Compiler Construction

You have been provided with the description of a programming language, J, intended for scripting applications. Its syntax is similar to a cut-down version of Java in that it consists of function definitions which have bodies containing if-then-else, while-do, assignments and (typed) declarations of variables. Only one statement or keyword may occur on a line so that it is sufficient to describe the start of a loop iteration with its line number. You need to explain to your boss the alternatives for implementing this so that a decision may be made as to the best implementation strategy.

The choice is between:

(a) compiling J to machine code;

(b) compiling J to “interpreted byte code”, and then interpreting this;

(c) parsing J to a syntax tree representation and then interpreting this using a function which walks the tree;

(d) keeping J in a text file and then interpreting it by reading each line (and acting on it) as and when the line is required.

For each of (a)–(d), (i) summarise the main phases of work that are done before execution in each case, giving a brief explanation of the main actions of the main interpreter loop (if any) during execution, and (ii) for each of the following possible erroneous forms, explain whether the error would be found before or during execution: malformed syntax, undeclared variable, type error, division by zero.

You are not expected to argue for or against any of the alternatives.