COSC 501

Lab 9

(50 points) Program 1: Pointer

Complete the function ***isPalindrome*** so that it returns true if the string cstr is a palindrome (the same backwards as forwards) and false if it is not. The function uses the cstring library: strlen(cstr) returns the length of the c-string.

```
bool isPalindrome(char* cstr)
{
    char* front = cstr;
    char* back = cstr + strlen(cstr)-1;
    while (front < back)
    {
        // Complete code here
    }
    return true;
}
```

This is a sample main function for quick testing:

```
int main()
{
    char s1[50] = "neveroddoreven";
    char s2[50] = "notapalindrome";
    cout << isPalindrome(s1) << endl; // true
    cout << isPalindrome(s2) << endl; // false
    return 0;
}
```

(50 points) Program 2: Dynamic Array

One problem with dynamic arrays is that once the array is created using the new operator, the size cannot be changed. For example, you might want to add or delete entries from the array as you can with a vector. This project asks you to create functions that use dynamic arrays to emulate the behavior of a vector.

First, write a program that creates a dynamic array of three strings. Store three names of your choice into the dynamic array. Next, complete the following two functions:

```
string* addEntry( string *dynamicArray, int &size, string newEntry);
```

> This function should create a new dynamic array one element larger than `dynamicArray`, copy all elements from `dynamicArray` into the new array, add the new entry onto the end of the new array, increment size, delete `dynamicArray`, and return the new dynamic array.

```
string* deleteEntry(string *dynamicArray, int &size, string entryToDelete);
```

This function should search `dynamicArray` for `entryToDelete`. If not found, the request should be ignored and the unmodified `dynamicArray` returned. If found, create a new dynamic array one element smaller than `dynamicArray`. Copy all elements except `entryToDelete` into the new array, delete `dynamicArray`, decrement size, and return the new dynamic array.

Test your functions by adding and deleting several names to the array while outputting the contents of the array. You will have to assign the array returned by `addEntry` or `deleteEntry` back to the dynamic array variable in your `main` function.

Sample Output: Red colored texts are user inputs. Other texts are the output of the program.

```
List of names:
Frank
Wiggum
Quimby

1. Add a new name
2. Remove a name
3. Quit
1

Please enter a new name to be added.
YourName

List of names:
Frank
Wiggum
Quimby
YourName

1. Add a new name
2. Remove a name
3. Quit
2

Please enter a name to be deleted.
Frank

List of names:
Wiggum
Quimby
YourName

1. Add a new name
2. Remove a name
3. Quit
3

Quit the program.
```