Course: ENSF 614–Fall2021

Lab #: Lab 4

Student Names: Graydon Hall, Jared Kraus

Submission Date: 2021-10-19

Exercise A

Our code output was this:

```
PS C:\Users\grayd\OneDrive\Documents\School\MEng\Semester 3\ENSF 614\Labs\Lab 4\GH> g++ .\MyArray.cpp, .\lab4ExA.cpp
PS C:\Users\grayd\OneDrive\Documents\School\MEng\Semester 3\ENSF 614\Labs\Lab 4\GH> ./a.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

Function definition

```cpp
/* File Name: Lab4ExB.cpp
* Lab # and Assignment #: Lab #4 Exercise B
* Lab section: 1
* Completed by: Graydon Hall and Jared Kraus
* Submission Date: 2021-10-19
*/
String_Vector transpose (const String_Vector& sv) {

    int sv_rows = sv.size();
    int sv_cols = sv.at(0).size();

    int vs_rows = sv_cols;
    int vs_cols = sv_rows;

    String_Vector vs;
    vs.resize(vs_rows);

    for(int i=0; i<vs_rows; i++){
        for(int j=0; j<vs_cols; j++){
            vs.at(i).push_back(sv.at(j).at(i));
        }
    }

    return vs;

}
```

Program Output

```
ABCD
EFGH
IJKL
MNOP
QRST

Transposed vector:
AEIMQ
BFJNR
CGKOS
DHLPT
```

Exercise C

Function definition

```
/* File Name: Lab4ExC.cpp
* Lab # and Assignment #: Lab #4 Exercise C
* Lab section: 1
* Completed by: Graydon Hall and Jared Kraus
* Submission Date: 2021-10-19
*/

void print_from_binary(char* filename) {


    // open input file stream we get cities from
    ifstream is(filename, ios::binary);
    if(is.fail()){
        cerr << "failed to open file: " << filename << endl;
        exit(1);
    }

    // output file we will write to
    ofstream ofs("output.txt", std::ofstream::trunc);
    if(ofs.fail()){
        cerr << "failed to open file: " << filename << endl;
        exit(1);
    }

    // read cities from input file
    City* cityHolder;

    // find nummber of cities, help from https://www.cppstories.com/2019/01/filesize/
    int begin = is.tellg();  // beginning value
    is.seekg (0, ios::end);  // go to end of file
    int end = is.tellg();  // find end value
    int num_cities = (end-begin)/sizeof(City);  // total number of cities
    is.seekg (0, ios::beg);  // go to beginning of stream again

    // read cities from file into array
    cityHolder = new City[num_cities];
    for(int i=0; i<num_cities; i++){
        is.read((char*)(&cityHolder[i]), sizeof(City));
    }
```

```
    // write to ouptput file
    for(int i=0; i< num_cities; i++){
        cout << "Name: " << cityHolder[i].name << ", x coordinate: "
        << cityHolder[i].x << ", y coordinate: " << cityHolder[i].y << endl;

        ofs << "Name: " << cityHolder[i].name << ", x coordinate: "
        << cityHolder[i].x << ", y coordinate: " << cityHolder[i].y << endl;
    }

    // close our files
    is.close();
    ofs.close();
}
```

The content of our generated text file (called output.txt)