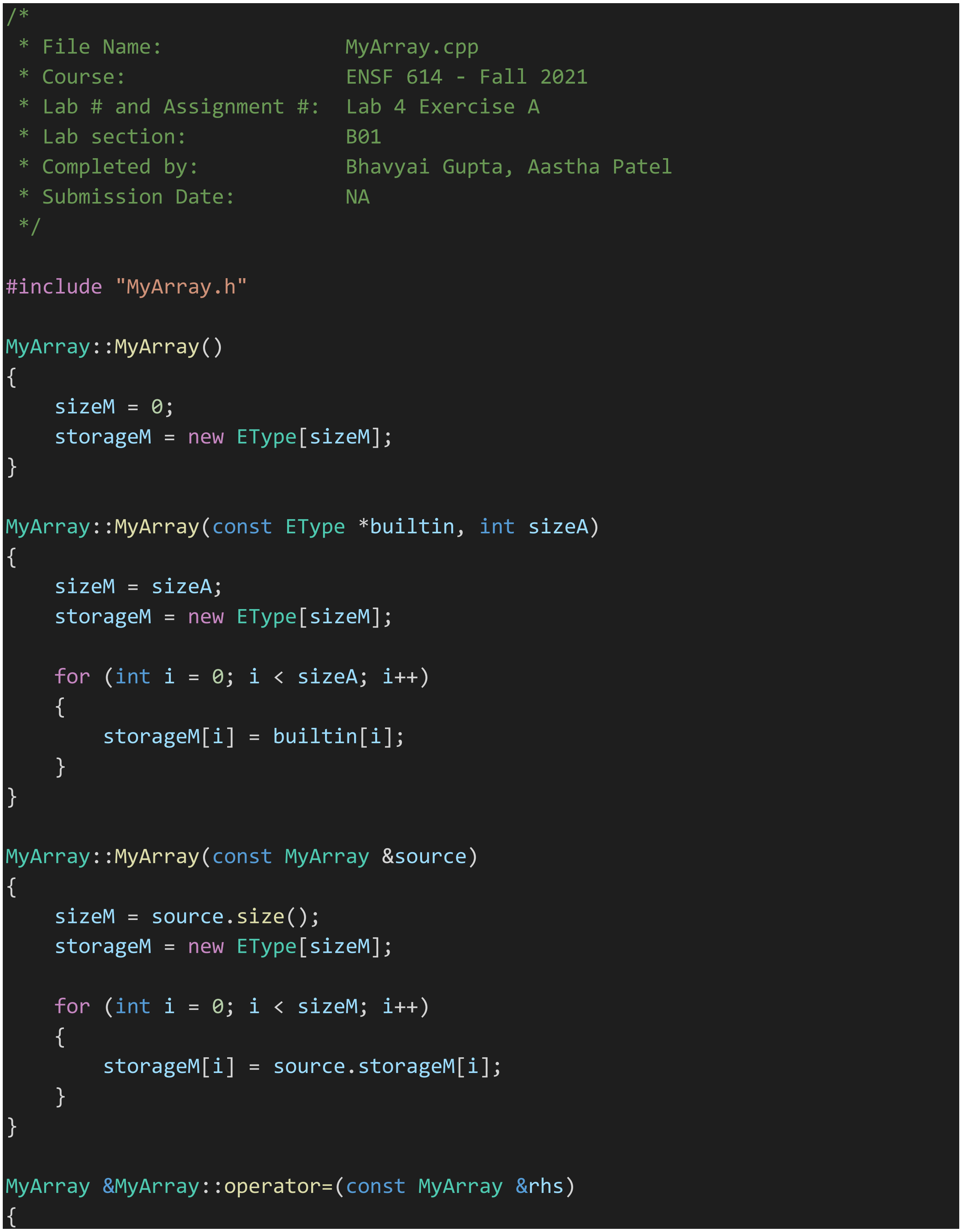
ENSF 614 – Fall 2021

Lab 4 – Tuesday, October 19

