

Lab 1: Lucky 111

Professor Patt loves the number 7. As a computer man, he would represent 7 in binary 111. That is his favorite binary pattern, called Lucky 111. Your job is to write a program to judge whether a given 16-bit value contains that pattern (three consecutive 1's).

The following examples satisfy this condition so Professor Patt loves them:

```
0000 0000 0000 0111
0011 1001 1010 1101
1111 1111 1111 1111
```

The following examples do not satisfy this condition:

```
0000 0000 0000 0000
0110 1101 1011 0110
1010 1010 1010 1010
```

Implementation Details

- You are required to write in **LC-3 machine codes** (0's and 1's).
- Your program should start at x3000, which means the first instruction of your program is located in position x3000.

```
0011 0000 0000 0000 ; (.ORIG x3000)
..... ; your first instruction of the program
```

- The input 16-bit value is located in memory location x3100. Your program should load the value and then examine it.
- If the input value satisfies, then set R2 to 1. Otherwise, set R2 to 0.
- Your program must halt after examining the value. The halt instruction is `1111 0000 0010 0101`.

For those who want a little challenge: please think how to code with no iteration. (no extra score)

Grading

Lab 1 takes **4%** of the total score, consisting of Check part (50%) and Report part (50%).

Check Part

- First upload your code to Learning in ZJU, then find a TA to check your code in person. TAs will first test the correctness of your program, then ask you some questions to make sure you understand what you code but not cheat.
- You can try again if you fail in checking, but there will be a penalty of -10% (of checking part) for each try.
- We strongly suggest you to make a thorough test by yourself before checking.
- We strongly suggest you to write enough comments in your code so that you will be aware of what's going on in your program and confident to answer TA's questions.

Report Part

- Report must be written **in English**, 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:
 - Algorithm. Flowchart or Pseudocode is preferred. The complexity of your algorithm will not affect your score.
 - Essential parts of your code with sufficient comments. Please only select the most important code phases and explain them.
 - Questions that TA asked you, and Answers.
- **No more than 2 A4 pages.** No template provided. Be sure to make it readable.

Penalty

- **Wrong Answer:** -10% of Check part each time.
- **Delay:** -20% of the corresponding part per day.
- **Cheating:** -100% of this lab. Additionally, -10% of the final score of this course. **Please note that uploading your answer to the Internet is also CHEATING!!!**