Project 2: Palindrome

Due: Tue, Feb 21

In this assignment you will implement a `palindrome` class. This class will detect if a given string is a palindrome or not (it reads the same in reverse).
You may NOT use the stack or queue class from the standard library. You may Not use a for-loop in this project.

Getting started

Clone the project stub into the `Projects` subdirectory of SVN working directory.

```
cd ~/ID/Projects
svn export https://dev.cs.uakron.edu/svn/cs316sp17/shared/Projects/Palindrome/
```

Be sure to `svn add` the newly exported `Palindrome` directory and then commit.

Investigate the contents of that directory. You should have the following files:

- `CMakeLists.txt` — The build system for your project. Read this file carefully.
- `palindrome.cpp` — This is where you write your main function.
- `stack.cpp` — This is where you provide the implementation of Stack.
- `stack.hpp` — This is where you provide the definition of Stack.
- `queue.cpp` — This is where you provide the implementation of Queue.
- `queue.hpp` — This is where you provide the definition of Queue.
- `input1.txt` — The text file given for testing.
- `output1.txt` — The expected output from using the `input1.txt` file.

Class requirements

Your task will be to determine if an input string is has the same letters when read forwards as it does backwards. For example: The word 'anna' has the same letters regardless of which end you start from. A palindrome For the purpose of this project we will ignore any non-alphanumeric characters (space, punctuation, non-printable characters).

NOTE: For this program, the case of each character is to be ignored, i.e. Anna would return -1

General Program Flow

```
while a string is entered on standard input and is not "q"
  store the string
  using recursion, add each alphanumeric character to both a stack and a queue
  using recursion, pop a character from the stack and dequeue a character from the queue
  if the characters are not equal
    output the index of the missmatch followed by std::endl
    **recursively empty the stack and queue
  if both the stack and queue are empty
      output -1 followed by std::endl
```

Testing

Create a build directory in the root directory of the project.

DO NOT ADD THE BUILD DIRECTORY TO SVN!

From this directory you should run the following code:

```
cmake ..
```

From this point you can type `make` inside the build directory to compile your code and

```
./palindrome < ../input1.txt
```

to test the output of your code (once it compiles). This command uses indirection to simulate keyboard input using a text file. The output of your program should match the `output1.txt` file in the program directory. An easy way to check this is to run the following commands from the build directory.

```
make
./palindrome < ../input1.txt > ../your_output.txt
diff ../output1.txt ../your_output.txt
```

Submission

Homework is submitted in two ways:

1. Committing it to your SVN repository.
2. Submitting a printout on the day the project is due.

To generate the printout, simply type `make printout` in your build directory. This will generate a code listing in `printout.pdf` with all of your hpp and cpp files files. Open the PDF (using Chrome, Firefox, or a PDF viewer) and print it (two-sided if possible). This is easily done from a lab computer. **Staple multiple sheets together.**

DO NOT FORGET TO BRING YOUR PRINTOUT TO CLASS.

Grading basis

If your homework is not in subversion OR you did not submit a printout, you will get a 0 on your assignment. You must submit **both** to receive a grade.

The total is out of 100 points.

- 85 You submitted code that compiles, and gives the correct output for the given input file but not additional input files.
- 100 You submitted code that compiles, and gives the correct output for the given input file as well as additional input files.
- You lose 10 points for having memory leaks. You can use valgrind to check for memory leaks on the knuth2 server.
- You lose 10 points for not using recursion where required.
- You will lose 10 points for every for loop used.
- You lose 30 points if you do not use your own stack and queue classes.