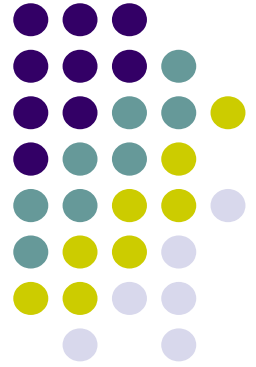




# Algoritmia e Programação

## Linguagens de Programação Introdução à Linguagem C

José Tavares – [jrt@isep.ipp.pt](mailto:jrt@isep.ipp.pt)  
ISEP, 2023/2024



## Paradigmas de programação

- Programação imperativa
  - C, Pascal, ...
- Programação funcional
  - Haskell, Lisp, ...
- Programação em lógica
  - Prolog,...
- Programação Orientada por Objectos
  - C++, Java, C#, ...
- Etc.



# Programação imperativa



- **Princípios**

- Reserva de memória feita através da declaração de variáveis
- Modelo Centralizado de Dados
  - Conjunto de todas as variáveis do programa não é mais do que o repositório central de dados que é manipulado pelas funções e operações usadas no programa
- Princípio fundamental
  - Nenhuma variável, constante ou função pode ser utilizada sem ter sido previamente declarada

APROG 2023/2024 © DEI/ISEP

# Linguagem de Programação



- **Léxico**

- Conjunto de palavras de uma linguagem

- **Sintaxe**

- Conjunto de regras de construção das frases de um texto
  - Gramática
- Frases são designadas por instruções e o texto é designado por programa

- **Semântica**

- Significado de cada instrução dum programa ou mesmo dum programa completo

APROG 2023/2024 © DEI/ISEP

# 1º Programa em C



```
/* Primeira experiência em linguagem C...*/
```

```
#include <stdio.h>
```

```
void main()  
{  
    printf("Ola Mundo!\n");  
}
```

APROG 2023/2024 © DEI/ISEP

## History of C



- Evolved from two previous languages – BCPL, B
- BCPL (Basic Combined programming language) used for writing OS & compilers
- B used for creating early versions of UNIX OS
- Both are “typeless” language
- C language evolved from B (Dennis Ritchie – Bell labs)
- BCPL and B are "type less" languages whereas C provides a variety of data types.
- In 1972 Dennis Ritchie at Bell Labs writes C and in 1978 the publication of The C Programming Language by Kernighan & Ritchie caused a revolution in the computing world.

APROG 2023/2024 © DEI/ISEP

# Nomes / Identificadores



- Há 3 classes de identificadores
  - Identificadores definidos pelo utilizador
    - Por exemplo, nomes de variáveis
  - Identificadores padronizados
    - Por exemplo, nomes de funções
  - Palavras reservadas
    - Não podem ser redefinidos

APROG 2023/2024 © DEI/ISEP

## Palavras reservadas C/C++



- Não podem ser usadas a não ser para o fim com que foram definidas
  - Não podem ser usadas, por exemplo, como nome de variáveis

Palavras chave do C++					
asm	auto	break	case	catch	char
class	const	continue	default	delete	do
double	else	enum	extern	float	for
friend	goto	if	inline	int	long
new	operator	private	protected	public	register
return	short	signed	sizeof	static	struct
switch	template	this	throw	try	typedef
union	unsigned	virtual	void	volatile	while

APROG 2023/2024 © DEI/ISEP



# Entidades e expressões

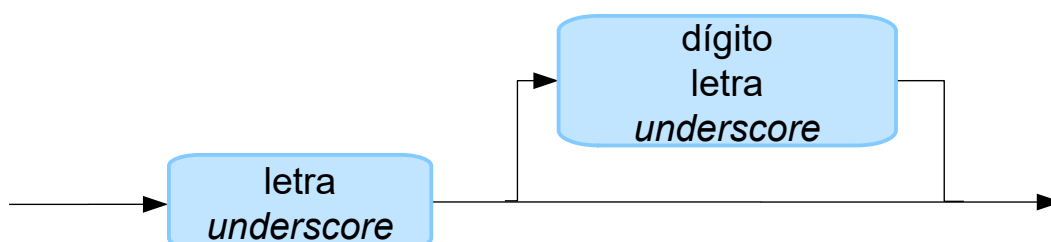
- Entidades
  - Identificadores ou nomes
  - Operadores
  - Literais
- Expressões
  - É uma construção sintática que tem um valor
  - É uma combinação de entidades que representa o cálculo de um valor
- Tipos de expressões
  - Expressão numérica
  - Expressão lógica
  - Expressão simbólica

