Lab 6b: LC-3 Executor

Write a program to execute LC-3 binary code.

Implementation Details

- You are required to write in **C** or any other high level programming language.

Instructions

The instructions you need to implement is listed in the following table.

	..00	**..01**	**..10**	**..11**
00..	BR	ADD	LD	ST
01..	JSR	AND	LDR	STR
10..	~~RTI~~	NOT	LDI	STI
11..	JMP	~~1101~~	LEA	~~TRAP~~*

- You can refer to the state machine in Figure C.2 on page 702 and the data path in Figure C.3 on page 704.
- Trap routines, interrupts, exceptions are not required.
    - The instructions RTI, 1101 are not required.
    - Privilege mode is not required. ACV is not detected by your executor.
    - The only TRAP instruction you need to implement is HALT. When HALT is executed, your executor should stop and exit.

Executing

- The default values of all registers and memory locations are x7777.
- After the executor halts, print the value of all registers.
    - When HALT is executed by your program, the value of R0, R1, R6, R7 should remain unchanged.

Input

The LC-3 binary code is input from *stdin*. It only contains 0, 1, and endline characters.

Sample 1

```
0011000000000000
1111000000100101
```

Sample 2

```
0011000000000000
0010000000001000
0101010010100000
0001011000000000
0001100011000011
0101001011000100
0101001001000000
0000010000000001
0001010010100001
1111000000100101
0001001000110100
```

- The first line is the starting address of the program.

Output

Print the results to *stdout*.

Sample 1

```
R0 = x7777
R1 = x7777
R2 = x7777
R3 = x7777
R4 = x7777
R5 = x7777
R6 = x7777
R7 = x7777
```

Sample 2

```
R0 = x1234
R1 = x0000
R2 = x0000
R3 = x2468
R4 = x48D0
R5 = x7777
R6 = x7777
R7 = x7777
```

- Print the result by `R%d = x%04X\n`. That is:
  - There are space characters between `=`.
  - The hexadecimal number should use capital letters.