CSE 450: Quiz # 5

Name: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Consider a switch statement that consists of the switch keyword followed by an expression wrapped in parentheses. The expression is evaluated and the result is sequentially tested by equality against the successive case statements. A case statement consists of the keyword case, followed by a literal, then a colon (":"), and lastly a statement that is executed if that case's literal is equal to the expression. After executing a statement, the execution of the switch ends (no break statement is needed). There is a special default case that always matches if no other case was met. If a default case is present, it must be the last case.

Below is an example tubular source file and symbol table; to the right is the generated Tube Intermediate Code with lines removed corresponding to each jump. We've shown the first of the missing jumps as an example (in bold). Please fill in the six remaining missing jump instructions.

Tubular Source Code:

val x = random(4);

val y = random(2);

switch (x + 1)

case 1: { y = 7; }

case 3: y = y && 8;

default: { x = y || 9; y = x; }

print(y);

Symbol Table:

x -> s1

y -> s2

Incomplete Tube IC:

val\_copy 4 s3

random s3 s4

val\_copy s4 s1

val\_copy 2 s5

random s5 s6

val\_copy s6 s2

val\_copy 1 s7

add s1 s7 s8

test\_eq 1 s8 s9

**jump\_if\_n0 s9 switch\_1\_case\_1**

test\_eq 3 s8 s9

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

switch\_1\_case\_1:

val\_copy 7 s10

val\_copy s10 s2

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

switch\_1\_case\_2:

test\_nequ 0 s2 s11

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

val\_copy 8 s12

test\_nequ 0 s12 s11

and\_2\_end:

val\_copy s11 s2

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

switch\_1\_default:

test\_nequ 0 s2 s13

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

val\_copy 9 s14

test\_nequ 0 s14 s13

or\_3\_end:

val\_copy s13 s1

val\_copy s1 s2

switch\_1\_end:

out\_val s2

out\_char '\n'