

CS 280
Spring 2023
Recitation Assignment 3
February 2, 2023

Due Date: Monday, February 6, 2023, 23:59
Total Points: 9

Write a C++ program that identifies and counts words and special names in a textual file. There are two kinds of special names considered, called Type 1 and Type 2 names. Counting is applied on all occurrences of words and specialized names.

A word is defined as a sequence of one or more non-whitespace characters separated by whitespace. The special names of Type 1 and Type 2 are identified by a special character at the beginning of the name. Type 1 names must start by a '\$' character, while Type 2 names must start by '@' character. All Type 1 and Type 2 names should consist of a letter following the special start character, followed by zero or more letters, digits or underscores. For example, \$value, @val9, \$num_234_ten, and @num_45 are valid Type 1 and Type 2 names, but \$9val, and @_num, \$num\$ are not.

The program may accept one or two command line arguments. The format of the command line arguments is specified as follows:

- The first argument must be a file name.
- The second optional argument is for one of the following flags:
 - ☐ "-all": the program displays the total number of words, the number of Type 1 names, and the number of Type 2 names.
 - ☐ "-type1": the program displays the number of Type 1 names only.
 - ☐ "-type2": the program displays the number of Type 2 names only.

If no file name is provided, the program should print on a new line " NO SPECIFIED INPUT FILE NAME.", and exit. If the file cannot be opened, print on a new line " CANNOT OPEN THE FILE", followed by the file name, and exit. The program should read from the file lines until the end of file is found. If the input file is empty, it prints out the message "File is empty." on a new line and then exits. In case the command line does not include any of the optional flags, the program should display the total number of words, and the total of Type 1 and Type 2 names. If an optional flag argument is not recognized, the program should print out the message "UNRECOGNIZED FLAG".

For example, given an input file “infile3” of the following contents:

Line number	File contents
1	# Various operations on scalar (string) variables.
2	\$fred = "Fred here";
3	\$barney = 56;
4	\$sum = 10 + \$barney ;
5	print 'The variable \$fred' . " contains \$fred. \n";
6	print "Sum is \$sum. \n";
End of File →	

The displayed output for executing the program with the “-all” flag would be as shown below:

```
Total number of words: 34
Number of Type 1 names: 4
Number of Type 2 names: 0
```

Hints:

1. Use <cctype> functions such as: isdigit(), isalpha(), isalnum() to check for digit characters, alphabetic characters, or alphanumeric characters, respectively.
2. You can use *getline()* method for reading from the input file.
3. Convert the read line from the file into an input string stream, then read words from the input string stream. See the slides for details.
4. Download the zipped file for the test cases from Canvas. These are the test cases you will be graded against on your submission to Vocareum. Use the test cases to test your implementation. There are 3 input files: infile1, infile2, and infile3. There are 8 test cases, see the designation of each case in the grading table below.
5. If you want to look at the input for one of the test cases, use the linux "cat" command. The cases are in the directory \$LIB/public/RA_Spring2023/RA3. For example, case 6 file is “all.correct”. You can look at “all.correct” file by using the command "cat \$LIB/public/RA_Spring2023/ all.correct ".

Submission Guidelines

1. Please name your file as “RAx_firstinitial_lastname.cpp”. Where, “firstinitial” and “lastname” refer to your first name initial letter and last name, respectively, and “x” refers to the recitation assignment number (e.g., 1, 2, etc). Your program Submission is to

Vocareum environment. Follow the link of Recitation Assignment 3 on Canvas in the Modules or Assignments pages to connect to the current assignment on Vocareum.

- 2. Submissions after the due date are accepted with a fixed penalty of 25% from the student's score. No submission is accepted after Wednesday 11:59 pm, February 8, 2023.**

Grading Table:

Testing Cases	Points
Case 1: No file name is found (nofile.correct)	1.0
Case 2: File cannot be opened	1.0
Case 3: Empty File (empty.correct)	1.0
Case 4: No flags found (noflag.correct)	1.0
Case 5: Invalid flag (invflag.correct)	1.0
Case 6: Counting using all flag (all.correct)	1.0
Case 7: Counting Type 1 names only (type1.correct)	1.0
Case 8: Counting Type 2 names only (type2.correct)	1.0
Compiles Successfully	1.0
Total	9