Lab 1

Write a program in **LC-3 machine language** (use zeros and ones) to detect whether or not a 16-bit value has at least three consecutive '1's.

The following examples satisfy this property:

```
0000 0000 0000 0111
0000 0001 1100 0000
1111 1101 1101 1101
```

The following examples **do not** satisfy this property:

```
0000 0000 0000 0000
0101 0101 0101 0101
1101 1011 0110 1101
```

Requirements

1. Your program should start at x3000 which means the first instruction of your program is located in position x3000.

   > Hint: The first line of each program specify the start address of the program.
   >
   > ```
   > 0011 0000 0000 0000 ; (.ORIG x3000)
   > .....               ; your first instruction of the program
   > ```

2. Your program should load memory location x3100 to get the value to be detected. If the value satisfy this property, store value 1 in R2, otherwise store 0.
3. Halt the machine after executing your program.

   > Hint: Machine code for HALT `TRAP x25` is `1111 0000 0010 0101`.

Grading

Lab 1 takes 5 points of the total score, consisting of Check part (60%) and Report part (40%).

Due: July 13, Monday

- **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:
    - Your algorithm. The complexity of your algorithm will not affect your score.
    - Brief explanation of your code.
    - 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.