APROG 2023/2024 © DEI/ISEP



## Nomes / Identificadores em C

- Regras para definição de nomes/identificadores
  - Diferenciação entre maiúsculas e minúsculas
  - Tamanho variável
  - Nomes válidos começam por letra ou \_
    - Válidos: `num`, `num5`, `num_`
    - Inválidos: `5num`, `num5%`, `const`
- Diagrama de sintaxe



APROG 2023/2024 © DEI/ISEP

# Operadores em C



- Operador atribuição
  - = (escrita de dados em memória)
- Operadores aritméticos
  - +, -, \*, /, %
- Operadores lógicos
  - &&, ||, !
- Operadores relacionais
  - <, <=, >, >=, !=, ==
- Operador “Incrementar”
  - ++, +=
- Operador “Decrementar”
  - --, -=

APROG 2023/2024 © DEI/ISEP

# Expressões em C



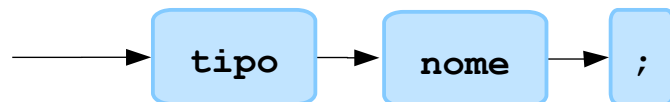
- Exemplos
  - 2 (literal inteiro)
  - num (nome de variável)
  - 5 + num (adição de literal e valor de variável)
  - sqrt(12) (chamada a função com argumento literal)
  - a && b || c (expressão com valor lógico associado)
  - !a (nega valor lógico de variável)
  - num++ (incrementa uma unidade ao valor da variável)

APROG 2023/2024 © DEI/ISEP



# Variáveis em C

- Objetos que armazenam informação (usualmente localizadas na memória)
- Têm nome e valor (conteúdo)
  - O valor de uma variável não é constante
- Podem mudar de valor através de uma instrução de atribuição
- Declaração de variáveis
  - Diagrama de sintaxe



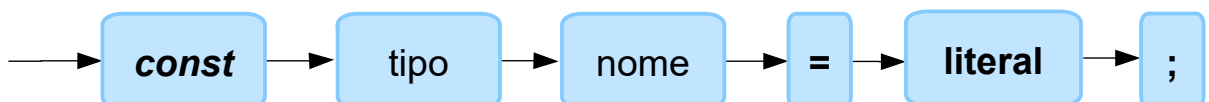
```
void main() {  
    int num;  
    num = 10;  
    ....  
}
```

APROG 2023/2024 © DEI/ISEP

# Constantes em C



- (Também são) Objetos que guardam informação
- (Também) Têm nome e valor (conteúdo)
- Uma constante tem um valor fixo ou imutável
  - O valor de uma variável não é constante...
- Declaração de constantes
  - Diagrama de sintaxe



```
void main() {  
    const int i = 10;  
}
```

APROG 2023/2024 © DEI/ISEP



# Tipos de dados básicos em C

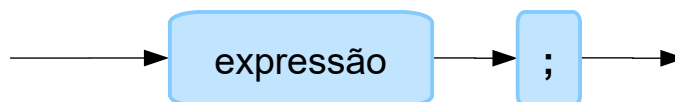
- int
  - 4 byte (2 byte em “antigas” máquinas/compiladores de 16bit)
  - Formato de Entrada/saída “%d”
- char
  - 1 byte
  - Formato de Entrada/saída “%c”
- float
  - 4 byte
  - Formato de Entrada/saída “%f” ou “%e” em notação científica
- double
  - 8 byte
  - Formato de Entrada/saída “%f” ou “%e” em notação científica

