Lab 3

In this lab, you are supposed to learn how interrupt-driven Input/Output can interrupt a running program, execute the interrupt service routine, then return to the interrupted program.

In this lab, you will use keyboard as your input device for interrupting the running program.

Your Task

Your program consists of these three parts:

1. The user program

The main program will continuously producing two different pattern based on the keyboard input.

One pattern is called "** checkerboard". You will be alternately printing the following two different lines. The first one consists of the pattern "** followed by **four** space" **eight** times. And the second one consists of **three** spaces and the pattern "** followed by **four** spaces" **seven** times.

```
**    **    **    **    **    **    **    **
   **    **    **    **    **    **    **
**    **    **    **    **    **    **    **
   **    **    **    **    **    **    **
**    **    **    **    **    **    **    **
   **    **    **    **    **    **    **
**    **    **    **    **    **    **    **
   **    **    **    **    **    **    **
...
```

The other pattern is called "## checkerboard". You will be alternately printing the following two lines. The first one consists of the pattern "## followed by **four** space" **eight** times. And the second one consists of **three** spaces and the pattern "## followed by **four** spaces" **seven** times.

```
##    ##    ##    ##    ##    ##    ##    ##
   ##    ##    ##    ##    ##    ##    ##
##    ##    ##    ##    ##    ##    ##    ##
   ##    ##    ##    ##    ##    ##    ##
##    ##    ##    ##    ##    ##    ##    ##
   ##    ##    ##    ##    ##    ##    ##
##    ##    ##    ##    ##    ##    ##    ##
   ##    ##    ##    ##    ##    ##    ##
...
```

Your program should start with "** checkerboard". To ensure the output is not too fast to be seen, the user program should include a piece of code that will count down from 2500 to 0 each time a line is printed on the monitor. In other words, your program will sleep for a short time.

A simple way is to delay, like following:

```
DELAY   ST      R1, SaveR1
        LD      R1, Count
REP     ADD     R1, R1, #-1
        BRp     REP
        LD      R1, SaveR1
        RET
COUNT   .FILL   #2500
SaveR1  .BLKW   #1
```

2. The keyboard interrupt service routine

The keyboard interrupt service routine will simply output to the monitor **ten** times whatever key the person typed, followed by a linefeed. **And change the pattern that will be printed to the monitor after the user pressed the key.**

Important: You may not use any TRAP instructions in your interrupt service routine. To print a character to the screen, you MUST poll the DSR and then write to the DDR. If you use TRAP or if you do not poll the DSR properly before writing to the DDR, your program is not correct even though it may appear to work.

Hint: *Don't forget to save and restore any registers that you use in the interrupt service routine*.

3. The operating system enabling code

To ensure your program work, you have to do some extra preparation.

- Firstly, you should initialize the stack pointer, so that when an interrupt occurs, the PC and the PSR can be pushed.(And after your interrupt service routine is finished, you can use RTI instruction to go back, meanwhile the PC and the PSR will be popped.) So please initialize R6 to x3000, indicating an empty stack.
- Secondly, you should establishes the interrupt vector table to contain the starting addresses of the corresponding interrupt service routines. The starting address of the interrupt vector table is x0100 and the corresponding interrupt vector for the keyboard is x80.
- Thirdly, you should set the IE bit of the KBSR.

Example

```
**    **    **    **    **    **    **    **
   **    **    **    **    **    **    **
**    **    **    **    **    **    **    **
   **    **    **    **    **    **    **
**    **    **    **    **    **    **    **aaaaaaaaaa
   ##    ##    ##    ##    ##    ##    ##
```

```
##    ##    ##    ##    ##    ##    ##    ##
   ##    ##    ##    ##    ##    ##    ## xxxxxxxxxx
**    **    **    **    **    **    **    **
   **    **    **    **    **    **    **
**    **    **    **    **    **    **    **
   **    **    MMMMMMMMMM
##    ##    ##    ##    ##    ##    ##    ##
   ##    ##    ##    ##    ##    ##    ##
...
```

In this example, the key 'a', 'x' and 'M' were pressed.

Requirements & Notes

- Write your program in LC-3 assembly language.
- Your user program should start at location x3000.
- **Note: The Linux/Unix LC-3 simulator may not support interrupts. You'd better use the Windows or Web version for this lab.**
- Since your user program contains an infinite loop, to stop the program, you must press the "Stop Execution" button.

Grading

Lab 3 takes 10 points of the total score, consisting of Check part (60%) and Report part (40%).

Due: July 21, Tuesday

- **Check** (60% of each lab)
    - Find a TA to check your code in person, TAs may ask you questions when grading your lab assignment, you will get 100%, 80% or 60% of the checking score according to your response.
    - You can try again if you fail in checking, but there will be a penalty of -10% (of checking part) for each try.
- **Report** (40% of each lab)
    - English report should be concise and carrying main ideas. Try to use the report to convince TAs that you complete the task by yourself.
    - Your lab report should contains the following contents:
        - **Introduction** of this lab. A brief explaination of what this lab requires your program do.
        - **Algorithm**. Brief explanation of your code. The complexity of your algorithm will not affect your score. You can use graphs to explain your algorithm.
        - Enough **Test Cases**. You can use the test cases in checking part, or make your own cases.
        - **Discussion and Experience**. Write some thinkings about this lab.
        - **Source code** with sufficient comments. Comments begin with semicolon ';'.
- **Penalty**

- Delay: -10% per day. If more than 5 days, -100%.
- Cheating: -100% of this lab. Besides, -10% of the total score.