

```

1: program PArvBin;
2: uses ArvBin;
3:
4: procedure ImprimirNo(Arvore : ArvoreBinaria);
5: begin
6:     write(Arvore^.Dado, ' ');
7: end;
8:
9: procedure PreOrdem(Arvore : ArvoreBinaria);
10: begin
11:     if not Vazia(Arvore) then
12:         begin
13:             ImprimirNo(Arvore);
14:             PreOrdem(Arvore^.Links[NoEsquerdo]);
15:             PreOrdem(Arvore^.Links[NoDireito]);
16:         end;
17: end;
18:
19: procedure ImprimirCaracteristicas(Arvore : ArvoreBinaria);
20: var
21:     NumNos, Altura : integer;
22:     CompMedio : real;
23: begin
24:     Caracteristicas(Arvore, NumNos, Altura, CompMedio);
25:     writeln('Numero de nos = ', NumNos);
26:     writeln('Altura = ', Altura);
27:     writeln('Comprimento medio = ', CompMedio);
28: end;
29:
30: var
31:     Arv : ArvoreBinaria;
32: BEGIN
33:
34:     Inicializar(Arv);
35:
36:     CriarNo(Arv, 'A');
37:     AdicionarFilho(Arv, NoEsquerdo, 'B');
38:     AdicionarFilho(Arv, NoDireito, 'E');
39:
40:     {
41:         A
42:        B   E
43:     }
44:     AdicionarFilho(Arv^.Links[NoEsquerdo], NoEsquerdo, 'C');
45:     AdicionarFilho(Arv^.Links[NoEsquerdo], NoDireito, 'D');
46:
47:     AdicionarFilho(Arv^.Links[NoDireito], NoEsquerdo, 'F');
48:
49:     AdicionarFilho(Arv^.Links[NoDireito]^Links[NoEsquerdo], NoDireito, 'G');
50:
51:     {
52:         A
53:        B   E
54:       C D  F
55:                G
56:     }
57:     { Imprime NumNos = 7, Altura = 3, CompMedio = 11/7 = 1.571429 }
58:     ImprimirCaracteristicas(Arv);
59:
60:     PreOrdem(Arv); { Imprime: A B C D E F G }
61:     writeln;

```

```

62:
63:     Deslocar(Arv, NoDireito); { Vai para o No E }
64:     writeln('Existe no esquerdo = ', ExisteNo(Arv, NoEsquerdo)); { Deve ser
TRUE }
65:     writeln('Existe no direito = ', ExisteNo(Arv, NoDireito)); { Deve ser
FALSE }
66:
67:     Deslocar(Arv, NoEsquerdo); { Vai para F }
68:     Deslocar(Arv, NoDireito); { Vai para G }
69:     writeln('Esperando G, obtido ', Arv^.Dado);
70:
71:     Deslocar(Arv, NoPai); { Vai para F }
72:     writeln('Esperando F, obtido ', Arv^.Dado);
73:
74:     Deslocar(Arv, NoRaiz); { Vai para A }
75:     writeln('Esperando A, obtido ', Arv^.Dado);
76:
77:     DisposeArvore(Arv);
78: END.

```