

Lab Assignment 8

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%) (a) Write an `IntegerList` class for an array of integers. In the class, use `std::list` as your internal data representation and provide the following constructors:

- `IntegerList()` // print a message said "I am a default constructor."
- `IntegerList(unsigned nelems)` // creates an `IntegerList` with the integers `0...nelems-1`
- `IntegerList(unsigned start, unsigned end)` // creates an `IntegerList` with the range `[start, end)`

In addition, provide a non-member `print` function. Test your program with the client code listed below:

```
#include <iostream>
#include "IntegerList.h"
using namespace std;

int main(){
    IntegerList a;
    IntegerList b(5);
    print(cout, b) << endl;
    IntegerList c(2, 5);
    print(cout, c) << endl;
}
```

```
I am a default constructor.
The elements in the IntegerList are: 0 1 2 3 4
The elements in the IntegerList are: 2 3 4
```

(b) Add an `intersection` member function that prints out the elements common to two arrays. Allow a sequential operation. Test your program with the client code listed below:

```
#include <iostream>
#include "IntegerList.h"
using namespace std;

int main(){
    IntegerList a(3);
```

```
IntegerList b(5);  
IntegerList c(2, 6);  
print(cout, c.intersection(b).intersection(a)) << endl;  
}
```

Your output looks like:

```
I am a default constructor.  
Intersection elements are: 2 3 4  
I am a default constructor.  
Intersection elements are: 2  
The elements in the IntegerList are: 2
```

Exercise 2 (30%): Write a program that allows users to continue input an integer and print them with commas if they have more than three digits. For example, -2036 and 123456789123456 would be printed as -2,036 and 123,456,789,123,456, respectively. Use ! to terminate the input. (**hint:** Method 1: you could use the `insert` member function in STL `std::string`, and manage the input as a string. Method 2 you can use a vector to store the digits, and print the comma when needed. You might need to use the `stoll()` function to convert a string into a `long long` integer.)

```
Enter an integer (! to quit):12345  
The integer with comma is: 12,345  
Enter an integer (! to quit):-12345  
The integer with comma is: -12,345  
Enter an integer (! to quit):123456789123456789  
The integer with comma is: 123,456,789,123,456,789  
Enter an integer (! to quit):-123456789123456789  
The integer with comma is: -123,456,789,123,456,789  
Enter an integer (! to quit):!
```