from the list. Then the first number after 2 that has not been removed from the list is the next prime. This number is 3. The multiples of 3 are not primes (for example 9, 15; notice that 6 and 12 have already been removed from the list). The algorithm continues this fashion until we reach N. All the numbers remained in the list are primes.

Write a C++ program that allow the users to supply N. Use a `list<int>` to store integers 2 through N and use its member function `erase` to remove the numbers that are not primes.

A sample run looks like:

```
Please input the number, N: 30
Original data:
2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30
After erasing with the base "2":
2 3 5 7 9 11 13 15 17 19 21 23 25 27 29
After erasing with the base "3":
2 3 5 7 11 13 17 19 23 25 29
After erasing with the base "5":
2 3 5 7 11 13 17 19 23 29
After erasing with the base "7":
2 3 5 7 11 13 17 19 23 29
After erasing with the base "11":
2 3 5 7 11 13 17 19 23 29
After erasing with the base "13":
2 3 5 7 11 13 17 19 23 29
After erasing with the base "17":
2 3 5 7 11 13 17 19 23 29
After erasing with the base "19":
2 3 5 7 11 13 17 19 23 29
After erasing with the base "23":
2 3 5 7 11 13 17 19 23 29
After erasing with the base "29":
2 3 5 7 11 13 17 19 23 29
The primes within 30 are listed in the following:
2 3 5 7 11 13 17 19 23 29
```

Exercise 2 (30%) Write a program to `sort` some Person objects.

```
class Person {
  private:
    string fn;     // first name
    string ln;     // last name
  public:
    Person() {
    }
    Person(const string& f, const string& n)
     : fn(f), ln(n) {
    }
    string firstname() const;
    string lastname() const;
    // ...
};

inline string Person::firstname() const {
    return fn;
}

inline string Person::lastname() const {
    return ln;
}

ostream& operator<< (ostream& s, const Person& p)
{
    s << "[" << p.firstname() << " " << p.lastname() << "]";
    return s;
}
```

Below is a sample run:

```
deque before sort():
[nicolai josuttis]
[ulli josuttis]
[anica josuttis]
[lucas josuttis]
[lucas otto]
[lucas arm]
[anica holle]
deque after sort():
[lucas arm]
[anica holle]
[anica josuttis]
[lucas josuttis]
[nicolai josuttis]
[ulli josuttis]
[lucas otto]
```