

```

1: program palindrom;
2: uses FilaE, PilhaE;
3:
4: function Palindromo(s : String) : boolean;
5: var
6:   F : Fila_Enc;
7:   P : Pilha_Enc;
8:   I : byte;
9:   DF : FilaE.Tipo_Do_Dado;
10:  DP : PilhaE.Tipo_Do_Dado;
11: begin
12:   FilaE.Inicializar(F);
13:   PilhaE.Inicializar(P);
14:
15:   { Adiciona todos os caracteres contidos na string
16:     na fila e na pilha }
17:   for I := 1 to length(s) do
18:     if s[I] in ['a'..'z', 'A'..'Z'] then
19:       begin
20:         DP.Dado := S[I];
21:         DF.Dado := S[I];
22:         Empilhar(DP, P);
23:         Inserir(DF, F);
24:       end;
25:
26:   Palindromo := true;
27:
28:   { Compara os caracteres retirados da pilha com os retirados
29:     da fila. A ordem de retirada da pilha e do final da string
30:     para o inicio, enquanto na fila e do inicio para o final }
31:   while (not PilhaE.Vazia(P)) and (not FilaE.Vazia(F)) do
32:     begin
33:       Desempilhar(DP, P);
34:       Retirar(DF, F);
35:
36:       if UpCase(DF.Dado) <> UpCase(DP.Dado) then
37:         begin
38:           Palindromo := false;
39:           break;
40:         end
41:       end;
42:
43:   { Esvazia a Fila }
44:   while not FilaE.Vazia(F) do
45:     Retirar(DF, F);
46:
47:   { Esvazia a Pilha }
48:   while not PilhaE.Vazia(P) do
49:     Desempilhar(DP, P);
50: end;
51:
52: var
53:   Str : String;
54: BEGIN
55:   writeln('Digite o suposto palindromo');
56:   readln(Str);
57:
58:   write('' + Str + '');
59:   if Palindromo(Str) then
60:     writeln('e um palindromo')
61:   else

```

```
62:      writeln('nao e um palindromo')
63: END.
```