	Departamento de Matemática da Universidade de Coimbra	
2007/2008	Métodos de Programação I	Frequência

Exercício 1

```
program calc_pot(input, output);
 var x, y : integer;
     z    : real;

 procedure ler_dados (a, b :integer);
 begin
   write('Escreva a base : '); readln(a);
   write('Escreva o expoente : '); readln(b);
 end;

 function pot (b, e : integer) : real;
 begin
   b := 0; i := 0;
   while i < e do
   begin
      i := i+1; potencia := potencia * b;
   end;
 end;

 procedure escreve_resultado (b, e : integer; r : real);
 begin
    writeln(b:1, ' ^ ', e:1, ' = ', r:4:2);
 end;

begin       {Programa Principal}
  ler_dados(x, y);
  z:=pot(x, y);
  escreve_resultado(x, y, z);
```

1.a) Descreva a o que faz cada um dos procedimentos acima e qual a respectiva lista de parâmetros.

1.b) Identifique e corrija todos os erros que encontra no programa acima.

1.c) Descreva todas as variáveis (se são locais ou não) e qual o seu domínio.

1.d) O programa trabalha correctamente para todos os possíveis valores de $x, y \in \mathbb{Z}$? Porquê?

1.e) O ciclo pot é o mais adequado para a situação em causa? Explique.

Exercício 2

2.a) Defina, na linguagem Pascal, um tipo, a que chamará *Vector*, capaz de guardar uma tabela de inteiros com, no máximo, 100 elementos.
(Tenha em atenção que este limite máximo deve ser declarado de modo a tornar o mais simples possível qualquer alteração posterior.)

2.b) Escreva e implemente em Pascal, um procedimento para ler e guardar numa tabela do tipo *Vector*, uma sequência de números inteiros **não–negativos**, devolvendo essa tabela e a quantidade de números armazenados (*dimensão*), para fora deste módulo.

2.b) Escreva um procedimento para, dada uma tabela do tipo *Vector*, *vec*, já ordenada por ordem **não–crescente** dos seus elementos, a respectiva dimensão, n, e um número inteiro não-negativo, *num*, inserir *num* **em ordem** na tabela dada e devolver os parâmetros actualizados.