Program 2: Finding
Program due: Wednesday, February 27th, 8 p.m.

Given two arrays of integers: **numbers** and **findNums**, write a MIPS program that for each value in **findNums** will locate places in **numbers** where that integer appears.

For example, given the following:

```
.data
numbers:
          .word   9
          .word -17
          .word  14
          .word   9
          .word -23
          .word   8
          .word -23
          .word   9
          .word  14
          .word -17
          .word   9
numNumbers:
          .word  11
findNums:
          .word  19
          .word   8
          .word   9
          .word   0
numFindNums:
          .word   4

# Your code goes below this line
```

The program should produce the following output:

```
19 not found
Found 8 at location 5
Found 9 at location 0
Found 9 at location 3
Found 9 at location 7
Found 9 at location 10
0 not found
```

Notes:

- **9** is present in **numbers** four times, and the program will print all four of the locations.
- **19** and **0** are not present in **numbers**, and the program will report this fact.
- **numNumbers** is the number of elements in the **numbers** array.
- **numFindNums** is the number of elements in the **findNums** array.

Input:

We will provide a **.data** portion for your program. In each test case, the label **numNumbers** will state the number of locations in the array **numbers**. The array **numbers** will then contain at least the number of elements indicated by **numNumbers**. We guarantee **numNumbers** will be **>= 0**. The same is true for **numFindNums** and the **findNums** array. There are some special cases:

- It is possible that **numFindNums** will be zero. In this case, your program will have nothing to print. See **test05.s** and **test05.s.out** for an example.
- It is possible that **numNumbers** will be zero. In this case, your program will print a result for each value in **findNums**. Of course, the result will be that the number from **findNums** is not present in **numbers**. See **test04.s** and **test04.s.out** for an example.
- It is possible that both **numNumbers** and **numFindNums** will be zero. In this case, your program will print nothing. See **test10.s** and **test10.s.out** for an example.

We will provide sample files for you to copy. They will also be on the Windows machines and on the Unix machines (lectura and the Fedora machines, fd01 to fd08). On the Windows systems, look in the shared Rotis drive: **Rotis->csc252->shared->prog2**. On the Unix machines, look in the directory: **~cs252/spring08/prog2**. In both cases, the files will be named: **test01.s**, **test02.s**, etc. Secure ftp can also be used to access the test cases on lectura from your home system. Please check with us if you have problems accessing these test cases.

Output:

For each entry in the **findNums** array, you will print one or more lines. If a particular entry in **findNums** is not present in the **numbers** array, then you will print:

```
19 not found
```

If a particular entry in **findNums** is present in the **numbers** array, then you will print:

```
Found 8 at location 5
```

This line will be repeated for each time that the entry appears in the **numbers** array.

Programming points:

You may use the **$s** and **$t** registers as needed. Put appropriate comments in your program to indicate what is happening in the program. Include comments at the beginning (below the required comment) that describe the purpose of the program.

The required comment line:

```
# Your code goes below this line
```

must be present in the **prog2.s** file that you turnin! Everything that you provide (comments and code) must go below this required comment line.

The **prog2.s** that you turn in may contain one of the test cases above the required comment, or the required comment can be the very first line (no test case data and no comments above it). Either is acceptable.

Turnin:

Note: We are asking for submission of programs via D2L only.

- Name your program: **`prog2.s`**
- Using a web browser, go to: http://d2l.arizona.edu/
- Login using the "UA NetID Login" (upper-left corner of the web page).
- You should now be at "My Home" on D2L. At the center bottom of the screen, under the heading "My Academic Courses", you should find "C SC252 SP08 001-002H Homer" listed under 2008 Spring. Click on this link.
- You will now be at the CSc 252 page for Spring 08.
- There is a row of links just underneath the Wildcat. In this row, you will find the link "Dropbox". Click on this link.
- You will be at a page that shows three Dropbox Folders: **`Program2`**, **`Program2-Late`**, **`Program2-Regrade`**. Only one of these will be active. The **`Program2`** folder is for on-time submissions. The **`Program2-Late`** folder will be available for late submissions. The **`Program2-Regrade`** folder is used for requesting a re-grade of the program after the grades have been returned. Click on "Program2". If you are doing a late turnin, click on "Program2-Late".
- You should now be at the page: "Submit Files - Program2". Click on the "Add a File" button. A pop-up window will appear. Click on the "Choose File" button. Use the file browser that will appear to select your **`prog2.s`** file. Click on the "Upload" button in the lower-right corner of the pop-up window.
- You are now back at the "Submit Files - Program2". Click on "Upload" in the lower-right of this window. You should now be at the "File Upload Results" page, and should see the message **`File Submission Successful`**.
- You can repeat this process as often as necessary. Each submission will be time-stamped. When we grade programs, we will grade only the most recent submission.