

Homework 4: Linked Lists, Pig Latin

cs244

Due: October 9, 8:00 AM

Allowances, if any, for late work and related penalties will be as described in the syllabus. As always, check with the instructor for any amendments or changes.

Overview

Implement a program to convert a user entered word into pig Latin form.

Topics: Linked lists, C++ classes, inheritance, STL strings, user I/O, switch statements, conditional branching, code testing, peer reviews

Preparation

Step 1: Create a Folder

In Linux in your .../Documents/Programs folder

Create a folder named **A04**

This is necessary as you will compress the A04 folder and its contents for submission.

Step 2: Check D2L

There may be starter code for this assignment available on D2L.

Files to Create

The files for this assignment MUST be named:

linkedList.h	unorderedLinkedList.h	pigLatin.h
pigLatin.cpp	pigLatinMain.cpp	README.txt

Your *main* function is to be located in pigLatinMain.cpp.

All code files should have comments with the file name, your name, the last modified date, and a description of the file contents. Each function should be appropriately commented in the header and implementation files – most should include pre and post conditions. The body of each function may need comments for the less than obvious parts of the code (if any).

The final submission of files should include a file named: **README.txt**

It provides instructions on how to compile/link your program (EX: `g++ file1.cpp file2.cpp`)

It also provides instructions on how to run your program (EX: after compile type: `./a.out`)

Homework 4: Linked Lists, Pig Latin

cs244

Assignment Description

This assignment is derived from content found in Data Structures Using C++ 2nd Edition by D.S. Malik. Understanding the contents of chapter 5 of that book as well as an examination of the code contained therein is necessary to complete this assignment.

Step 1:

Add the below described rotate function to the class `LinkedListType` as described in chapter 5 of your book.

```
void rotate();  
// Function to remove the first node of a linked list and put it  
// at the end of the linked list.
```

Step 2:

Write a `pigLatin` class that has two public member functions.

The first function should be named **askAndConvert**, taking no parameters and have a return type of `void`. It should prompt the user to input a string and then output the string in the pig Latin form. The rules for converting a string into pig Latin form are as follows:

- a) If the string begins with a vowel, add the string "-way" at the end of the string. For example, the pig Latin form of the string "eye" is "eye-way".
- b) If the string does not begin with a vowel, first add "-" at the end of the string. Then rotate the string one character at a time; that is, move the first character of the string to the end of the string until the first character of the string becomes a vowel. Then add the string "ay" at the end. For example, the pig Latin form of the string "There" is "ere-Thay".
- c) Strings such as "by" contain no vowels. In cases like this, the letter y can be considered a vowel. So, for this program the vowels are a, e, i, o, u, y, A, E, I, O, U, and Y. Therefore, the pig Latin form of "by" is "y-bay".
- d) Strings such as "1234" contain no vowels. The pig Latin form of the string "1234" is "1234-way". That is, the pig Latin form of a string that has no vowels in it is the string followed by the string "-way".

The second function should be named **convertAndPrint**, having one parameter of type `std::string` and returning type `void`. This function should convert the sent string into pig Latin form and output the result to the console using `std::cout`

Your `pigLatin` class member function(s) must store the characters of a string into an unordered linked list and use the function `rotate`, as described in step 1, to rotate the string.

Write a main function in the file `pigLatinMain.cpp` to test your code.

Homework 4: Linked Lists, Pig Latin

cs244

Hints

At some point while coding this...

You should find it necessary to use variables such as:

```
unorderedLinkedList<char> list;
```

and

```
linkedListIterator<char> it;
```

You may need additional private helper member functions in your class. Some to consider might be like:

```
bool pigLatin::isVowel(char ch)
```

```
void pigLatin::convertListString(unorderedLinkedList<char>& pStr)
```

Tester File

Design your main function, in the pigLatinMain.cpp file, to demonstrate and test the capabilities of your pigLatin class and *rotate* function added to linkedListType.

```
// add code and comments as needed
#include <iostream>
#include <string>
#include "pigLatin.h"
using namespace std;

int main()
{
    pigLatin pg;
    pg.askAndConvert();

    string inStr = "";
    cout << "Enter a word (quit to end): ";
    cin >> inStr;
    cout << endl;
    while (inStr != "quit")
    {
        // some code skipped here - add your own
        pg.convertAndPrint(inStr);
        // some code skipped here - add your own
    }
    // some code skipped here - add your own
}
```

Example Tester File, pigLatinMain.cpp --- incomplete

makefile

N/A – this assignment does not *require* a makefile but you can provide one or *for this assignment* you may apply the following

To Compile at the Command Line Type:

```
g++ pigLatin.cpp pigLatinMain.cpp
```

To Run the executable created at the Command Line Type: `./a.out`

Homework 4: Linked Lists, Pig Latin

cs244

Grading

100 points possible

Assuming it is turned in on time:

Your program will be peer reviewed and evaluated by 2 members of your class.

Each will assign a grade to it from 0 to 100.

The instructor will also evaluate your work, scoring it from 0 to 100.

The final grade will then be assigned by the instructor based on the three scores.

Grading will be based on compilability, runability, correctness, comments, programming style, and similar attributes. Further details may be given in class.

Turn-In Directions

Correctly submitting your work is worth 0 points, BUT if not done correctly will likely result in nothing to grade.

Preparation

In Ubuntu Linux browse to your A04 folder

Make sure your source code files are in the folder

**Right click on the A04 folder,
Select Compress...**

Set the file name to be *A04_yourlastname*.tar.gz
where *yourlastname* is your last name

Example: if your last name is Gollygee
then the filename would be
A04_Gollygee.tar.gz

Click on the Create Button

This should create the file named *A04_yourlastname*.tar.gz

Submit

Submit the *A04_yourlastname*.tar.gz file
to the correct course drop box in D2L

Peer Evaluations

For peer review evaluations you should follow the instructions given in class.

In most cases this will involve giving the above compressed files to
the peers assigned to evaluate your program.

Again in most cases, they will then review and evaluate your program.

This often means your code will need to function with a test program they created.

This emphasizes the importance of coding to specification and doing so in a readable
and maintainable manner.