APROG 2023/2024 © DEI/ISEP



## Instruções em C

- Diagrama de sintaxe



- Exemplos

- `printf("ola mundo!\n");`
- `x = 2 * x + 6;`
- `++cont;`

APROG 2023/2024 © DEI/ISEP





# Tipos de instruções em C

- É uma das seguintes entidades:
  - Uma instrução-bloco
  - Uma instrução-de-atribuição
  - Uma instrução-if
  - Uma instrução-while
  - Uma instrução-for
  - Uma instrução-switch
  - Uma instrução-break
  - Uma instrução-continue
  - Uma instrução-return

APROG 2023/2024 © DEI/ISEP

## Programação em C/C++



- Um programa em C/C++ é constituído por:
  - Uma ou mais funções,
  - É obrigatório existir a função ***main()***
  - Cada função definida por
    - Tipo de dados que a função devolve
    - Nome da função
    - Parâmetros formais
    - Corpo (definido entre "{...}") com
      - Instruções
        - Declarações
        - Decisões
        - Repetições
      - Comentários
  - Inclusão de ficheiros ***header*** (bibliotecas)
    - Protótipos de funções, macros, instruções, declarações de tipos de dados

```
#include <stdio.h>

void main() {
    //Mensagem de boas vindas
    printf("Olá Mundo\n");
}
```

APROG 2023/2024 © DEI/ISEP

# Introdução ao C



- Para escrever o código-fonte, necessitamos de um IDE, ou seja, um ambiente de desenvolvimento integrado
  - DevC++
  - Eclipse
  - Visual Studio (C/C++)
  - NetBeans
  - etc, etc, ...
- Depois de escrever o código é só compilar, se não existirem erros gera-se o executável, que é o resultado do código...

APROG 2023/2024 © DEI/ISEP

# Introdução ao C



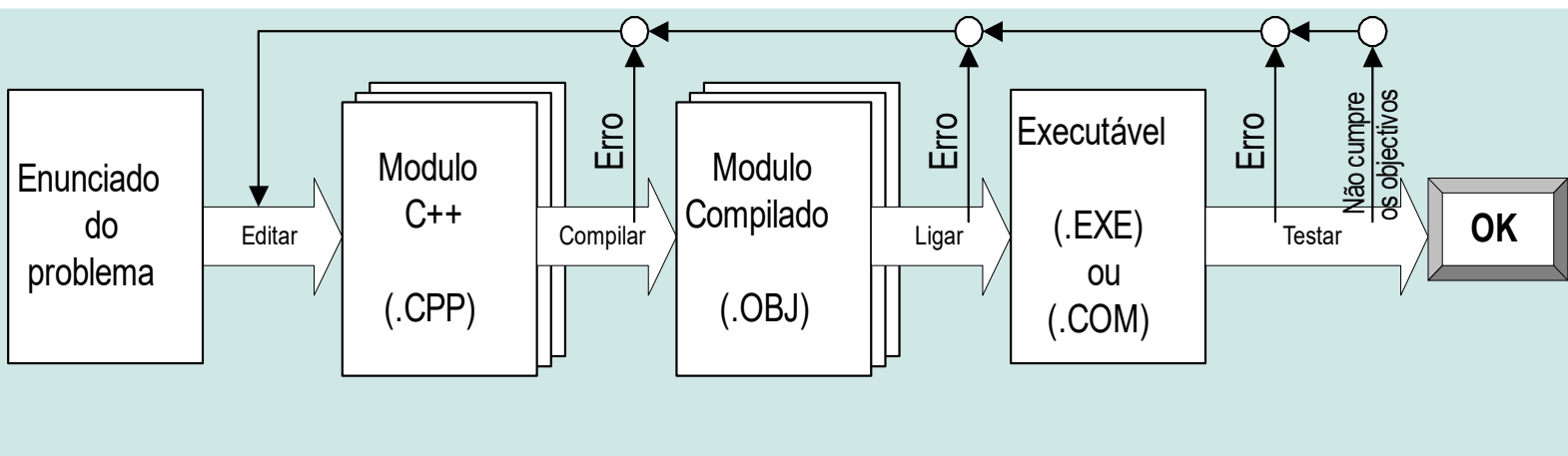
- Ambiente de desenvolvimento
  - Editar código (texto)
    - Ex.: MS Visual C++
  - Compilação
    - Verificar se a sintaxe das instruções está correcta
    - Efetuada pelo compilador – cria ficheiro objecto
    - Compilação pode detetar erros ou situações “suspeitas” (assinaladas como *warning*)
  - “Linkar” / “build” / ligar módulos
    - O “*linker*” cria o ficheiro executável (programa) a partir do ficheiro objeto, obtido através da compilação, e através das bibliotecas com funções do próprio C
  - “Debug” / depurar erros

