OPTION 1: Implement the verification method (called RS-method), described below, of whether a given formula in the propositional calculus is a tautology.

Assume that $L_0$ is a language of order zero (propositional calculus). Letter $S$ (with indices) will denote finite sequences $(\alpha_1, \alpha_2, \ldots, \alpha_m)$ of formulas in $L_0$.

If $S_1 = (\alpha_1, \alpha_2, \ldots, \alpha_m)$ and $S_2 = (\beta_1, \beta_2, \ldots, \beta_n)$ and $\alpha, \beta$ are any formulas then $S_1$, $\alpha$, $S_2$ and $S_1$, $\alpha$, $\beta$, $S_2$ denote sequences $(\alpha_1, \alpha_2, \ldots, \alpha_m, \alpha, \beta_1, \beta_2, \ldots, \beta_n)$ and $(\alpha_1, \alpha_2, \ldots, \alpha_m, \alpha, \beta, \beta_1, \beta_2, \ldots, \beta_n)$.

A formula in $L_0$ is indecomposable if it is either a propositional variable or negation of propositional variable.

A sequence is indecomposable provided it is formed only of indecomposable formulas

A sequence is fundamental if it simultaneously contains a formula $\alpha$ and its negation $\neg \alpha$.

We consider two types of schemas: $S_1/S_2$ and $S_1/(S_2; S_3)$. $S_1$ is called is a premise and $S_2$, $S_3$ conclusions. If a schema is of the form $S_1/(S_2; S_3)$, then $S_2$ is left conclusion and $S_3$ right conclusion. The following 7 schemas are considered:

$$[S_1,(\alpha \lor \beta),S_2]/[S_1,\alpha,\beta,S_2] \quad [S_1,(\alpha \land \beta),S_2]/[S_1,\alpha,S_2; S_1,\beta,S_2]$$

$$[S_1,\neg(\alpha \lor \beta),S_2]/[S_1,\neg \alpha,\neg \beta ,S_2] \quad [S_1,\neg(\alpha \land \beta),S_2]/[S_1,\neg \alpha,S_2; S_1,\neg \beta,S_2]$$

$$[S_1,(\alpha \rightarrow \beta),S_2]/[S_1,\neg \alpha,\beta,S_2] \quad [S_1,\neg(\alpha \rightarrow \beta),S_2]/[S_1,\alpha,S_2; S_1,\neg \beta,S_2]$$

$$[S_1,\neg \neg \alpha,S_2]/ [S_1,\alpha,S_2],$$

where $S_1$ is indecomposable in all schemas.

Let’s denote by $D(\alpha)$ the diagram (tree) built for $\alpha$ using the above seven schemas.

FACT: Formula $\alpha$ is a propositional tautology if and only if all end sequences in the diagram $D(\alpha)$ (leaves in $D(\alpha)$) are fundamental.

System Input: propositional formula; System output: yes/no

OPTION 2: Write a program which solves N queens puzzle ($8 < N < 13$).

Input: $N$; Output: $N \times N$ Boolean array with 1’s showing final position of queens.

Your program should use heuristics minimizing the search space.