Linguagem P--

1. <programa> ::= program ident ; <corpo> .
2. <corpo> ::= <dc> begin <comandos> end
3. <dc> ::= <dc_c> <dc_v> <dc_p>
4. <dc_c> ::= const ident = <numero> ; <dc_c> | λ
5. <dc_v> ::= var <variaveis> : <tipo_var> ; <dc_v> | λ
6. <tipo_var> ::= real | integer
7. <variaveis> ::= ident <mais_var>
8. <mais_var> ::= , <variaveis> | λ
9. <dc_p> ::= procedure ident <parametros> ; <corpo_p> <dc_p> | λ
10. <parametros> ::= (<lista_par>) | λ
11. <lista_par> ::= <variaveis> : <tipo_var> <mais_par>
12. <mais_par> ::= ; <lista_par> | λ
13. <corpo_p> ::= <dc_loc> begin <comandos> end ;
14. <dc_loc> ::= <dc_v>
15. <lista_arg> ::= (<argumentos>) | λ
16. <argumentos> ::= ident <mais_ident>
17. <mais_ident> ::= ; <argumentos> | λ
18. <pfalsa> ::= else <cmd> | λ
19. <comandos> ::= <cmd> ; <comandos> | λ
20. <cmd> ::= read (<variaveis>) |
    write (<variaveis>) |
    while (<condicao>) do <cmd> |
    if <condicao> then <cmd> <pfalsa> |
    ident := <expressão> |
    ident <lista_arg> |
    begin <comandos> end
21. <condicao> ::= <expressao> <relacao> <expressao>
22. <relacao> ::= = | <> | >= | <= | > | <
23. <expressao> ::= <termo> <outros_termos>
24. <op_un> ::= + | - | λ
25. <outros_termos> ::= <op_ad> <termo> <outros_termos> | λ
26. <op_ad> ::= + | -
27. <termo> ::= <op_un> <fator> <mais_fatores>
28. <mais_fatores> ::= <op_mul> <fator> <mais_fatores> | λ
29. <op_mul> ::= * | /
30. <fator> ::= **ident** | <numero> | (<expressao>)
31. <numero> ::= numero_int | numero_real

Além disso:

- é preciso incluir o comando **for** (ver exemplo mais abaixo)
- comentários de única linha, entre chaves { }
- identificadores e números são itens léxicos da forma:
  - ident: sequência de letras e dígitos, começando por letra
  - número inteiro: sequência de dígitos (0 a 9)
  - número real: pelo menos um dígito, seguido de um ponto decimal, seguido de uma sequência de um ou mais dígitos

---

Exemplos de programas em P--

```
program fatorial;
{exemplo 1}
var x, aux, fat: integer;
begin
read(x);
fat:=1;
for aux:=1 to x do
begin
fat:=fat*aux;
end;
write(fat);
end.
```

```
program leimprime;
{exemplo 2}
var a: real;
var b: integer;
procedure nomep(x: real);
var a, c: integer;
begin
read(c, a);
if a<x+c then
begin
a:= c+x;
write(a);
end
else c:= a+x;
end;
begin {programa principal}
read(b);
nomep(b);
end.
```