Problemas:

1)

O facto de não conseguir ter todas as palavras reservadas como tokens:  
Exemplo do main, dos reads e dos writes.

Foi necessário os tomar como literais pelo parser, não se é alguma limitação.  
Não fiquei satisfeita com a solução porém ficou a funcionar.

Fiquei com a ideia de que quando uma palavra pode ser usada como identificador e também como palavra-chave, é preferível tratá-la como literal ('main', 'read', etc.) diretamente nas regras do parser, porém não encontrei fundamento teórico para esta reflexão.

2)

Acessos a vetores (v[i]) confundidos com chamadas de função (v(x)) foi outro problema.

Isto porque aqui se colocou um problema de ambiguidade da gramática que tinha criado até então, ou seja, garantir que o parser conseguisse distinguir corretamente entre chamadas de função e acessos a vetores.

A resolução passou por explicitar regras distintas para cada caso, eliminando a ambiguidade.

3)

Definição da regra expressão

Como foi seguido as regras facultadas no enunciado uma a uma, inicialmente, a gramática apenas tratava expressões básicas com operadores e identificadores, mas não suportava corretamente alguns casos mais complexos, como chamadas a funções com argumentos (fact(n)), acessos a elementos de vetores (v[i]), ou operações de casting ((int) x).

Sempre que se tentava usar estas construções em testes, surgiam erros de parsing, mesmo com sintaxe válida

A solução passou por expandir a regra adicionando casos específicos para garantir os casos mais complexos :

Acessos a vetores: IDENTIFICADOR [ expressao ]

Chamadas a funções definidas pelo utilizador: IDENTIFICADOR ( argumentos )

Operações de casting: ( tipo ) expressao