Question 1:

Hardest Sudoku Puzzle:

```
puzzle = [
    [8, 0, 0, 0, 0, 0, 0, 0, 0],
    [0, 0, 3, 6, 0, 0, 0, 0, 0],
    [0, 7, 0, 0, 9, 0, 2, 0, 0],
    [0, 5, 0, 0, 0, 7, 0, 0, 0],
    [0, 0, 0, 0, 4, 5, 7, 0, 0],
    [0, 0, 0, 1, 0, 0, 0, 3, 0],
    [0, 0, 1, 0, 0, 0, 0, 6, 8],
    [0, 0, 8, 5, 0, 0, 0, 1, 0],
    [0, 9, 0, 0, 0, 0, 4, 0, 0]

]
```

Solution found:

[8, 1, 2, 7, 5, 3, 6, 4, 9]
[9, 4, 3, 6, 8, 2, 1, 7, 5]
[6, 7, 5, 4, 9, 1, 2, 8, 3]
[1, 5, 4, 2, 3, 7, 8, 9, 6]
[3, 6, 9, 8, 4, 5, 7, 2, 1]
[2, 8, 7, 1, 6, 9, 5, 3, 4]
[5, 2, 1, 9, 7, 4, 3, 6, 8]
[4, 3, 8, 5, 2, 6, 9, 1, 7]
[7, 9, 6, 3, 1, 8, 4, 5, 2]

Solution In table format:

8	1	2	7	5	3	6	4	9
9	4	3	6	8	2	1	7	5
6	7	5	4	9	1	2	8	3
1	5	4	2	3	7	8	9	6
3	6	9	8	4	5	7	2	1
2	8	7	1	6	9	5	3	4
5	2	1	9	7	4	3	6	8
4	3	8	5	2	6	9	1	7
7	9	6	3	1	8	4	5	2

Question 2:

We can check if a puzzle has multiple solutions by adding additional constraints to our solver. After finding the first solution, we can introduce a new constraint to our solver that the propositional variables that were true in the first solutions can NOT be all true and then ask the solver to solve again. This forces the solver to look for a solution that is at least one assignment different from the original solution and therefore find a new solution. This can be extended to more than two solutions by just repeating the same process if theoretically a puzzle can have more than two solutions.

We implemented a sample method solve_with_precluded_solution in our code that introduces this constraint. We found that the solution to part 1 is unique. Please check our code. We also changed the puzzle given to the puzzle found in question 1.

This is an example puzzle that has two solutions:

```
[
    [2, 9, 5, 7, 4, 3, 8, 6, 1],
    [4, 3, 1, 8, 6, 5, 9, 0, 0],
    [8, 7, 6, 1, 9, 2, 5, 4, 3],
    [3, 8, 7, 4, 5, 9, 2, 1, 6],
    [6, 1, 2, 3, 8, 7, 4, 9, 5],
    [5, 4, 9, 2, 1, 6, 7, 3, 8],
    [7, 6, 3, 5, 3, 4, 1, 8, 9],
    [9, 2, 8, 6, 7, 1, 3, 5, 4],
    [1, 5, 4, 9, 3, 8, 6, 0, 0]
]
```

Question 3:

$$\Delta := \{C_1, C_2, C_3, C_4, C_5\}$$

$$:= \{(P_1, \neg P_2, \neg P_3), (\neg P_1, P_2, P_3), (\neg P_1, \neg P_3), (P_1, P_2), (\neg P_2, P_3)\}$$

- Resolve C_1 & C_5 to remove P_3 & get C_6:

  $\{P_1, \neg P_2\}$

- Resolve C_6 and C_4 to remove P_2 & get C_7:

  $\{P_1\}$

- Resolve C_2 & C_3 to remove P_3 & get C_8:

  $\{\neg P_1, P_2\}$

- Resolve C_8 and C_5 to remove P_2 & get C_9:

  $\{\neg P_1, P_3\}$

- Resolve C_9 and C_3 to remove P_3 and get C_{10}:

  $\{\neg P_1\}$

- Resolve C_7 and C_{10} to remove P_1 & get C_{11}:

  $\{\} \Rightarrow$ empty clause

$\Rightarrow$ Encountered an empty clause

$\therefore$ Set of clauses is unsatisfiable

(If Δ is satisfiable then C_{11} is satisfiable which is impossible)

Question 4:

unit clause

$$\Delta_0 = \{\{p_1, p_2, \neg p_4\}, \{\neg p_4, p_6\}, \{\neg p_1, \neg p_2, p_3\}, \{\neg p_6\}, \{p_1, p_6, \neg p_5\}, \{\neg p_4, \neg p_6\}, \{p_1, p_2\}, \{p_1, p_7\}, \{\neg p_1, \neg p_7\}\}$$

1st) Unit Propagation rule on p_6:

- Remove all clauses containing $\neg p_6$ (including the unit clause)
- Remove all instances of p_6 from clauses in the formula

$$\{\{p_1, p_2, \neg p_4\}, \{\neg p_4\}, \{\neg p_1, \neg p_2, p_3\}, \{p_1, \neg p_5\}, \{p_1, p_2\}, \{p_1, p_7\}, \{\neg p_1, \neg p_7\}\}$$