APROG 2023/2024 © DEI/ISEP

# Introdução ao C



- Fases da geração de um programa



APROG 2023/2024 © DEI/ISEP

## Saída de Dados



- Função de escrita no ecrã: **printf()**
  - Não faz parte da linguagem C, pertence à sua extensa biblioteca de funções
    - Para ter acesso às funções existentes nas Bibliotecas do C, é necessário incluir a sua definição no programa, através da diretiva que indica ao compilador que deverá adicionar ao processo de compilação o ficheiro chamado **stdio.h**
  - A função **printf** escreve todos os caracteres que lhe sejam enviados entre aspas
  - A mudança de linha é indicada pelo símbolo **"\n"**

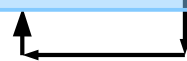
APROG 2023/2024 © DEI/ISEP

# Leitura e Escrita



- Para escrita de variáveis, usar a função **printf** utilizando os tipos de dados adequados (%c, %d, %f)
- Sempre que se pretende escrever o valor guardado numa variável, deve substituir-se o valor dessa variável pelo formato de escrita, que naquele preciso local representará a variável a escrever
- Exemplo com variável inteira

```
printf("O valor de num = %d \n", num);
```



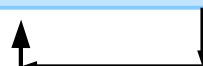
APROG 2023/2024 © DEI/ISEP

# Leitura e Escrita



- Da mesma forma que existe a função **printf** para escrita de valores, existe uma função correspondente para a leitura de valores – a função **scanf**
- A *string* enviada na função **scanf**, deve conter apenas os caracteres indicadores de formato
- Exemplo leitura de um valor para uma variável inteira

```
scanf("%d", &num);
```



APROG 2023/2024 © DEI/ISEP

# Introdução ao C



- Elabore um programa que calcule a média de dois valores indicados pelo utilizador

```
Algoritmo média
ED: num1, num2, media real      `definição das estruturas de dados
Início
    escrever("Introduza os valores")  `pedido dados de input
    ler(num1, num2)                  `leitura dados de entrada
    media ← (num1+num2)/2             `cálculo da média
    escrever( "média = ", media)      `output do resultado
Fim
```

APROG 2023/2024 © DEI/ISEP

# Introdução ao C



- Elabore um programa que calcule a média de dois valores indicados pelo utilizador

```
#include <stdio.h>
void main()
{
    float num1, num2, media;

    printf("Introduza os valores\n");
    scanf("%f%f", &num1, &num2);

    media = (num1+num2)/2;

    printf( "media = %f\n", media);
}
```

APROG 2023/2024 © DEI/ISEP

# Sequência – Exercício 1



- Descreva um algoritmo capaz de converter um valor em metros para centímetros

```
Algoritmo conversão
ED: m, cm real
Início
    escrever("Introduza o valor em metros:")
    ler(m)
    cm ← m * 100
    escrever(m, "m = ", cm, " cm")
Fim
```

APROG 2023/2024 © DEI/ISEP

# Sequência – Exercício 1



- Descreva um algoritmo capaz de converter um valor em metros para centímetros

```
#include <stdio.h>

void main()
{
    float m, cm;

    printf("Introduza o valor em metros:");
    scanf("%f", & m);

    cm = m * 100;

    printf("%f m = %f cm", m, cm);
}
```