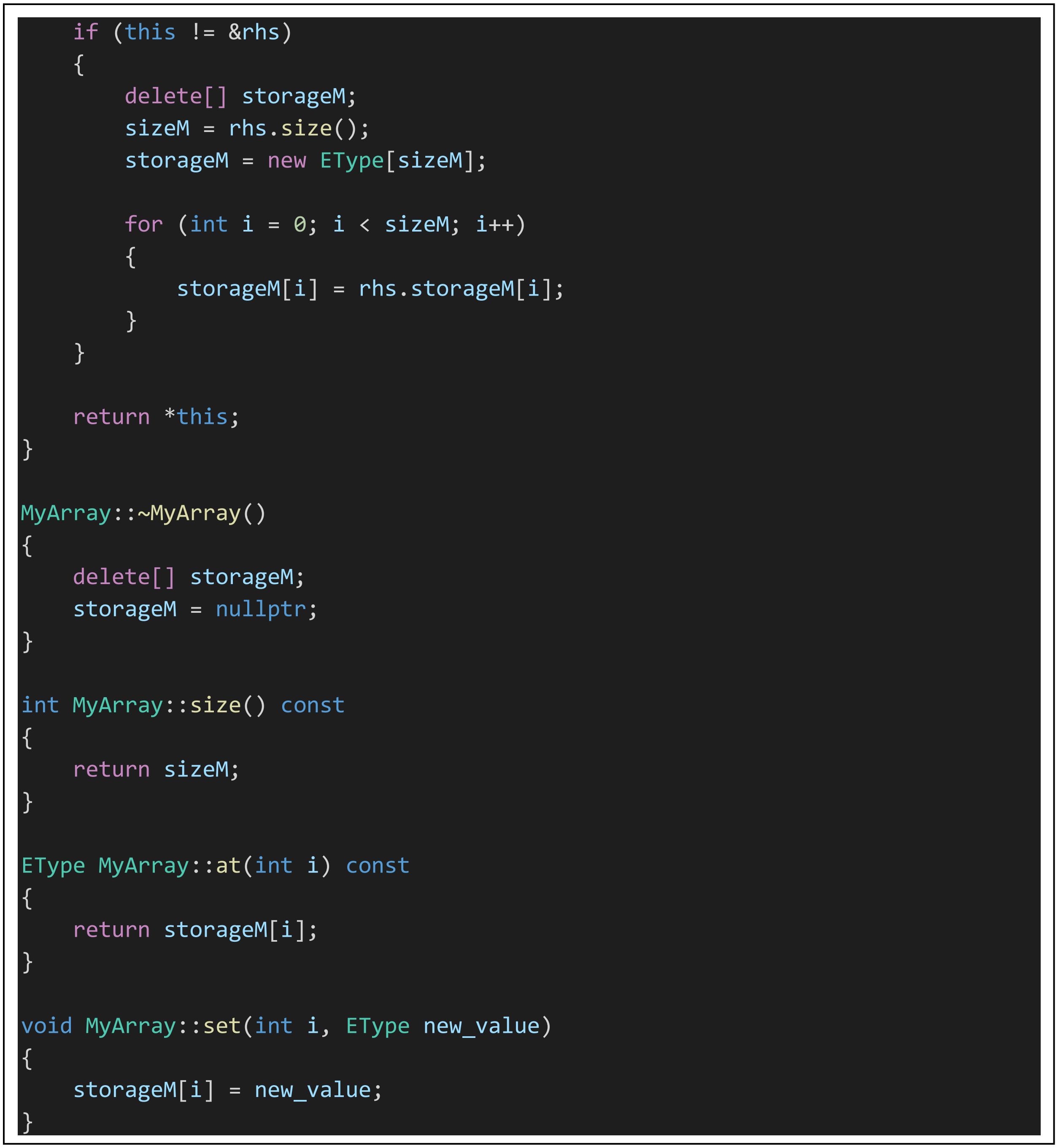
Student Name: Aastha Patel and Bhavyai Gupta

