



1. Considere o programa

```
1 program Teste(input, output);

2 type caixa = array [1 .. 30, 1 .. 30] of integer;
3 var C: caixa;
4     n: integer;

5 procedure Ler(var A: caixa; n: integer);
6 var i, j: integer;
7 begin readln(n);
8     for i := 1 to n do
9         for j := 1 to n do read(A[i,j]);
10 end;

11 function Verifica(A: caixa; n: integer): boolean;
12 var i, j: integer;
13 begin Verifica := true;
14     for i := 1 to n do
15         begin for j := 1 to i-1 do
16             if A[i,j] <> 0 then Verifica := false;
17             if A[i,i] <> 1 then Verifica := false;
18             for j := i+1 to n do
19                 if A[i,j] <> 0 then Verifica := false;
20         end;
21 end;

22 begin C := Ler(C, n);
23     if Verifica(C) then writeln('Sim!')
24     else writeln('Nao!')
25 end.
```

- (a) Indique e corrija três erros que encontrar no procedimento `Ler` e no programa principal, explicando porque são um erro. Utilize a numeração que identifica as linhas do programa.
- (b) Descreva o resultado produzido pelo programa e a funcionalidade de cada subprograma, após a correção.
- (c) Apresente os parâmetros formais e os parâmetros concretos do programa, após a correção.
- (d) Compare a utilização de parâmetros em subprogramas e a utilização de variáveis não locais.
- (e) Proponha uma nova versão da função `Verifica`, que seja mais eficiente do que a apresentada.

2. Escreva um programa que leia um número inteiro positivo e indique qual o seu maior dígito.

3. Considere as declarações

```
type elm = record Num, Pos: integer;
             end;
VEsparso = record Vector: array [1..400] of elm;
             Dim: integer;
             end;
```

- (a) Faça um subprograma que leia uma sequência com um número dado de inteiros e armazene numa variável do tipo `VEsparso` apenas os seus elementos não nulos, e respectivas posições. Por exemplo, dados: 3 0 0 -1 0 7 0, deve ter uma variável `V` tal que `V.Dim = 3` e `V.Vector` contém os elementos:

| | | | |
|-----|---|----|---|
| Pos | 1 | 4 | 6 |
| Num | 3 | -1 | 7 |

- (b) Faça um subprograma que imprima um `VEsparso` como um vector com todos os seus elementos. (No exemplo acima a saída produzida deve ser: 3 0 0 -1 0 7 0.)
- (c) Elabore um subprograma para adicionar dois `VEsparso`'s, sem utilizar os vectores com todos os seus elementos.
- (d) Escreva um programa que leia dois `VEsparso`'s, adicione as sequências correspondentes, se tiverem igual dimensão, e imprima o resultado no monitor.
Apresente apenas o código novo, substituindo as declarações dos subprogramas já desenvolvidos pelo respectivo cabeçalho e reticências.
- (e) Assuma que os elementos na tabela de um `VEsparso` se encontram por ordem crescente do campo `Pos` (a sua posição no vector inicial).
Implemente um método recursivo de procura binária, que devolva a posição de um inteiro, `k`, num `VEsparso`, `V`, ou um valor indicador de que `k` não se encontra em `V`.