Notes About Given Code:

* switch\_int1:
  + move x to register eax, then tv64(size 4 at ebp-8)
  + compare to cases 2,1,3.
    - jump if equal
    - Perform given instructions
  + if not equal to any cases, jump to default
  + jump to LN1
* switch\_enum1:
  + move x to register eax, then tv64(size 4 at ebp-8)
  + compare to cases 2,1.
    - jump if equal
    - Perform given instructions
  + if not equal to any cases, jump to default
  + jump to LN1
* switch\_int2:
  + move x to register eax, then tv64(size 4 at ebp-8)
  + add 4 to tv64
  + compare to int 54, if above, jump to default
  + else, compare to cases in order from smallest to largest
  + if equal to a case, Perform given instructions
  + jump to LN1
* switch\_enum2:
  + move x to register eax, then tv64(size 4 at ebp-8)
  + sub 1 from tv64
  + compare tv64 to largest value (5), if above jump to default
  + else, compare to cases in order from smallest to largest
  + if equal to a case, Perform given instructions
  + jump to LN1
* modswitch\_int2:
  + move x to register eax, then tv64(size 4 at ebp-8)
  + add 4 to tv64
  + compare to int 54, if above, jump to default
  + else, compare to cases in order from smallest to largest
  + if equal to a case, Perform given instructions
  + jump to LN1

Notes About Possible Rules:

* When in the switch statement, the compiler is always compared to the lowest case values first, then works its way up.
* If there are more than 3 cases, the compiler takes the value of the largest case, adds or subtracts a value O, compares it to the switch value. If the switch value (x) is greater than the largest case value +- O, the compiler jumps straight to the default case.