/\*

\* Project: lab01\_main.cpp (Submission for EECE2160 Lab 01)

\* Authors: Matthew Springer and Matthew Schomacker

\* Date: Created on Jan 17, 2017

\* Purpose: This program initializes an array and allows for dynamic array modification via user input

\*/

\*\* Assignment 1 \*\*

-- CODE--

/\*

\* Function: Grow

\* Input: None

\* Output: None

\* Purpose: Dynamically adjusts the size of the memory allocated for the array

\*/

void Grow(){

int new\_size;

new\_size = size \*2;

double\* new\_v = new double[new\_size];

for (int i = 0; i < count; i++) {

new\_v[i] = v[i];

}

delete [] v;

v = new\_v;

cout << "Vector grown" << endl;

cout << "Previous capacity: " << size << " elements" << endl;

cout << "New capacity: " << new\_size << " elements" << endl;

size = new\_size;

}

\*\* Assignment 2 \*\*

-- CODE --

/\*

\* Function: PrintVector

\* Input: None

\* Output: None

\* Purpose: Print the current array of elements

\*/

void PrintVector(){

if (count > 0) {

cout << "Current array: [";

for (int i = 0; i < count - 1; i++) {

cout << v[i] << ", ";

}

cout << v[count-1] << "]" << endl;

}

else {

cout << "Current array: []" << endl;

}

}

/\*

\* Function: AddElement

\* Input: None

\* Output: None

\* Purpose: Append a new element to the end of the array

\*/

void AddElement(){

if (count == size) {

Grow();

}

float new\_elem;

cout << "Enter the new element: ";

cin >> new\_elem;

v[count] = new\_elem;

count++;

}

-- COMMAND LINE OUTPUT --

user409@localhost:~/lab01$ g++ lab01\_main.cpp -o main

user409@localhost:~/lab01$ ./main

Main menu:

1. Print the array

2. Append element at the end

3. Remove last element

4. Insert one element

5. Exit

Select an option: 2

You selected 'Append element at the end'

Enter the new element: 2.6

Main menu:

1. Print the array

2. Append element at the end

3. Remove last element

4. Insert one element

5. Exit

Select an option: 2

You selected 'Append element at the end'

Enter the new element: 5.7

Main menu:

1. Print the array

2. Append element at the end

3. Remove last element

4. Insert one element

5. Exit

Select an option: 2

You selected 'Append element at the end'

Vector grown

Previous capacity: 2 elements

New capacity: 4 elements

Enter the new element: 7.0

Main menu:

1. Print the array

2. Append element at the end

3. Remove last element

4. Insert one element

5. Exit

Select an option: 1

You selected 'Print the array'

Current array: [2.6, 5.7, 7]

Main menu:

1. Print the array

2. Append element at the end

3. Remove last element

4. Insert one element

5. Exit

Select an option: 5

You selected 'Exit'

user409@localhost:~/lab01$

\*\* Assignment 3 \*\*

-- CODE --

/\*

\* Function: RemoveElement

\* Input: None

\* Output: None

\* Purpose: Remove the last element in the array

\*/

void RemoveElement(){

if (count == 0) {

cout << "Error: there are no elements in the array to remove. Please select another option" << endl;

}

else {

cout << "Deleting element " << v[--count] << " at index " << count << endl;

}

}

-- COMMAND LINE OUTPUT --

user409@localhost:~/lab01$ g++ lab01\_main.cpp -o main

user409@localhost:~/lab01$ ./main

Main menu:

1. Print the array

2. Append element at the end

3. Remove last element

4. Insert one element

5. Exit

Select an option: 2

You selected 'Append element at the end'

Enter the new element: 4.5

Main menu:

1. Print the array

2. Append element at the end

3. Remove last element

4. Insert one element

5. Exit

Select an option: 1

You selected 'Print the array'

Current array: [4.5]

Main menu:

1. Print the array

2. Append element at the end

3. Remove last element

4. Insert one element

5. Exit

Select an option: 3

You selected 'Remove last element'

Deleting element 4.5 at index 0

Main menu:

1. Print the array

2. Append element at the end

3. Remove last element

4. Insert one element

5. Exit

Select an option: 3

You selected 'Remove last element'

Error: there are no elements in the array to remove. Please select another option

Main menu:

1. Print the array

2. Append element at the end

3. Remove last element

4. Insert one element

5. Exit

Select an option: 2

You selected 'Append element at the end'

Enter the new element: 7.2

Main menu:

1. Print the array

2. Append element at the end

3. Remove last element

4. Insert one element

5. Exit

Select an option: 1

You selected 'Print the array'

Current array: [7.2]

Main menu:

1. Print the array

2. Append element at the end

3. Remove last element

4. Insert one element

5. Exit

Select an option: 5

You selected 'Exit'

user409@localhost:~/lab01$

\*\* Assignment 4 \*\*

-- CODE --

/\*

\* Function: InsertElement

\* Input: None

\* Output: None

\* Purpose: Insert an element in the array at a specified index

\*/

void InsertElement(){

Grow();

int insert\_index;

float new\_elem;

cout << "Enter the index for the new element: ";

cin >> insert\_index;

if (AssertIndexBounds(insert\_index)) {

cout << "Enter the new element: ";

cin >> new\_elem;

