Resolution-refutation as Search

You need to implement a propositional logic theorem-prover based on the resolution-refutation algorithm, posing it as a search problem.

Input Format:
-   The first line contains two integer values 'n' and 'm'. The value 'n' denotes the number of formulas in the KB, and 'm' denotes the mode.
- The next 'n' lines contain one formula per line.
- The last line contains the query that needs to be proved.
- There is no space between two consecutive operators/variables.

The following characters are used to denote different operators:
OR : |
AND : &
NOT : !
IMPLICATION : >
IFF (bidirectional) : =
OPENING BRACKET : (
CLOSING BRACKET :)

Example input:
4 0
P|(Q/\(R>T))
P>R
Q>T
Q>(R=T)
R

The output should be 1 if the KB entails the query, and 0 otherwise. Additionally, there should be an option to print the resolution steps used in the proof.

Output Format:
If m = 0, print only the result (integer value 0/1)
If m = 1, first print the resolution steps (one step per line) and then print the result (integer value 0/1) in the last line.

Part-0
Convert each formula into CNF.

Part-A
(1) Implement the resolution-refutation algorithm posing it as an uninformed search problem.
(2) Implement the resolution-refutation algorithm posing it as a greedy search problem. For this, design a task-specific heuristic function.

Part-B
Run the above two implementations on inputs of different sizes and complexities, and analyze and compare them in terms of the number of nodes explored and execution time.

Deliverables

(1) A folder containing your codes and a detailed readme file. You may use any programming language.

(2) A report (PDF) describing the experimental details, results, observations, analyses, etc.

(3) Create a single zipped file name <RollNo_Assig3.zip> containing the above two and upload.

General instructions

(1) Do not paste your codes in the report.

(2) Cite all the resources in the report.

(3) If anything is missing or not clear from the above description, you may make appropriate assumptions and clearly mention them in the report.

(4) A submission which does follow any of the guidelines will be awarded a penalty.

(5) Plagiarism will result in a zero in this assignment, and an additional penalty in the total score in the course.

Penalty for late submission

There will be a penalty of 25% per day. A submission which is >=3 days late will not be evaluated. The time recorded in google-classroom will be considered.