ECE 71 – Engineering Computations in C

Professor Kriehn – Fall 2017

Code Due By: Midnight on Friday, Nov 17, 2017
Writeup Due By: Class on Mon/Tue, Nov 20/21, 2017

HOMEWORK #34 – Palindromes

Write a program that reads a message, and then tests the line for a palindrome.

Specifications:
You may assume the message contains 100 or fewer characters. Therefore, define a global constant `N` equal to 101, since the last element of a character array must be a ‘`\0`’ (the NULL character). Use `N` to declare the length of your C-string character array, and a `while()` loop with the `getchar()` function to read in characters into the array.

Use a function called `isPalindrome()` that has the following function prototype:

```
bool isPalindrome(char *cstr);
```

The function returns `true` if the characters stored in the C-string for a palindrome, and `false` if they do not.

You must define two pointers to help you code your solution. The first pointer must initially point the 0th character in the character array, and the second pointer should point to the last character in the line. Then use (move) the pointers to point toward different elements within the string and check to see if you have a palindrome. Ignore non-alpha characters. You may use the `isalpha()` and `toupper()` functions to help you. For instance:

```
bool isPalindrome(char *cstr)
{
     char *front = cstr;
     char *back = cstr + strlen(cstr) - 1;

     while (front<back)
     {
          // Complete code here
     }

     return (true);
}
```

If you execute the program, the following information should be displayed:

```
~> main.o
Enter a message: He lived as a devil, eh?
The line is a palindrome.
~>
```

```
~> main.o
Enter a message: Madam, I am Adam.
The line is not a palindrome.
~>
```

HOMEWORK #35 – Dynamic Arrays as "Vectors"

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 homework problem asks you to create functions that use dynamic arrays to emulate the behavior of a vector.

Specifications:
Use the following function prototypes:

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

This function should create a new `dynamicArray` one element larger then `dynamicArray`, copy all elements from `dynamicArray` into the new array, add the new entry into the end of the new array, increment the `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.

```
void output(string *dynamicArray, int size);
```

This function should print each string stored in `dynamicArray` to the screen. Print a return character after each string entry is printed.

Test your functions by creating an initial variable length array with 5 string elements. Store the following strings in each of the elements of the array:

"Gregory", "Bobby", "Helen", "Raylan", and "Ava"

Then perform the following actions:

1. Print the array
2. Add "Kristin" to the array
3. Print the array
4. Delete "Greg" from the array

5. Print the array
6. Delete “Gregory” from the array
7. Print the array
8. Delete “Kristin” from the array
9. Print the array
10. Delete “Helen” from the array
11. Print the array.

If you execute the program, the following information should be displayed:

```
~> main.o
Output 1:
Gregory
Bobby
Helen
Raylan
Ava

Output 2:
Gregory
Bobby
Helen
Raylan
Ava
Kristin

Output 3:
Gregory
Bobby
Helen
Raylan
Ava
Kristin

Output 4:
Bobby
Helen
Raylan
Ava
Kristin

Output 5:
Bobby
Helen
Raylan
Ava

Output 6:
Bobby
```

Raylan
Ava
~>