

# Programação / Programação I

## LEI/1, LTSI/1, LMAT/1

Hugo Pedro Proença

Universidade da Beira Interior  
Departamento de Informática

# Resumo

- A Linguagem C
  - Variáveis
    - Declaração de vetores
  - Instruções de iteração
    - for
    - while
    - do...while
  - Exercícios

# Vectorres

- Tal como visto anteriormente, um vector será uma variável que contém espaço para “n” elementos do mesmo tipo.
  - Inteiros, reais, caracteres,...
- É importante referir que todos os elementos ficam fisicamente armazenados em espaços de memória adjacente.
  - Acesso directo → Aumento eficiência.
  - Problemas de espaço.

# Vetores: Declaração

- Sintaxe:

**<tipo> <variável> [<dimensao1>] [<dimensao2>] ... ;**

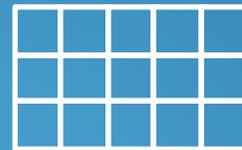
- Exemplos:

- **int x[10];**



Estrutura unidimensional:  
x[0], x[1], ..., x[n-1]

- **float y[5][3];**



Estrutura bidimensional:  
y[0][0], y[1][0], ..., y[c-1][0]

...  
y[0][l-1], y[1][0], ..., y[c-1][l-1]

# Noção de Bloco: Revisão

- **Bloco: agrupamento “lógico” de instruções**
  - Blocos condicionais
  - Blocos iterativos
  - Blocos iterativos dentro de ...

Instrução A ;

{ instrução 1 ;  
instrução 2 ;  
Instrução 3 ; }

Instrução B ;



Após a execução de “A”, cada instrução dentro do bloco {1,2,3} será executada 1 vez. A execução passará depois para “B”.

# Blocos Iterativos

- Agrupamento “lógico” de instruções que serão executadas mais que uma vez:

Instrução A ;

{ instrução 1 ;  
instrução 2 ;  
Instrução 3 ; }

Instrução B ;

Sequência de execução:

•A

•1  
•2  
•3  
•1  
•2  
•3  
•...

“n” vezes

•B

# Blocos Iterativos: Componentes

- **Inicialização**
  - Parte em que atribuem os valores iniciais das variáveis que controlarão o número de execuções do bloco.
- **Incremento**
  - Alteração efectuada às variáveis que controlam a execução do bloco em cada iteração.
- **Critério de Paragem**
  - Preposição que determina a saída ou a continuação da execução dentro do bloco iterativo. Quando é avaliada com o valor TRUE, a execução continua dentro do ciclo.

# Blocos Iterativos: for

- Permite a execução de um bloco [0,n] vezes.

- Sintaxe:

```
for(<inicialização>; <critério paragem>; <incremento>){  
  
}
```

- Tipicamente utilizado em situações onde o programador sabe à partida o número de execuções de um ciclo.
  - Exemplo: percorrer um vector,...

# Blocos Iterativos: for, exemplos

- Escrita de uma mensagem 100 vezes.
- Execução múltipla de uma instrução de printf

```
int ic;
```

Variável que vai controlar o número de vezes que o bloco iterativo é executado

```
...
```

```
for (ic=0; ic<100; ic=ic+1){  
    printf("Olá mundo\n");  
}
```

# Blocos Iterativos: for, exemplos

- Contagem dos elementos positivos de um vector (V) de 500 posições

```
int ic, conta=0;  
...  
for (ic=0; ic<500; ic=ic+1){  
    if(V[ic]>0)  
        conta=conta+1;  
}
```

Bloco condicional dentro de bloco iterativo

Variável "ic" vai controlar o número de vezes que o bloco iterativo é executado.

# Blocos Iterativos: while

- Similarmente ao “for”, permite a execução de um bloco [0,n] vezes.

- Sintaxe:

```
<inicialização>;  
while(<critério paragem>){  
    ...  
    <incremento>;  
}
```

- Tipicamente utilizado em situações onde o programador não sabe à partida o número de execuções de um ciclo.
  - Exemplo: pesquisar por um elemento num conjunto,...

# Blocos Iterativos: while, exemplos

- Pesquisa num vector por um elemento par:

```
int ic;  
...  
ic=0;  
while (V[ic]%2!=0){  
    printf("O elemento da posição %d não é par\n",ic);  
    ic=ic+1;  
}
```



Potencialmente  
problemático!  
Porquê?

# Blocos Iterativos: do ... while

- Contrariamente ao “for” e “while”, permite a execução de um bloco [1,n] vezes.
  - Critério de paragem é avaliado apenas no fim do bloco.
  - Garante sempre uma execução.