2nd) Pure literal rule on p_4:

p_4 only appears as $\neg p_4 \Rightarrow$ delete all clauses containing the variable. (p_4 = false)

$$\{\{\neg p_1, \neg p_2, p_3\}, \{p_1, \neg p_5\}, \{p_1, p_2\}, \{p_1, p_7\}, \{\neg p_1, \neg p_7\}\}$$

3rd) Resolution on p_1:

$$\{\{\neg p_2, p_3, \neg p_5\}, \{p_2, \neg p_2, p_3\}, \{p_7, \neg p_2, p_3\}, \{\neg p_5, \neg p_7\}, \{p_2, \neg p_7\}, \{p_7, \neg p_7\}\}$$

clashing clause

clashing clause

Remove clashing clauses:

$$\{\{\neg p_2, p_3, \neg p_5\}, \{p_7, \neg p_2, p_3\}, \{\neg p_5, \neg p_7\}, \{p_2, \neg p_7\}\}$$

4th) Pure literal rule on P_3:

P_3 only appears as only P_3 $\Rightarrow$ Remove all clauses containing P_3; (P_3 = true)

$\{\{\neg P_5, \neg P_7\}, \{P_2, \neg P_7\}\}$

5th) Pure literal rule on P_7: (P_7 = false)

P_7 only appears as $\neg P_7$ $\Rightarrow$ Remove all clauses containing $\neg P_7$:

$$\emptyset$$

$\therefore$ The set of clauses is satisfiable.

Question 5:

$$\Delta_0 := \{C_1 = \{1,2,3\},\ C_2 = \{\neg 1, \neg 2, \neg 3\},\ C_3 = \{\neg 1, 2, 3\},$$
$$C_4 = \{\neg 2, 3\},\ C_5 = \{2, \neg 3\}\}$$

M	Δ	rule
1	$\{C_1 = \{\cancel{1,2},3\}, C_2 = \{\cancel{\neg 1}, \neg 2, \neg 3\}, C_3 = \{\cancel{\neg 1}, 2, 3\}, C_4 = \{\neg 2, 3\}, C_5 = \{2, \neg 3\}\}$	Decide
$1. \neg 2$	$\{C_1 = \{1,2,3\}, C_2 = \{\neg 1, \neg 2, \neg 3\}, C_3 = \{\neg 1, 2, 3\}, C_4 = \{\neg 2, 3\}, C_5 = \{2, \neg 3\}\}$	Decide
$1. \neg 2\ 3$	$\{C_1 = \{1,2,3\}, C_2 = \{\neg 1, \neg 2, \neg 3\}, C_3 = \{\neg 1, 2, 3\}, C_4 = \{\neg 2, 3\}, C_5 = \{2, \neg 3\}\}$	Propagate C_3 (become unit clauses according to our partial assignment M)
	$\{C_1 = \{1,2,3\}, C_2 = \{\neg 1, \neg 2, \neg 3\}, C_3 = \{\neg 1, 2, 3\}, C_4 = \{\neg 2, 3\}, C_5 = \{2, \neg 3\}\}$	
$1.\ 2$	$\{C_1 = \{1,2,3\}, C_2 = \{\neg 1, \neg 2, \neg 3\}, C_3 = \{\neg 1, 2, 3\}, C_4 = \{\neg 2, 3\}, C_5 = \{2, \neg 3\}\}$	Backtrack (C_5 is conflicti[illegible] clause)

$1 . \; 2 \; \neg 3$ $\{C_1 = \{1,2,3\}, C_2 = \{\neg 1, \neg 2, \neg 3\}, C_3 = \{\neg 1, 2, 3\}, C_4 = \{\neg 2, 3\}, C_5 = \{2, \neg 3\}\}$ Propagate C_2

$\neg 1$ $\{C_1 = \{1,2,3\}, C_2 = \{\neg 1, \neg 2, \neg 3\}, C_3 = \{\neg 1, 2, 3\}, C_4 = \{\neg 2, 3\}$, ($C_4$ is conflicting) $C_5 = \{2, \neg 3\}\}$ Backtrack

$\neg 1 . \; 2$ $\{C_1 = \{1,2,3\}, C_2 = \{\neg 1, \neg 2, \neg 3\}, C_3 = \{\neg 1, 2, 3\}, C_4 = \{\neg 2, 3\}, C_5 = \{2, \neg 3\}\}$ Decide

$\neg 1 . \; 2 \; 3$ $\{C_1 = \{1,2,3\}, C_2 = \{\neg 1, \neg 2, \neg 3\}, C_3 = \{\neg 1, 2, 3\}, C_4 = \{\neg 2, 3\}, C_5 = \{2, \neg 3\}\}$ Propagate (C_4 becomes unit clause)

All satisfied

$\langle M, \Delta_0 \rangle$

Therefore DPLL was able to prove that assignment $\{1: false, 2: true, 3: true\}$ satisfies Δ_0