Submission date: October 19, 2021

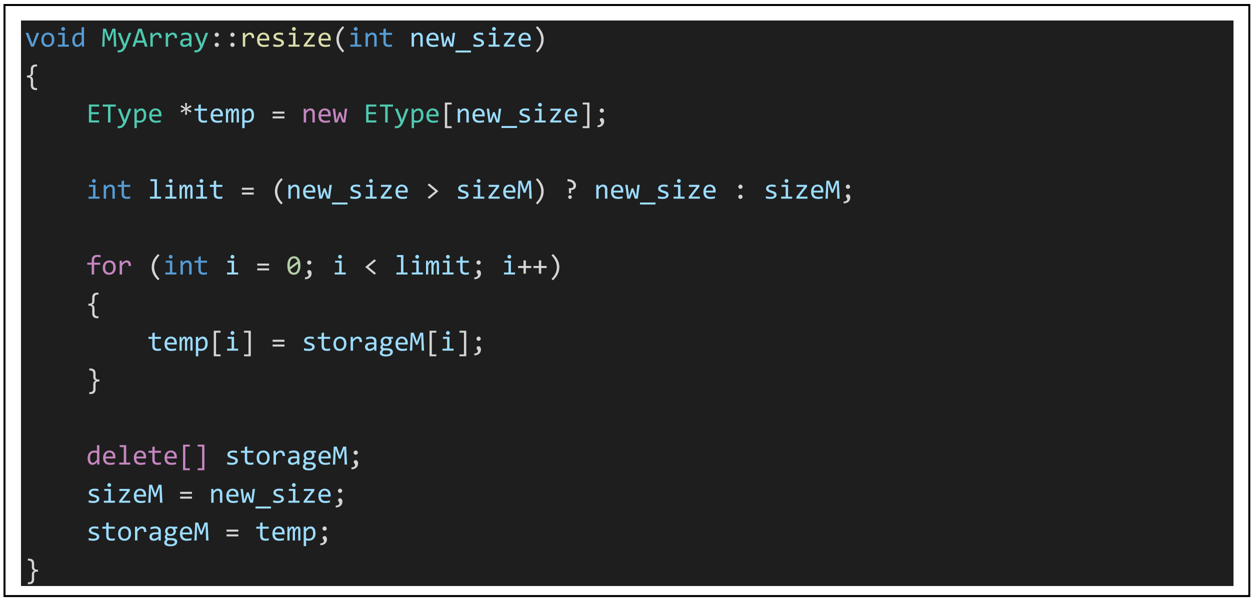
Exercise A – Source file MyArray.cpp



```
/*
 * File Name:                 MyArray.cpp
 * Course:                    ENSF 614 - Fall 2021
 * Lab # and Assignment #:    Lab 4 Exercise A
 * Lab section:               B01
 * Completed by:              Bhavyai Gupta, Aastha Patel
 * Submission Date:           NA
 */

#include "MyArray.h"

MyArray::MyArray()
{
    sizeM = 0;
    storageM = new EType[sizeM];
}

MyArray::MyArray(const EType *builtin, int sizeA)
{
    sizeM = sizeA;
    storageM = new EType[sizeM];

    for (int i = 0; i < sizeA; i++)
    {
        storageM[i] = builtin[i];
    }
}

MyArray::MyArray(const MyArray &source)
{
    sizeM = source.size();
    storageM = new EType[sizeM];

    for (int i = 0; i < sizeM; i++)
    {
        storageM[i] = source.storageM[i];
    }
}

MyArray &MyArray::operator=(const MyArray &rhs)
{
```

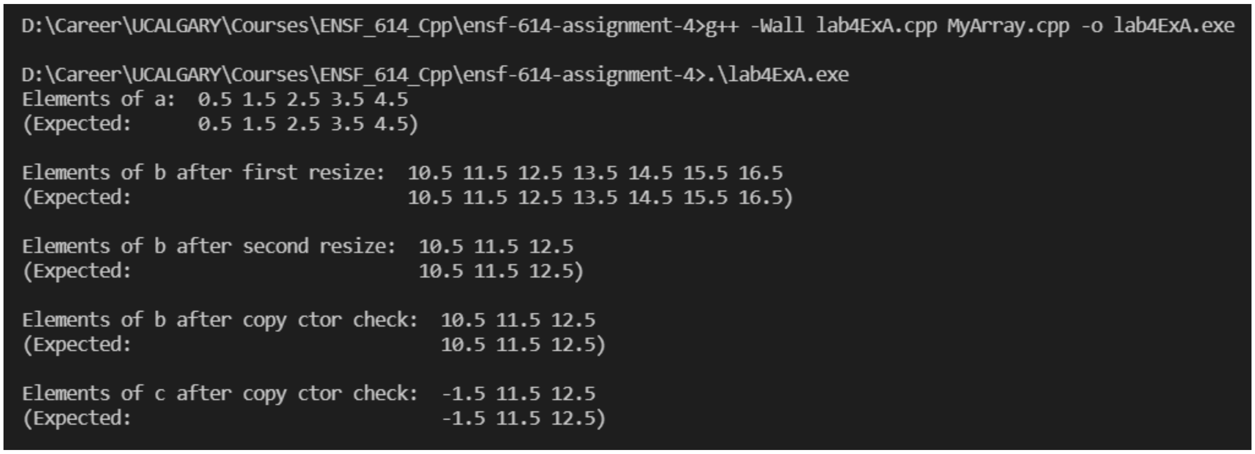


```
    if (this != &rhs)
    {
        delete[] storageM;
        sizeM = rhs.size();
        storageM = new EType[sizeM];

        for (int i = 0; i < sizeM; i++)
        {
            storageM[i] = rhs.storageM[i];
        }
    }

    return *this;
}

MyArray::~MyArray()
{
    delete[] storageM;
    storageM = nullptr;
}

int MyArray::size() const
{
    return sizeM;
}

EType MyArray::at(int i) const
{
    return storageM[i];
}

void MyArray::set(int i, EType new_value)
{
    storageM[i] = new_value;
}
```