- Sintaxe:

```
<inicialização>;  
do{  
    <incremento>;  
}  
while(<critério paragem>;);
```

- Típicamente utilizado em situações onde o programador quer garantir a execução de um bloco pelo menos 1 vez.
  - Exemplo: leitura de valores do utilizador.

# Blocos Iterativos: do... while, exemplos

- **Leitura de um valor contido num intervalo ([0,20]):**

```
int valor;  
...  
do{  
    printf("Sr. Utilizador. Introduza uma nota (0,20)\n");  
    scanf("%d",&valor);  
}while ((valor<0)||valor>20);
```



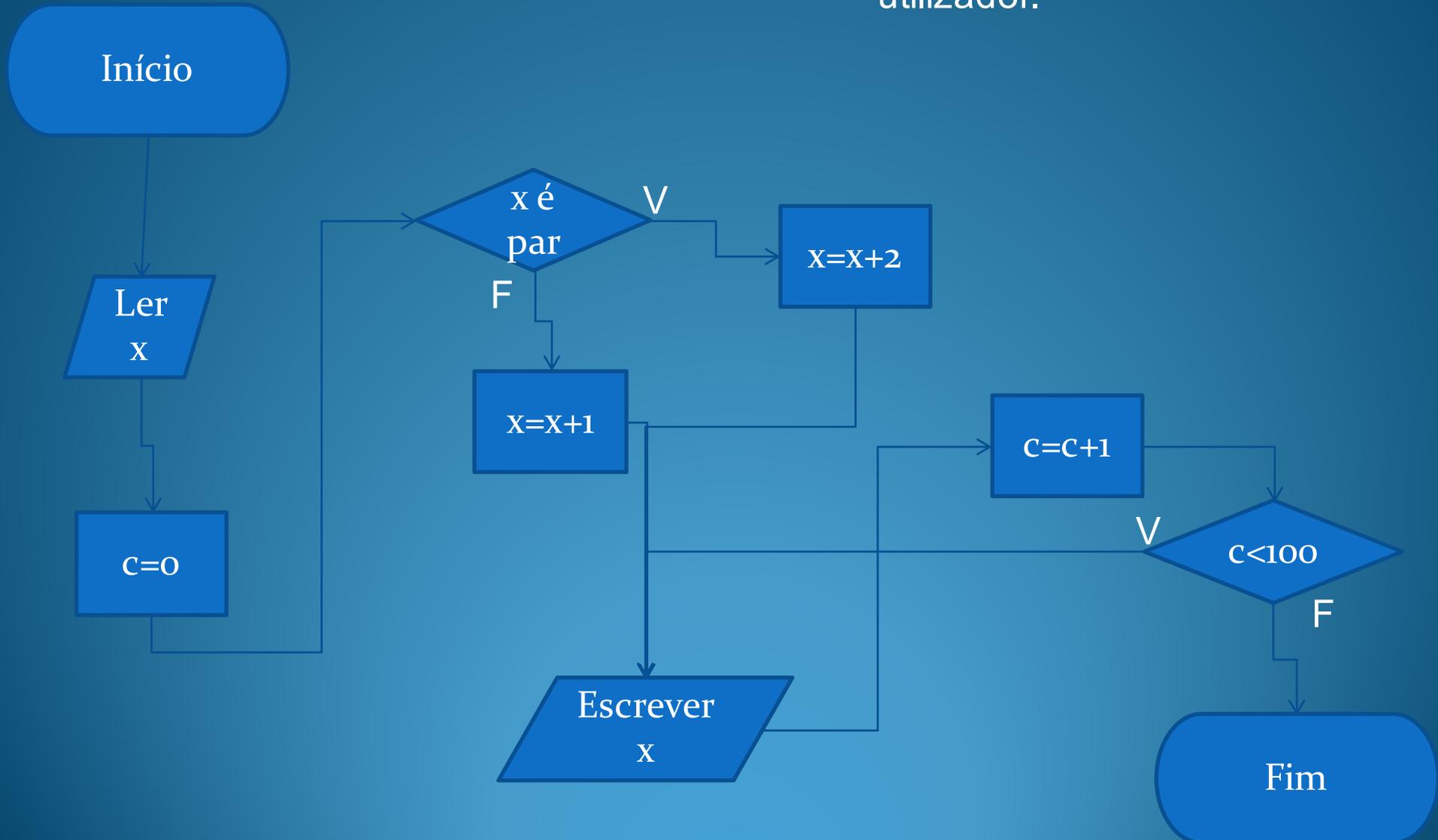
Onde está o incremento?  
E a inicialização ?  
Porquê ?

