

```

1 #####
2 # <Ben Hunt>, CS 2318-253, Assignment 2 Part 1 Program B
3 #####
4 # Checks if a user-entered int between 0 & 255 (inclusive) is less than 32
5 # & displays 0 if so, otherwise (32 or higher) displays 1
6 # Uses Bitwise operations to acheive this goal. AND OR NOR XOR ect.
7 # 14 Lines or less added after #read input integer
8 ##### data segment #####
9     .data
10 legend1:    .ascii "0: less than 32\n"
11 legend2:    .ascii "1: 32 or higher\n"
12 inPrompt:   .ascii "Enter an integer between 0 and 255: "
13 outLab:     .ascii "It is "
14 ##### code segment #####
15
16 .text
17     .globl main
18 main:
19     li $v0, 4
20     la $a0, legend1
21     syscall          # print legend line 1
22     la $a0, legend2
23     syscall          # print legend line 2
24     la $a0, inPrompt
25     syscall          # print input prompt
26
27     li $v0, 5
28     syscall          # read input integer
29
30
31 #####
32 # Write NO MORE THAN 14 lines of code that involve using
33 # ONLY the following:
34 # - syscall
35 # - syscall supporting instructions (e.g.: li to load $v0)
36 # - instruction to make a saved copy
37 # - bit manipulating instructions (ANDing, ORing, XORing,
38 #   NORing and shifting - only whatever that are needed)
39 # so that the program will work just like the sample runs
40 # shown at the bottom.
41 # You MUST test your completed program for AT LEAST the
42 # test cases shown (and include the result in hardcopy).
43 #####
44
45     move $t0, $v0      # $t0 = $v0 save current number in temporary variable
46     srl $t0, $t0, 0x02  # shift $t0 right by 2 places
47     or $t0, $t0, $v0    # or $t0 with $v0 (temporary variable from earlier)
48     move $t1, $t0      # save result in $t1, $t1 = $t0
49     srl $t0, $t0, 0x01  # Shift $t0 right by 1 place
50     or $t0, $t0, $t1    # or $t0 with $t1
51     andi $t0, $t0, 0x3F # and $t0 with 0x00111111 clear 2 MSB's
52     srl $t0, $t0, 0x05  # shift $t0 right by 5 places
53
54     la $a0, outLab      # put address into $a0
55     li $v0, 4           # print string (It is)
56     syscall
57
58     move $a0, $t0        # $a0 = $t0 (Preparing for output,
59                          # $v0 requires $a0 as argument. so have to set $a0)
60     li $v0, 1           # print integer
61     syscall
62
63
64
65
66

```

```

67
68
69
70
71
72
73
74
75
76
77
78
79 #####
80
81     li $v0, 10           # exit
82     syscall
83
84 ##### sample test runs #####
85 # 0: less than 32
86 # 1: 32 or higher
87 # Enter an integer between 0 and 255: 0
88 # It is 0
89 # -- program is finished running --
90 #
91 #
92 # Reset: reset completed.
93 #
94 # 0: less than 32
95 # 1: 32 or higher
96 # Enter an integer between 0 and 255: 31
97 # It is 0
98 # -- program is finished running --
99 #
100 #
101 # Reset: reset completed.
102 #
103 # 0: less than 32
104 # 1: 32 or higher
105 # Enter an integer between 0 and 255: 32
106 # It is 1
107 # -- program is finished running --
108 #
109 #
110 # Reset: reset completed.
111 #
112 # 0: less than 32
113 # 1: 32 or higher
114 # Enter an integer between 0 and 255: 64
115 # It is 1
116 # -- program is finished running --
117 #
118 #
119 # Reset: reset completed.
120 #
121 # 0: less than 32
122 # 1: 32 or higher
123 # Enter an integer between 0 and 255: 128
124 # It is 1
125 # -- program is finished running --
126 #
127 #
128 # Reset: reset completed.
129 #
130 # 0: less than 32
131 # 1: 32 or higher
132 # Enter an integer between 0 and 255: 255

```

```
133 # It is 1
134 # -- program is finished running --
135 ##### end sample test runs #####
```