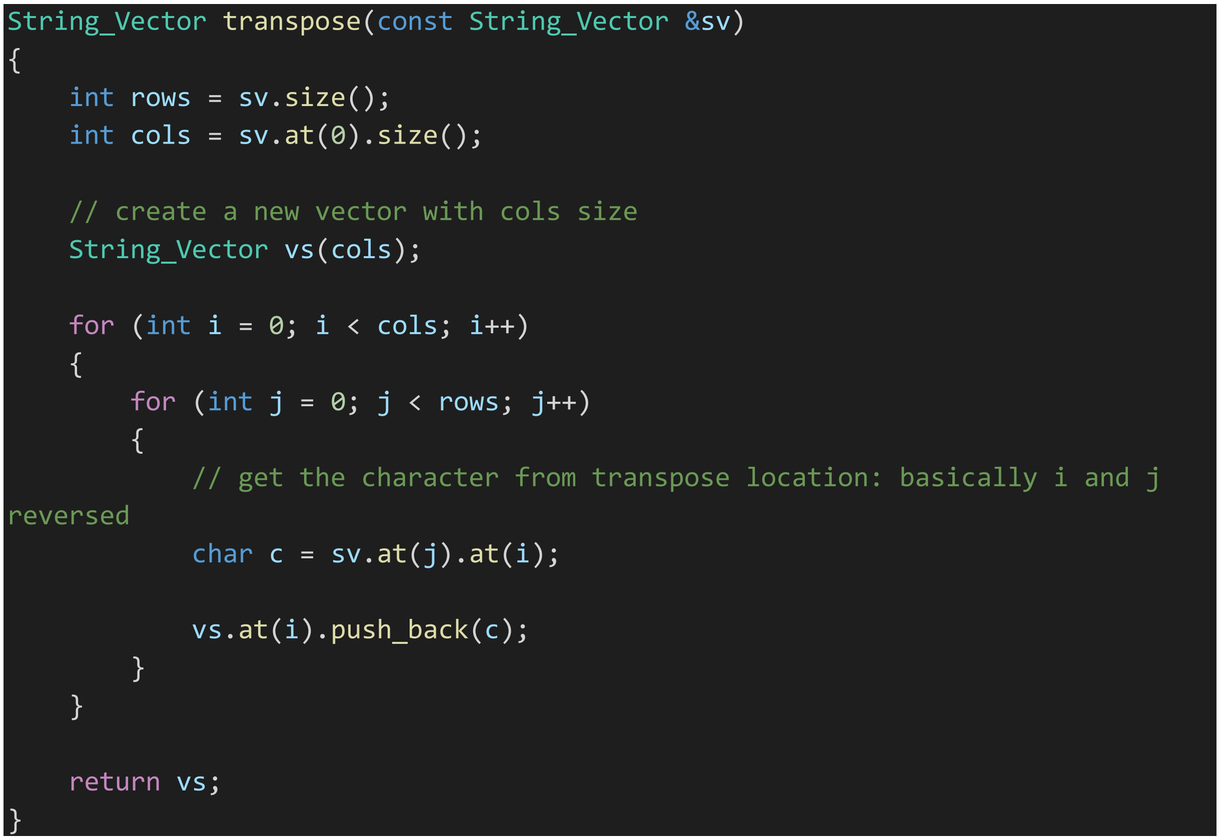
```
void MyArray::resize(int new_size)
{
    EType *temp = new EType[new_size];

    int limit = (new_size > sizeM) ? new_size : sizeM;

    for (int i = 0; i < limit; i++)
    {
        temp[i] = storageM[i];
    }

    delete[] storageM;
    sizeM = new_size;
    storageM = temp;
}
```

