

**→ Algoritmos de pesquisa**

1 - Construa um programa de dado um vector,  $C$ , de caracteres com dimensão ( $N \geq 1$ ) e uma carácter  $x$  determine a primeira ocorrência de  $x$  em  $C$ .

2 - Pesquisar um valor num array torna-se uma tarefa mais simples se o array estiver ordenado. O bloco de código abaixo implementa o algoritmo de pesquisa binária:

```

...
p:=1;
u:=N;
meio:= (p+u) div 2;
encontrou:= false;
while ((p<=u) and not encontrou ) do
  begin
    if ( V[meio]=x ) then
      encontrou:=true
    else
      begin
        if ( V[meio]>x ) then
          u:=meio-1
        else
          p:=meio+1;
          meio:=(p+u) div2
        end;
      end;
  end;
...

```

- Depois de o estudar com atenção, complete-o de forma a construir uma função que, recebendo um array de valores inteiros, a sua dimensão,  $N$ , e um valor a pesquisar,  $x$ , devolva como resultado a posição onde se encontra o valor  $x$ . Caso  $x$  não pertença ao array, a função deve devolver o valor zero.

Elabore um programa que use a função anterior.

## → Algoritmos de ordenação

**1** - Construa um programa que ordene um array de inteiros por ordem decrescente, usando o algoritmo de ordenação por selecção.

**2** - O bloco de programa abaixo implementa o algoritmo de ordenação "bubble sort" .

( Dado um array com N elementos, o algoritmo consiste em percorrer o array N-1 vezes, comparando os elementos em posições sucessivas e trocando-os sempre que encontra um par de elementos fora de ordem. Após i passagens, os últimos i elementos do vector já estão na posição correcta.)

...

```

for i:=N downto 2 do
  for j:=1 to i-1 do
    if (V[j] > V[j+1] ) then
      begin
        aux:=V[j];
        V[j]:=V[j+1];
        V[j+1]:=aux
      end;

```

...

**a)** Construa um programa que use os seguintes procedimentos:

- um procedimento que leia um array de inteiros,
- um procedimento que escreva um array de inteiros,
- um procedimento que ordene um array de inteiros usando o algoritmo acima.

**b)** O algoritmo anterior pode ser optimizado se pararmos de percorrer o array quando verificamos que numa passagem já não foi necessário fazer nenhuma troca. Tente alterar o algoritmo usando uma variável lógica, *troca*, que lhe permita terminar o ciclo mais externo após uma passagem pelo array em que não seja efectuada qualquer troca.