// shift all elements at or above the given index to the right

for (int i = count-1; i >= insert\_index; i--) {

v[i+1] = v[i];

}

v[insert\_index] = new\_elem;

count++;

cout << "Element " << new\_elem << " inserted at index " << insert\_index << endl;

}

else {

cout << "Error: you have entered an invalid index. Please enter an index between 0 and " << count << endl;

}

}

/\*

\* Function: AssertIndexBounds

\* Input: int index

\* Output: bool isValid

\* Purpose: Returns true if the given index is valid for an insertion into the array

\*/

bool AssertIndexBounds(int index){

bool isValid = false;

if (index >= 0 && index <= count) {

isValid = true;

}

return isValid;

}

-- COMMAND LINE OUTPUT --

user409@localhost:~/lab01$ g++ lab01\_main.cpp -o main

user409@localhost:~/lab01$ ./main

Main menu:

1. Print the array

2. Append element at the end

3. Remove last element

4. Insert one element

5. Exit

Select an option: 2

You selected 'Append element at the end'

Enter the new element: 3.4

Main menu:

1. Print the array

2. Append element at the end

3. Remove last element

4. Insert one element

5. Exit

Select an option: 2

You selected 'Append element at the end'

Enter the new element: 7.8

Main menu:

1. Print the array

2. Append element at the end

3. Remove last element

4. Insert one element

5. Exit

Select an option: 1

You selected 'Print the array'

Current array: [3.4, 7.8]

Main menu:

1. Print the array

2. Append element at the end

3. Remove last element

4. Insert one element

5. Exit

Select an option: 4

You selected 'Insert one element'

Vector grown

Previous capacity: 2 elements

New capacity: 4 elements

Enter the index for the new element: 1

Enter the new element: 5.6

Element 5.6 inserted at index 1

Main menu:

1. Print the array

2. Append element at the end

3. Remove last element

4. Insert one element

5. Exit

Select an option: 1

You selected 'Print the array'

Current array: [3.4, 5.6, 7.8]

Main menu:

1. Print the array

2. Append element at the end

3. Remove last element

4. Insert one element

5. Exit

Select an option: 4

You selected 'Insert one element'

Enter the index for the new element: -1

Error: you have entered an invalid index. Please enter an index between 0 and 3

Main menu:

1. Print the array

2. Append element at the end

3. Remove last element

4. Insert one element

5. Exit

Select an option: 5

You selected 'Exit'

user409@localhost:~/lab01$

\*\* Assignment 5 \*\*

-- CODE --

/\*

\* Function: Shrink

\* Input: None

\* Output: None

\* Purpose: Reallocates the array with 1/2 its current size when the count falls below 1/3 its

\* current capacity. Minimum array size of 2

\*/

void Shrink(){

// if there are no elements and the array is larger than 2, shrink to the default size

if (count == 0 and size > 2) {

v = new double[2];

size = 2;

}

else {

int comparator;

comparator = count \* 3;

// if the capacity is > 3x the current number of elements in the array, shrink to 1/2 capacity

// change from (count < 30% size) to (count < 1/3 size) approved by Prof. Kimani

if (count > 0 and comparator < size) {

int new\_size;

new\_size = size / 2;

double \* new\_v = new double[new\_size];

for (int i = 0; i < count; i++) {

new\_v[i] = v[i];

}

delete [] v;

v = new\_v;

new\_v = NULL;

cout << endl << "Vector shrunk" << endl;

cout << "Previous capacity: " << size << " elements" << endl;

cout << "New capacity: " << new\_size << " elements" << endl << endl;

size = new\_size;

}

}

}

-- COMMAND LINE OUTPUT --

user409@localhost:~/lab01$ g++ lab01\_main.cpp -o main

user409@localhost:~/lab01$ ./main

/\*

\* Array initialization commands removed

\*/

Main menu:

1. Print the array

2. Append element at the end

3. Remove last element

4. Insert one element

5. Exit

Select an option: 1

You selected 'Print the array'

Current array: [1, 2, 3, 4, 5]

Main menu:

1. Print the array

2. Append element at the end

3. Remove last element

4. Insert one element

5. Exit

Select an option: 3

You selected 'Remove last element'

Deleting element 5 at index 4

Main menu:

1. Print the array

2. Append element at the end

3. Remove last element

4. Insert one element

5. Exit

Select an option: 3

You selected 'Remove last element'

Deleting element 4 at index 3

Main menu:

1. Print the array

2. Append element at the end

3. Remove last element

4. Insert one element

5. Exit

Select an option: 3

You selected 'Remove last element'

Deleting element 3 at index 2

Vector shrunk

Previous capacity: 8 elements

New capacity: 4 elements

Main menu:

1. Print the array

2. Append element at the end

3. Remove last element

4. Insert one element

5. Exit

Select an option: 5

You selected 'Exit'

user409@localhost:~/lab01$