Exercise A – Program Output



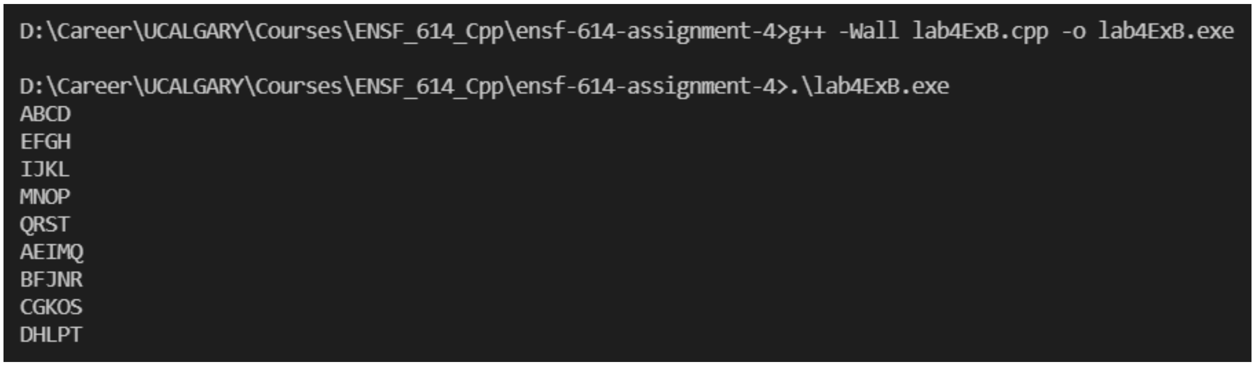
```
D:\Career\UCALGARY\Courses\ENSF_614_Cpp\ensf-614-assignment-4>g++ -Wall lab4ExA.cpp MyArray.cpp -o lab4ExA.exe

D:\Career\UCALGARY\Courses\ENSF_614_Cpp\ensf-614-assignment-4>.\lab4ExA.exe
Elements of a:  0.5 1.5 2.5 3.5 4.5
(Expected:      0.5 1.5 2.5 3.5 4.5)

Elements of b after first resize:  10.5 11.5 12.5 13.5 14.5 15.5 16.5
(Expected:                         10.5 11.5 12.5 13.5 14.5 15.5 16.5)

Elements of b after second resize:  10.5 11.5 12.5
(Expected:                          10.5 11.5 12.5)

Elements of b after copy ctor check:  10.5 11.5 12.5
(Expected:                            10.5 11.5 12.5)

Elements of c after copy ctor check:  -1.5 11.5 12.5
(Expected:                            -1.5 11.5 12.5)
```

Exercise B – Source Code of transpose



```
String_Vector transpose(const String_Vector &sv)
{
    int rows = sv.size();
    int cols = sv.at(0).size();

    // create a new vector with cols size
    String_Vector vs(cols);

    for (int i = 0; i < cols; i++)
    {
        for (int j = 0; j < rows; j++)
        {
            // get the character from transpose location: basically i and j
reversed
            char c = sv.at(j).at(i);

            vs.at(i).push_back(c);
        }
    }

    return vs;
}
```

