Assignment#2 Key

1.

Prove: ((P => Q) => Q) <=> (P v Q)

We can simplify the left part of the assumption by the following steps:

(P => Q) => Q

(~P v Q) => Q

~(~P v Q) v Q

(P ^ ~Q) v Q

(P v Q) ^ (~Q v Q)

(P v Q)

This shows both sides are equivalent.

We can also attack as follows (essentially a Truth Table approach.

Case 1: Assume Q is False, P is False

Then have ((False => False) => False) <=> (False v False)

False => False is true, so left side is (True => False), which is False, as is False v False

Case 2: Assume Q is False, P is True

Then have ((True => False) => False) <=> (True v False)

True => False is false, so left side is (False => False), which is True, as is True v False

Case 3: Assume Q is True, P is False

Then have ((False => True) => True) <=> (False v True)

False => True is True, so left side is (True => True), which is True, as is False v True

Case 4: Assume Q is True, P is True

Then have ((True => True) => True) <=> (True v True)

True => True is True, so left side is (True => True), which is True, as is True v True

This shows the expression is a Tautology.

2.

The table of transitions is like below:

States	**Nickels**	**Dimes**	**Quarters**	**Cancel**	**Buy**
>S0	S5	S10	S25	S0	S0
S5	S10	S15	S30	S0	S0
S10	S15	S20	S35	S0	S0
S15	S20	S25	S40	S0	S0
S20	S25	S30	S45	S0	S0
S25	S30	S35	S50	S0	S0
S30	S35	S40	S0	S0	S0
S35	S40	S45	S0	S0	S0
S40	S45	S50	S0	S0	S0
S45	S50	S0	S0	S0	S0
S50	S5	S10	S25	S0	SFinal
SFinal	S5	S10	S25	S0	S0

In a real machine, we would need to have output, e.g., return coins and deliver purchase.