```
1: ###############################################################################
2: # Ethan West, CS 2318-253, Assignment 2 Part 1 Program B
3: #
4: # This program is designed to check if the bits at the 16 & 256 place value
5: # positiona of a number are both 1.
6: # It is to print a 0 if the 2 bits are both 1 & to print a 1 if at least 1
7: # of the 2 bits is 0.
8: ############################ data segment ###################################
9:      .data
10: legend:     .asciiz "0 = both 1, 1 = at least a 0\n"
11: inPrompt:   .asciiz "Enter an integer: "
12: outLab:     .asciiz "Integer entered is of type "
13: ############################ code segment ###################################
14:         .text
15:         .globl main
16: main:
17:         li $v0, 4
18:         la $a0, legend
19:         syscall         # print legend
20:         la $a0, inPrompt
21:         syscall         # print input prompt
22:         li $v0, 5
23:         syscall         # read integer input
24:         move $t0, $v0
25:
26:
27:
28:
29:         li $v0, 4
30:         la $a0, outLab
31:         syscall         # output label
32:         li $v0, 1
33:
34:         ###########################################################
35:         # Insert NO MORE THAN 6 lines of code that involve ONLY
36:         #   bit manipulating instructions (ANDing, ORing, XORing,
37:         #   NORing and shifting - only whatever that are needed)
38:         # so that the program will work just like the sample runs
39:         # shown at the bottom (some blank lines edited out).
40:         # HINT: Risking telling the obvious, the instructions you
41:         #       insert are related to making the value in $a0 to
42:         #       the desired value (which should be either 0 or 1
43:         #       when printed as an integer).
44:         # You should test your completed program for AT LEAST the
45:         # test cases shown.
46:         ###########################################################
47:
48:
49:         andi $t1, $t0, 0x10
50:         srl $t1, $t1, 4
51:         andi $t2, $t0, 0x100
```

```
52:         srl $t2, $t2, 8
53:         and $t0, $t1, $t2
54:         xori $a0, $t0, 0x1
55:
56:
57:         syscall         # display desired output
58:
59:         ############################################################
60:
61:         li $v0, 10      # exit gracefully
62:         syscall
63:
64: ########################## sample test runs ##############################
65: # 0 = both 1, 1 = at least a 0
66: # Enter an integer: 16
67: # Integer entered is of type 1
68: # -- program is finished running --
69: #
70: # Reset: reset completed.
71: # 0 = both 1, 1 = at least a 0
72: # Enter an integer: 256
73: # Integer entered is of type 1
74: # -- program is finished running --
75: #
76: # Reset: reset completed.
77: # 0 = both 1, 1 = at least a 0
78: # Enter an integer: 272
79: # Integer entered is of type 0
80: # -- program is finished running --
81: #
82: # Reset: reset completed.
83: # 0 = both 1, 1 = at least a 0
84: # Enter an integer: 12345678
85: # Integer entered is of type 1
86: # -- program is finished running --
87: #
88: # Reset: reset completed.
89: # 0 = both 1, 1 = at least a 0
90: # Enter an integer: 87654321
91: # Integer entered is of type 0
92: # -- program is finished running --
93: ######################## end sample test runs ############################
```