CSE108 – Computer Programming Laboratory

Final Lab.

Date: Monday May 27, 2019

Hand-in: A student with number 20180000001 should hand in a file named 20180000001.c for this lab.

Part 1. [40pts] Write the function that takes a string from the user and splits the string into words. For each word, you are asked to find three specifications; number of vowels, length and type of the word and hold the word with these specifications by using a structure and pass the structures in to a dynamic array. There are three type of words: "head", "regular" and "proper noun". The head is type of first word of any sentence, proper noun is unique names like person names or city names and the regular word is all the word that are not proper noun or head. The word type must be values of an enumerated type. Your function should print the extracted words as following. You can only use memcpy() from string.h library.

Example:

```
Long may Brandon reign!
---------------------------------
Word: Long
Type: head
 # of vowels: 1
Length: 4
---------------------------------
Word: may
Type: regular
 # of vowels: 1
Length: 3
---------------------------------
…
```

void parse_string(char * text)

Part 2. [30pts] You must create a linked list of numbers, where numbers represent order of a person. You must get 10 numbers from the user and insert them to the linked list using **void create (struct node **head)** function. Subsequently, you must get **delete** number from the user until there is only one person left in the list. For these, implement the function **int winner (struct node **head, int delete)**.

Example:

```
Delete number:3        list->1 2 3 4 5 6 7 8 9 10
3 is eliminated        list->1 2 4 5 6 7 8 9 10
6 is eliminated        list->1 2 4 5 7 8 9 10
9 is eliminated        list->1 2 4 5 7 8 10
2 is eliminated        list->1 4 5 7 8 10
7 is eliminated        list->1 4 5 8 10
1 is eliminated        list->4 5 8 10
8 is eliminated        list->4 5 10
5 is eliminated        list->4 10
10 is eliminated       list->4
4 is winner
```

Part 3. [30pts] Implement a stack which contains integer numbers. Write a function that reverses content of a stack recursively. Assume that the stack has at least 3 integer values.

Example:

```
Stack Content: 1 2 3
Reversed Stack Content: 3 2 1
```