COSC2430: Programming and Data Structures

Print a file in reverse using array-based stack

Introduction

In this homework, you will create a C++ program that uses an **array-based stack** to read an input file and print the words in the text in reverse order. The initial size of the stack should be of size 10. Every time the stack overflows (attempt to store an item in an already full stack), you will have to create a new stack of double the size of the previous one to continue the operation.

Input and Output

The input is a single text file containing a paragraph of unknown length.

Example 1 of input

Once upon a time, there was a little girl who lived in a village near the forest. Whenever she went out, the little girl wore a red riding cloak, so everyone in the village called her Little Red Riding Hood.

The output of your program should be a text file containing all the words in the original file in reverse order. **Only words must enter the stack.** Punctuation and numbers (including words including numbers or special characters) will not be included in the output.

At the end of the output, you will write the number of times the program had to double the stack to adapt to the length of the text.

Example 1 of output

Hood Riding Red Little her called village the in everyone so cloak riding red a wore girl little the out went she Whenever forest the near village a in lived who girl little a was there time a upon Once
2

As you can see, this text includes 40 words, so the stack has to be double twice to fit it (once from 10 to 20, and once from 20 to 40).

Example 2 of input

Disguised as a human, a lethal cyborg assassin known as a Terminator travels from 2029 to 1984 to kill Sarah Connor.

Example 2 of output

Connor Sarah kill to to from travels Terminator a as known assassin cyborg lethal a human a as disguised
1

In this example, you can see how the numbers were excluded from the output. Furthermore, since the numbers never entered the stack, we only had to double its size once.

A word **containing** a special character or a number should not be included in the output. Examples:

- IPhone6
- test#two
- end,beginning
- one-two

However, the program should be able to handle regular punctuation (not include in output but process the word before it). Examples:

- however,
- great!
- fine.

The main C++ program will become the executable to be tested by the TAs. The result should be written on another text file (output file), provided on the command line. The input and output files are specified in the command line, not inside the C++ code.

The general call to the executable (sum_rowcol, in this example) is as follows:

```
reverse "A=<file>;C=<file>"
```

Call example with one input file and another output file.

```
reverse "A=a.txt;C=c.out"
```

Requirements

- **It is NOT allowed to use vector classes or other classes provided in the STL.**
- Your C++ code must be clear, indented and commented.
- Your program will be tested with GNU C++. Therefore, you are encouraged to work on Unix, or at least make sure that your code compiles and runs successfully on the server.

- You can use other C++ compilers, but the TAs cannot provide support or test your programs with other compilers.

Testing for grading:

- Your main cpp program will be automatically compiled. If there is an error in compilation it will not be executed. **Programs that do not compile on the server will receive a score of 0.** You are responsible of the content of your submission folder.
- **The name of your submission folder must be hw3**. The folder must not be nested in another folder.
- Submitted files must be limited to headers and source files (.h and .cpp).
- Your program should not crash, halt unexpectedly, take an unusual amount of time to run (more than 10 seconds) or produce unhandled exceptions.
- Your program will be tested with 10 test cases, going from easy to difficult.

DEADLINE: October 30th, 5:00PM – NO LATE SUBMISSIONS

The due date cannot be changed. No exceptions, unless there are medical or exceptional reasons.

Grading

- A program not submitted by the deadline is zero points.
- A program that compiles but does not work program is worth 10 points. .
- A code with no comments will receive a 5 points penalty.

Plagiarism and cheating: C++ code is individual: it cannot be shared (other than small fragments used in class or in lab examples). Programs will be compared for plagiarism. If the TAs detect that a portion of your C++ code is highly similar to C++ on the Internet or other student the grade will be decreased at least 50%.

UPDATE: about punctuation

Please read the following comments that clarify what you should expect in term of punctuation and how to handle it:

Scenario	Example	Admit in stack	Notes
Words containing only letters	word	word	
Words including numbers	word11	no	
Words including apostrophes	word’s I’ve	word’s I’ve	
Words preceded by regular punctuation	“word	word	Admitted before a word: (“
Words followed by regular punctuation	word.	word	Admitted after a word:) , . ; : ? “ !
Words with special characters	w@rd	no	
Words with punctuation in the wrong place	wo.rd	no	

You can assume that the test cases will not include more than one punctuation character consecutively. Therefore, you will NOT be tested on examples such as:

- (finally!)
- “How are you?”
- Fine…