COMP3350
Lab 3

Important Notes:

- There will be **two** submitted **assembly language files (.asm file)**. The **code** must be accompanied by adequate comments. Writing only the code could lead to **zero** credits.
- The submitted **lab report** should be **a single PDF file** unless otherwise mentioned. Texts of the reports are expected to be typed in for which you could use software programs like LaTeX, Microsoft Word etc. All screenshots (if any) should be included in this PDF file, clearly titled, and accompanied with adequate descriptions.
- The PDF file should be clearly formatted to help the TA/instructor to locate the answer to each corresponding question. If the student fails to label the answers with reasonable efforts or submit files with poor readability, the submission may only receive zero or partial score.
- Please use the following naming conventions for your submitted files.
    - Lab3a.asm
    - Lab3b.asm
    - Lab3report.pdf
- Students are encouraged to complete each task with best efforts. Even if the final outcome isn't fully correct, partial credit may be awarded if the instructor and TA recognize that the student demonstrates a certain level of understanding.
- Solutions turned in must be your own. Please, mention references (if any) at the end of each question.

Please Complete the Following Task(s).

- **Task a (10 points) Word Count:** Write a MIPS assembly program in the MARS simulator that performs the following steps:

  The program allocates space for an array of 100 characters. The program then prompts the user to enter a string of size less than 100 characters; it reads the string input by the user (no need to test with invalid inputs). The program then iterates through the string (until the null character is encountered) to perform a word count. A word starts with an alphabet a-z or A-Z and ends when a non-alphabet (not a-z and not A-Z) is encountered. The program ends by printing the string "Word count: " followed by the number of words in the string. For reference, here is an ASCII table. Here is an example run of the program:

  ```
  Enter a string of size less than 100 characters:
  I never realized it would be this insanely fun to program in assembly.
  Word count: 13
  ```

- **Task B (10 points) Calculating a Modified Fibonacci Number in MIPS with Recursion:**

The **Modified Fibonacci Sequence** starts with two user-defined integers (**F0** and **F1**) instead of the traditional 0 and 1. Each subsequent number is the sum of the previous two.

For example:
If **F0 = 3** and **F1 = 5**, the sequence would be:
3, 5, 8, 13, 21, 34, 55, ...

Write a recursive MIPS assembly program in the MARS simulator to print the Nth integer in the modified Fibonacci sequence. Recall that procedures are invoked with the jal instruction and you must perform appropriate saves/restores before/after the procedure invocation. The pseudo code for procedure modFib is shown below:

```
procedure modFib(N)
if (N == 0) return F0
if (N == 1) return F1
return modFib(N-1) + modFib(N-2)
```

Prompt the user for F0, F1, and N (we will only test with integer N greater than or equal to 0). An example run of the program:

```
Enter the first number (F0): 3

Enter the second number (F1): 5

Enter the position (N): 5

The 6-th Modified Fibonacci number is: 34
```