Exercise B – Program Output

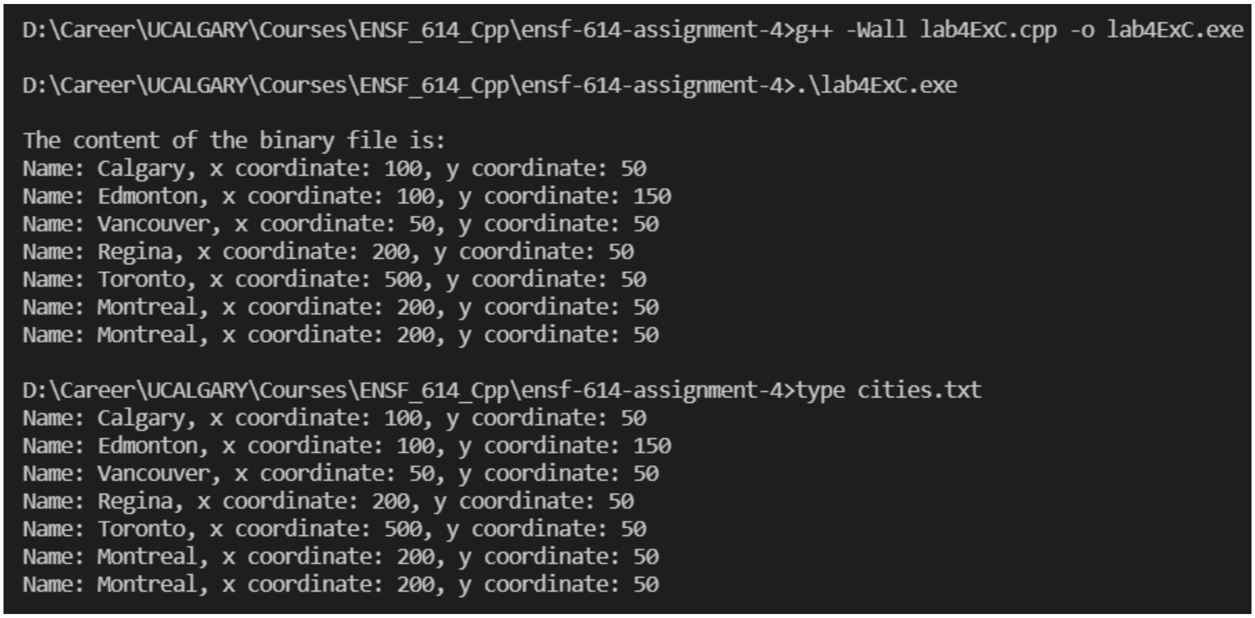


```
D:\Career\UCALGARY\Courses\ENSF_614_Cpp\ensf-614-assignment-4>g++ -Wall lab4ExB.cpp -o lab4ExB.exe

D:\Career\UCALGARY\Courses\ENSF_614_Cpp\ensf-614-assignment-4>.\lab4ExB.exe
ABCD
EFGH
IJKL
MNOP
QRST
AEIMQ
BFJNR
CGKOS
DHLPT
```

Exercise C – Source Code of print_from_binary

```
void print_from_binary(char *filename)
{
    ifstream in_stream(filename, ios::in | ios::binary);

    if (in_stream.fail()) {
        cerr << "failed to open file: " << filename << endl;
        exit(1);
    }

    int length = strlen(filename);
    char *filename_txt = new char[length + 1];

    for (int i = 0; i < length - 3; i++) {
        filename_txt[i] = filename[i];
    }
    filename_txt[length - 3] = 't';
    filename_txt[length - 2] = 'x';
    filename_txt[length - 1] = 't';
    filename_txt[length] = '\0';

    ofstream out_stream(filename_txt);

    if(out_stream.fail()) {
        cerr << "failed to open file: " << filename_txt << endl;
        exit(1);
    }

    City c;

    while (!in_stream.eof()) {
        in_stream.read((char *)&c, sizeof(City));
        cout << "Name: " << c.name << ", x coordinate: " << c.x << ", y
coordinate: " << c.y << endl;
        out_stream << "Name: " << c.name << ", x coordinate: " << c.x << ", y
coordinate: " << c.y << endl;
    }

    in_stream.close();
    out_stream.close();
    delete[] filename_txt;
}
```

Exercise C – Program Output



```
D:\Career\UCALGARY\Courses\ENSF_614_Cpp\ensf-614-assignment-4>g++ -Wall lab4ExC.cpp -o lab4ExC.exe

D:\Career\UCALGARY\Courses\ENSF_614_Cpp\ensf-614-assignment-4>.\lab4ExC.exe

The content of the binary file is:
Name: Calgary, x coordinate: 100, y coordinate: 50
Name: Edmonton, x coordinate: 100, y coordinate: 150
Name: Vancouver, x coordinate: 50, y coordinate: 50
Name: Regina, x coordinate: 200, y coordinate: 50
Name: Toronto, x coordinate: 500, y coordinate: 50
Name: Montreal, x coordinate: 200, y coordinate: 50
Name: Montreal, x coordinate: 200, y coordinate: 50

D:\Career\UCALGARY\Courses\ENSF_614_Cpp\ensf-614-assignment-4>type cities.txt
Name: Calgary, x coordinate: 100, y coordinate: 50
Name: Edmonton, x coordinate: 100, y coordinate: 150
Name: Vancouver, x coordinate: 50, y coordinate: 50
Name: Regina, x coordinate: 200, y coordinate: 50
Name: Toronto, x coordinate: 500, y coordinate: 50
Name: Montreal, x coordinate: 200, y coordinate: 50
Name: Montreal, x coordinate: 200, y coordinate: 50
```