* Main:
  + Load hex codes to register
  + la $s1,hexcode Loads entire hex code to $s1
* Loop:
  + Increment register to access each hex code
  + (lw $t1,0($s1)) Load hex code that $s1 points to $t1
  + (addi $s1,$s1,4) Increment $s1 to next hex code
  + (beqz $t1,exit) If hex code is 0, branch to exit
  + Jump to Print\_Format to determine opcode and print format
  + Use andi to isolate opcode bits
  + Use beq to branch to format type
* Decode\_Instruction
  + Rinst:
    - Break down code and save onto array called Parsed based on format. Each item in the array will be each item of the hex code (Opcode, RS1, RS2, RD, Func3, Func7, or Immediate)
  + Iinst:
    - Break down code and save onto array called Parsed based on format. Each item in the array will be each item of the hex code (Opcode, RS1, RS2, RD, Func3, Func7, or Immediate)
  + Sinst:
    - Break down code and save onto array called Parsed based on format. Each item in the array will be each item of the hex code (Opcode, RS1, RS2, RD, Func3, Func7, or Immediate)

Access code

Determine if it’s zero

If it is, jump to exit