#include <iostream>

using namespace std;

class DynamicStringArray

{

private:

string \* dynamicArray;

int SIZE;

public:

DynamicStringArray()

{

SIZE = 0;

dynamicArray = new string[SIZE];

}

DynamicStringArray(const DynamicStringArray &obj)

{

SIZE = obj.getSize();

dynamicArray = new string[SIZE];

for (int i = 0; i < SIZE; i++)

{

dynamicArray[i] = obj.getElement(i);

}

}

int getSize() const

{

return SIZE;

}

string getElement(int x) const

{

return dynamicArray[x];

}

void addEntry(string s)

{

string \* dynamicArray2;

dynamicArray2 = new string[SIZE + 1];

for (int i = 0; i < SIZE; i++)

{

dynamicArray2[i] = dynamicArray[i];

}

dynamicArray2[SIZE] = s;

SIZE++;

delete[] dynamicArray;

dynamicArray = dynamicArray2;

}

bool deleteEntry(string s)

{

bool flag = false;

for (int i = 0; i < SIZE; i++)

{

if (dynamicArray[i] == s) flag = true;

}

if (flag)

{

string \* dynamicArray2;

dynamicArray2 = new string[SIZE - 1];

int x = 0;

for (int i = 0; i < SIZE; i++)

{

if (dynamicArray[i] != s)

{

dynamicArray2[x] = dynamicArray[i];

}

else if (dynamicArray[i] == s)

{

x--;

}

x++;

}

SIZE--;

delete[] dynamicArray;

dynamicArray = dynamicArray2;

return true;

}

else return false;

}

string getEntry(int x)

{

if (x > SIZE || x < 0)

{

cout << "Index out of bounds." << endl;

return "error";

}

else

{

return dynamicArray[x];

}

}

~DynamicStringArray()

{

delete[] dynamicArray;

}

};

int main()

{

DynamicStringArray names;

// List of names

names.addEntry("Frank");

names.addEntry("Wiggum");

names.addEntry("Nahasapeemapetilon");

names.addEntry("Quimby");

names.addEntry("Flanders");

// Output list

cout << "List of names:" << endl;

for (int i = 0; i < names.getSize(); i++)

cout << names.getEntry(i) << endl;

cout << endl;

// Add and remove some names

names.addEntry("Spuckler");

cout << "After adding a name:" << endl;

for (int i = 0; i < names.getSize(); i++)

cout << names.getEntry(i) << endl;

cout << endl;

names.deleteEntry("Nahasapeemapetilon");

cout << "After removing a name:" << endl;

for (int i = 0; i < names.getSize(); i++)

cout << names.getEntry(i) << endl;

cout << endl;

names.deleteEntry("Skinner");

cout << "After removing a name that isn't on the list:" << endl;

for (int i = 0; i < names.getSize(); i++)

cout << names.getEntry(i) << endl;

cout << endl;

names.addEntry("Muntz");

cout << "After adding another name:" << endl;

for (int i = 0; i < names.getSize(); i++)

cout << names.getEntry(i) << endl;

cout << endl;

// Remove all of the names by repeatedly deleting the last one

while (names.getSize() > 0) {

names.deleteEntry(names.getEntry(names.getSize() - 1));

}

cout << "After removing all of the names:" << endl;

for (int i = 0; i < names.getSize(); i++)

cout << names.getEntry(i) << endl;

cout << endl;

names.addEntry("Olivia");

cout << "After adding a name:" << endl;

for (int i = 0; i < names.getSize(); i++)

cout << names.getEntry(i) << endl;

cout << endl;

cout << "Testing copy constructor" << endl;

DynamicStringArray names2(names);

// Remove Olivia from names

names.deleteEntry("Olivia");

cout << "Copied names:" << endl;

for (int i = 0; i < names2.getSize(); i++)

cout << names2.getEntry(i) << endl;

cout << endl;

cout << "Testing assignment" << endl;

DynamicStringArray names3 = names2;

// Remove Olivia from names2

names2.deleteEntry("Olivia");

cout << "Copied names:" << endl;

for (int i = 0; i < names3.getSize(); i++)

cout << names3.getEntry(i) << endl;

cout << endl;

cout << "Enter a character to exit." << endl;

char wait;

cin >> wait;

return 0;

}

