**Problem 3:**

1. R: arbitary logical position

v: variable

t: term

(∃x₀.R)[t/v] = ∃ x₀.R[t/v]?

* We have to consider the variable x₀ in R
* If R is x₀ free. This condition ensures that subtitution preserves the meaning of quantified expression.
* If R is t free. It means that subtituing t for v in R does not introduce any new variable.

1. Ex: (∃x₀.R)[t/v] = ∃ x₀.R[t/v] does not hold

R(x\_0) = “x₀ % 2 == 0”

v = x₀

t = x₀ + 1

(∃x₀.R)[t/v] = (∃x₀. (x₀ % 2==0))[t/v]

-> (∃x₀. (x₀ % 2==0))[ x₀ + 1/ x₀]

-> (∃x₀. (x₀+1 % 2==0))

- there is an x₀ sothat x₀+1 % 2 == 0

(∃x₀.R)[t/v] ≠ ∃x0.R[t/v]

In this example the logical proposition doesn’t hold because ((∃x₀.R)[t/v] say that: “there’s x₀ that x₀ % 2 == 0 where as ∃ x₀.R[t/v] say that also x₀+1%2==0. -> contradiction ⊥.

c)Floyd’s rule {P } v := t {∃ v₀.(P [v₀/v] ∧ v = t[v₀/v])}.

1. {P} v := t {Q[t/v]} (Hoare's rule)

2. {P} v := t {Q[t/v] ∧ v = t} (Conjunction introduction)

3. {P} v := t {∃v₀.(Q[t/v] ∧ v = t[v₀/v])} (Existential introduction)

4. {P} v := t {∃v₀.(P[v₀/v] ∧ v = t[v₀/v])} (Substitution)

**Problem 4:**

1. Determine under which circumstances the following program establishes 0 <= y < 100.

Refactor the code:

if (x < 34 && x == 2)

y := x + 1;

else if (x < 34 && x != 2)

y := 233;

else if (x >= 34 && x < 55)

y:=21;

else if (x >= 34 && x >= 55)

y:=144;

Conditions:

1. For any x in the range x==2, the value of y will be x + 1, which satisfies 0 <= y < 100.
2. For any x in the range x<2 or 2 < x < 34, the value of y will be 233, which does not satisfy the condition
3. For any x in the range 34 <= x < 55, the value of y will be 21, which satisfies 0 <= y < 100.
4. For any x in the range 55 <= x < 99, the value of y will be 144, which does not satisfy the condition.

b) if B1 then C1 elif B2 then C2 else C3

{P ∧ B1} C1 {Q}, {P ∧ ¬B1 ∧ B2} C2 {Q}, {P ∧ ¬B1 ∧ ¬B2 ∧ B3} C3 {Q}

if B1 then C1 elif B2 then C2 else C3