Building an array of vectors of strings

This lab assumes that you successfully finished all the previous labs from 1 to 7.

Lab details:

The goal of this lab is to build an array that holds vectors of strings. The last lab, you built a vector to holds strings. Now, you construct a composite data structure, which is basically an array of a vector of strings. You have worked in the lecture on a vector that would hold integers and other types.

You need to implement the following function:

```
#ifndef ARRAY_VECTOR_STR_H
#define ARRAY_VECTOR_STR_H

#include <stdio.h>
#include "vector_string.h"


// Precondition: words is a file contains all the words in
// a dictionary up to length 29
// Postcondition: returns an array of pointers to vector of strings
// The function will do the following:
// 1- First, allocate a memory for an array sufficient to
// hold 30 vector_string_ptr
// 2- Loop through each entry in the array to allocate a
// memory for each vector_string_ptr
// 3- Extract all the strings from the file using the
// string_extract method (from the string object)
// 4- Get the length of the string. Based on its length,
// add it to the corresponding vector. For example, if the
// word was hello, the length is 4. Then, you add the word
// hello to the vector_string at index 4 in the array
Vector_String_Ptr * load_array_vector_string(FILE * words);

// Precondition: pArray is a pointer to an array of vector_string_ptr
// Postcondition: destroy the whole array, free the memory and
// free the memory for each single vector_string_ptr inside the
// array, and then finally, set the array pointer to null
void destroy_array_vector_string(Vector_String_Ptr ** pArray);


#endif
```

Inside your main, you will open the “dictionary.txt” file, which contains an unabridged dictionary with over 120,000 words. After words, you will send the file object to the load_array_vector_string function.

TA CHECKPOINT 1. Demonstrate to your TA that you can get your array of vectors of strings to store all of the words in the dictionary in vectors where all the words in that vector are the same length. Have your program print out the size of each one of these vectors (essentially providing a count of how many words of each length that you have).

What you will need to submit:

- string.c file holding the implementation of the above functions
- string.h file holder the functions declaration
- vector_string.h file holding the above header file
- vector_string.c holding the implementation for the functions
- array_vector_string.h and .c holding the implementation and declaration for the above function
- main_driver.c reading from the dictionary.txt file and testing your functions
- A Makefile to use to compiler your code
- A valgrind report showing that you do not have any memory leaks
- Combine all of the above files into a single compressed file. Name the compressed file: Lab8<first-initals><LastName>
- Submit the compressed file using the submit command to your lab TA.