APROG 2023/2024 © DEI/ISEP

# Instruções

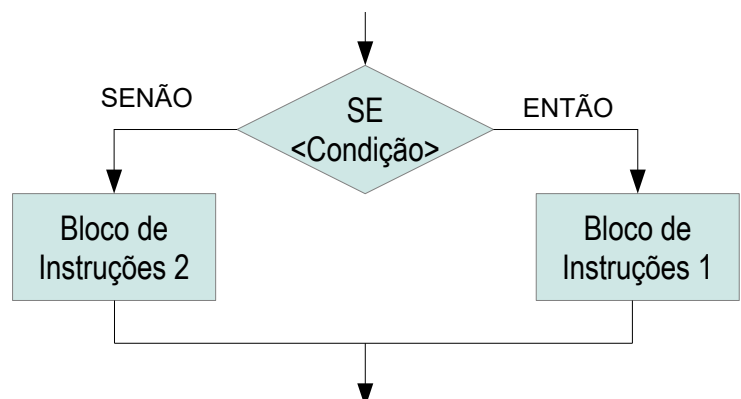
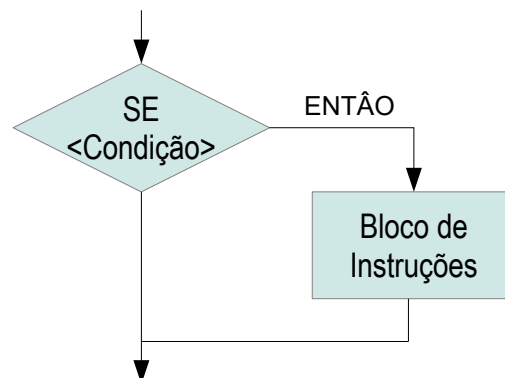


- Instruções compostas incluem sub-instruções
  - Bloco {...}
  - Estruturas de controlo
    - Decisão
      - `if`
      - `if - else`
      - `switch`
    - Repetição
      - `while`
      - `do - while`
      - `for`

APROG 2023/2024 © DEI/ISEP

## Algoritmia - Decisão

- Se <Condição> Então  
    <Bloco de instruções>  
FimSe
- Se <Condição> Então  
    <Bloco de instruções 1>  
Senão  
    <Bloco de instruções 2>  
FimSe



