1.

In the domain of colloquial arguments formed from general statements, assume that an argument is formed by a premise and a conclusion statement. Conclusions can be inferred through chaining arguments. Some arguments may, however, be fallacious and conclusions drawn from those arguments must be discarded. Note that a fallacy is a flaw in the structure of an argument. Assume that the following GNU Prolog program is written to help a user check what conclusions can be drawn from their arguments.

Predicate Definitions:

- `argument(Prem,Conc)`: the user has asserted an argument with premise `Prem` and conclusion `Conc`
- `fallacy(Prem,Conc)`: a pattern stating that any argument concluding `Conc` from `Prem` is a fallacy
- `infer(Root,Conc)`: a conclusion `Conc` can be inferred from `Root`
- `getConclusions(Root)`: prints all conclusions that can be inferred from `Root`
- `check(X,Y)`: checks whether an asserted argument is fallacious.

Program:

```
argument(a,b).
argument(b,c).
argument(a,e).
fallacy(a,c).
fallacy(a,e).

infer(Root,Conc):-argument(Root,Conc).
infer(Root,Conc):-argument(Root,Z),infer(Z,Conc).

check(X,Y):-fallacy(X,Y),argument(X,Y),!,fail.
check(X,Y).

getConclusions(Root):-infer(Root,Conc),write(Conc),nl,fail.
```

(i) Analyse how Prolog resolves a query for goal `?-getConclusions(a).` In your description, please list key steps taken by the algorithm to satisfy the goal, justify its output, and highlight backtracking and matching steps if any occur. [4 marks]

(ii) Modify the program such that conclusions inferred from arguments that are fallacious are discarded with the help of predicate `check(X,Y)`. Justify your solution and discuss any changes in the execution in comparison with the previous program when querying for goal `?-getConclusions(a).` You may avoid reporting unaffected clauses for this question. [3 marks]