Objectives:
- Arrays
- Strings
- Pointers
- Functions

REMINDERS and GRADING CRITERIA:
- Read these sections from lab1... they haven’t changed.

LAB DESCRIPTION

PART 1. GRADE CALCULATOR (45%) Mandatory filename → lab2p1.c

Based on the grading scheme as assigned in your syllabus, determine the final grade percent you would earn. However, the program should still work if you were to enter the grading scheme for any other class as well.

You already know how to do user/data entry so use #define statements to identify the constants needed for the following 4 grade categories (homeworks, labs, quizzes, midterms):
- Number of homeworks
- Number of labs
- Number of quizzes
- Number of midterms
- Total percent of homework
- Total percent of labs
- Total percent of quizzes
- Total percent of midterms
- Percent of final exam
- Final exam score

You should have 4 arrays – one for each category plus a variable for the final exam score; or if you are feeling froggy, use can use one two dimensional array.

Since you already know how to do user/data entry and based on the information above, initialize the appropriate arrays with the correct number of scores; and don’t forget to assign a final exam score to the final exam variable. These scores should all be in percent value i.e. if you get an 8/8 on a quiz, then your score is 100%.

Be sure to use two functions in this assignment (your choice) and completely document what each function is doing, describing the necessary arguments/parameters being passed and how they are being passed. Be sure to use at least one pass by reference parameter in this assignment.

ASSUMPTIONS:
- The percent of each assignment in a given category will all be the same. For instance, if you have 5 homeworks and the total percent of homework is 20% of the total grade, then each homework is worth 4%.
• Be sure to account for the possibility that the number of scores in any particular category is zero; which
would make the total percent of that category also zero.
• The #define constants and array score initializations should be able to change and your program still work.

PART 2. SIMPLE SIPHER (45%) Mandatory filename → lab2p2.c

You have a message that you want to encode. The scheme to encode is for each “word” in the phrase, put the first
letter at the end of the word, moving remaining part of the word over (i.e. to the left one position) so the output
phrase doesn’t have additional blanks or different size string than the input.

You will need to use the underscore character ‘_’ instead of a blank for the input. However, when you create your
encoded output, be sure to convert the underscore to a blank.

In addition, output all lower case characters i.e. convert any upper case characters from the input to lower case for
the output.

Example Input/Output:
  War_eagle → arw_aglee
  Reading_records_of_variable_length? → eadingr_ecordsr_fo_ariablev_engthl?
  Have_fun._Doing_this_lab_:) → aveh_unf_oingd_hist_abl:_)

ASSUMPTIONS:
• So that you do not have to worry about array bound checking issues, use 100 as the array size and make sure
to tell the user not to enter more than 100 characters at a time (after all, texting has a character limit so why
can’t we?!). We’ll assume the user knows how to follow directions per the previous lab.
• You can assume that reaching a non-alphabetic character and/or a blank is the end of a word.

PART 3. POINTER MADNESS SIMPLIFIED (10%) → Mandatory filename → lab2p3.c

A. Write a loop that uses a pointer reference as the loop counter. In other words, convert the variable x to a pointer
reference in the below code so that it works exactly the same:

```c
int x;
for (x= 1; x < 5; x++)
    printf("loop counter value is %d \n",x);
```

B. What is the value of the loop counter once the for loop has finished executing? Write a printf statement to output
this value using the pointer variable. Write another printf statement to output this value using the variable x.

C. What is the value of the pointer variable; not the value that it points to, but the value of the pointer itself (i.e. the
address)? Write a printf statement to output this value using the pointer variable. Write another printf statement to
output this value using the variable x.

NOTE: All 3 parts (A, B, and C) should be in the same file.

LAB SUBMISSION

Read this section from lab1... it hasn’t changed except for:

The labname for this lab is: lab2
Be sure to submit the following files: lab2p1.c lab2p2.c lab2p3.c