CSE 341
Fall 2023
Project #2

Due: Friday, October 20, 2023 at 11:59 PM

It is important that you start early and do NOT procrastinate. To encourage this, help will not be available for this project the day it is due. ***START EARLY!***

Task

Write a MIPS assembly language subroutine called ***Hamming*** that will process 7-bit single error detecting, single error correcting Hamming codes. The subroutine should present the user with a menu consisting of the following options.

- Allow the user to enter a 7-bit Hamming code. The code should be stored as a NULL terminated ASCII string in memory at the address passed in the Hamming routine in register `a0`.
- Extract the correct 4-bit data word encoded by the 7-bit Hamming code stored in memory at the passed into the Hamming routine in register `a1` as a NULL terminated ASCII string. This means correcting a single error in the 4-bit data word if there is one while extracting it. The 4-bit data word should be stored in memory as a NULL terminated ASCII string at the address passed into the Hamming routine in register `a1`.
- Determine if there is an error in the 7-bit Hamming code stored in memory as a NULL terminated ASCII string at the address passed into the Hamming routine in register `a0`. If there is an error, the position where the error exists should be reported. If there is not an error, the fact that the code is error free should be reported.
- Allow the user to enter a 4-bit data word. The code should be stored in memory as a NULL terminated ASCII string at the address passed into the Hamming routine in register `a1`.
- Encode the 4-bit data word stored in memory as a NULL terminated ASCII at the address passed into the Hamming routine in register `a1` to a 7-bit Hamming code. The resulting Hamming code should be stored in memory as a NULL terminated ASCII string at the address passed into the Hamming routine in register `a0`.
- Display the 4-bit data word stored in memory as a NULL terminated ASCII string at the address passed into the Hamming routine in register `a0` to the console.
- Display the 7-bit Hamming code stored in memory as a NULL terminated ASCII string at the address passed into the Hamming routine in register `a1` to the console.

A partial (not complete) sample of a menu that might be used is shown below.

```
CSE 341 Hamming Code System
Select:
      1 - Enter a 7-bit Hamming code
      2 - Extract the 4-bit data word encoded by a 7-bit Hamming
      code
      3 - Determine if there is an error in a 7-bit Hamming code

      . . .

      8 - Quit

Enter Choice:
```

The routine should accept the address where the 7-bit Hamming code will be stored in register `a0`, and the address where the 4-bit data word will be stored in register `a1`. Addresses passed in registers `a0` and `a1` will be in the data segment above `0x10000400`. You may use addresses in the range `0x10000000` through `0x10000400` for your needs (such as strings storing prompts). Note that the *read char syscall* (12) does not work in SPIM. You may use the *read string syscall* (8) or read integer syscall (5) to obtain the user's choice. The native multiplication (mult, multu) and division (div, divu) instructions cannot be used in your program. It is strongly suggested that each of the tasks in the list above be implemented by its own subroutine which can be called when the task is selected by the user.

Submission & Grading

Your program must be written using only native MIPS instructions, so be sure to test in SPIM using bare mode. Submit your commented assembly language code using the submit command (*submit_cse341 project_2.s*) on *timberlake.cse.buffalo.edu* ***before 11:59 PM on Friday, October 20, 2023***. **In an effort to encourage students to start the project early, the CSE 341 staff will not answer questions related to this project on the day it is due. Hence it is important to start early.**