# Exercícios

- **Efectue a passagem de cada um dos fluxogramas anteriormente apresentados nas aulas teóricas para código em linguagem C.**

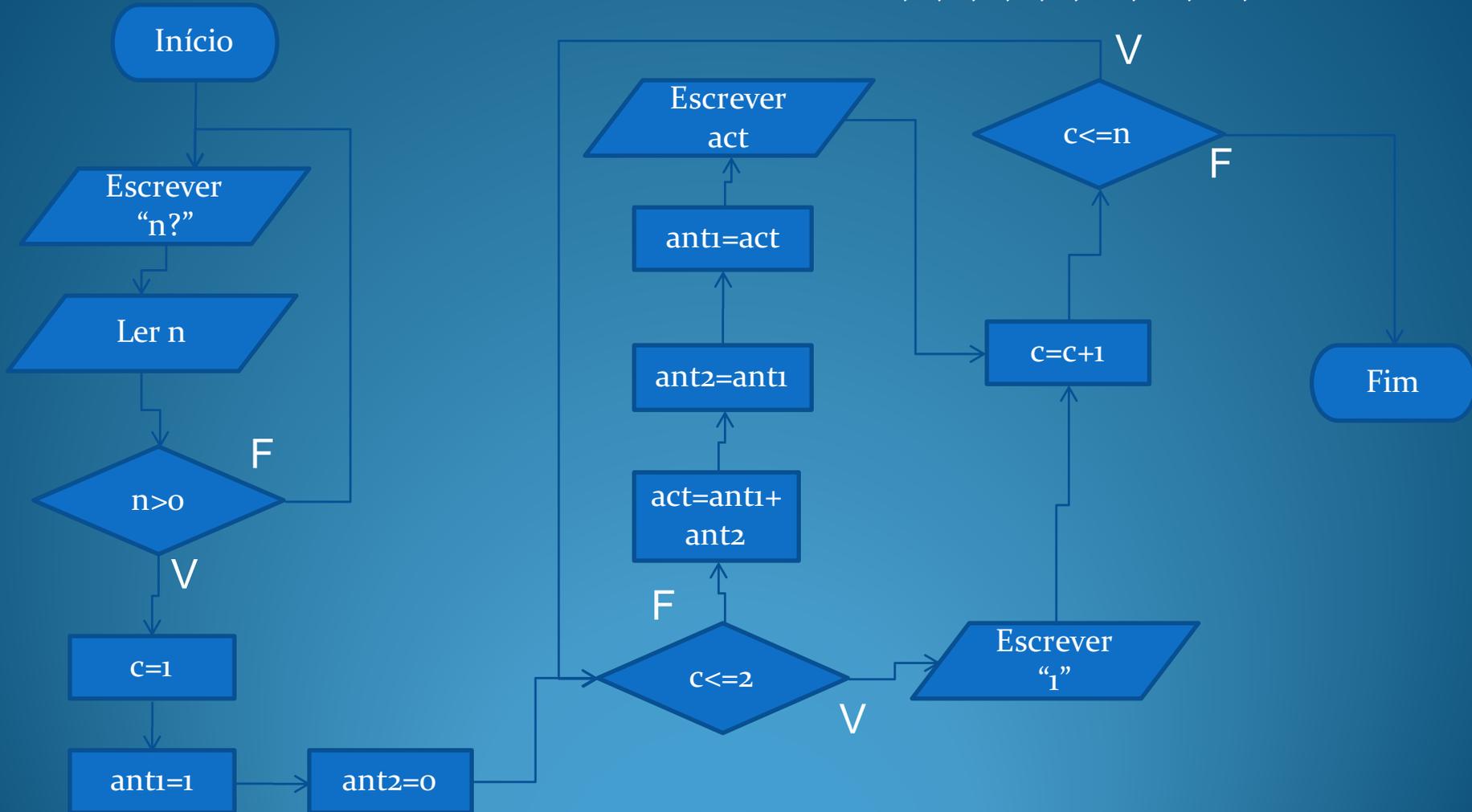
# Exercícios

Escrita do 100 números pares imediatamente superiores a um valor recebido do utilizador.



# Exercícios

Escrita dos primeiros "n"  
(valor recebido do utilizador)  
números de fibonnacci  
1,1,2,3,5,8,13,21,34,...



# Exercícios

Leitura de um vetor de 100 elementos reais. Escrita do maior e menor valores do vector.