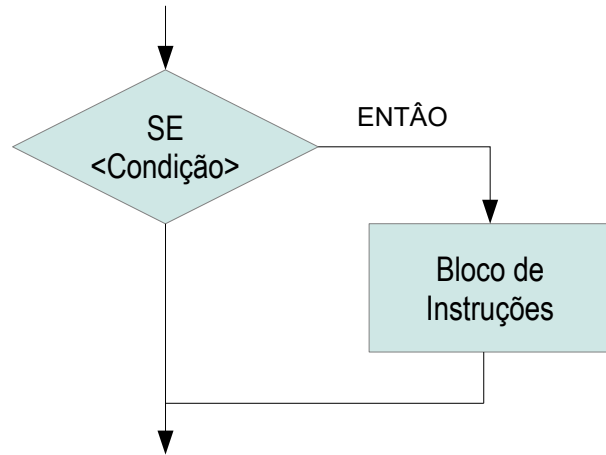
APROG 2023/2024 © DEI/ISEP

# Instruções de Decisão



```
if (condição)  
    instrução ;
```

```
if (condição)  
{  
    instruções ;  
}
```

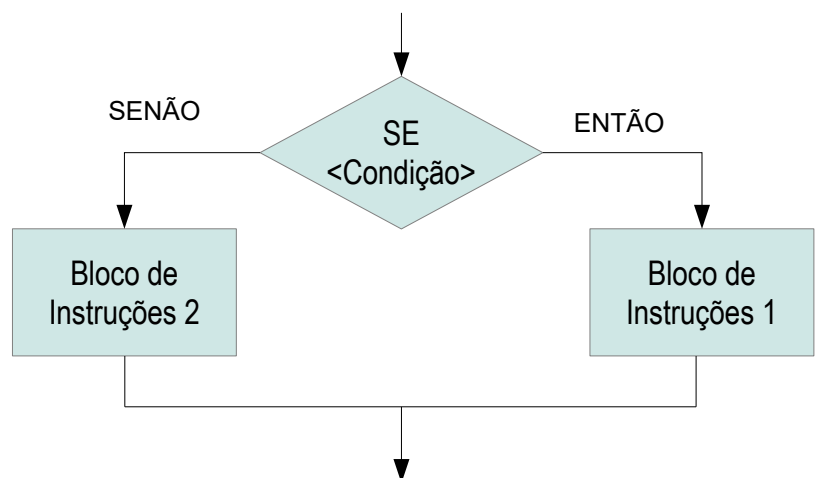


APROG 2023/2024 © DEI/ISEP

# Algoritmia - Decisão



```
if (condição)  
{  
    instruções1 ;  
}  
else  
{  
    instruções2 ;  
}
```



APROG 2023/2024 © DEI/ISEP



# Decisão – Exercício 1



- Descreva um algoritmo que permita verificar se um número inteiro introduzido pelo utilizador é par ou impar

```
Algoritmo par_impar
ED: num inteiro
Início
    Escrever("Introduza o número:")
    Ler(num)
    Se num é par ` verificar resto da divisão por zero
        Então Escrever("O número é par")
    Senão Escrever("O número é impar")
    FimSe
Fim
```

APROG 2023/2024 © DEI/ISEP

# Decisão – Exercício 1



- Elabore um programa que permita verificar se um número inteiro introduzido pelo utilizador é par ou ímpar

```
#include <stdio.h>

void main()
{
    int num;

    printf("Insira um valor inteiro.\n");
    scanf("%d", & num);
    if (num % 2 == 0)
        printf("O numero %d e par\n", num);
    else
        printf ("O numero %d nao e par\n", num);
}
```

APROG 2023/2024 © DEI/ISEP

## Decisão – Exercício 2



- Descreva um algoritmo que determine o maior de dois números inteiros introduzidos pelo utilizador. Assuma que os dois números são diferentes.

```
Algoritmo maior_2
ED: num1, num2, max inteiro
Início
    Escrever("Introduza os números:")
    Ler(num1, num2)
    Se num1 > num2 Então
        max ← num1
    Senão
        max ← num2
    FimSe
    Escrever(max)
Fim
```

APROG 2023/2024 © DEI/ISEP

## Decisão – Exercício 2



- Codifique em Linguagem C o algoritmo que determina o maior de dois números inteiros introduzidos pelo utilizador. Assuma que os dois números são diferentes.

```
#include <stdio.h>

void main() {
    int num1, num2, max;

    printf("Escreva dois valores inteiros.\n");
    scanf("%d%d", &num1, &num2);

    if(num1 > num2)
        max = num1;
    else
        max = num2;

    printf("O maior valor inserido foi o %d\n", max);
}
```

APROG 2023/2024 © DEI/ISEP



## Decisão – Exercício 3

- Descreva um algoritmo que lê três números inteiros e determina qual é o maior. Assuma que os três números são diferentes.

```
Algoritmo maior_3
ED: A, B, C, MAX inteiro
Início
    Escrever("Introduza os números:")
    Ler(A, B, C)
    Se A > B Então
        Se A > C Então
            MAX ← A
        Senão MAX ← C
    FimSe
    Senão
        Se B > C Então
            MAX ← B
        Senão MAX ← C
    FimSe
    FimSe
    Escrever(MAX)
Fim
```

APROG 2023/2024 © DEI/ISEP

## Decisão – Exercício 3

- Escreva em linguagem C o programa que lê três números inteiros e determina qual é o maior. Assuma que os três números são diferentes.

```
#include <stdio.h>

void main(){
    int num1, num2, num3, max;

    printf("Escreva 3 numeros.\n");
    scanf("%d%d%d", &num1, &num2, &num3);
    if(num1 > num2)
    {
        if( num1 > num3) max = num1;
        else max = num3;
    }
    else
    {
        if( num2 > num3) max = num2;
        else max = num3;
    }
    printf("O maior valor e o %d\n", max);
}
```



APROG 2023/2024 © DEI/ISEP

# Introdução ao C



- Seleção

- Muitas vezes a escolha não está dependente apenas de uma condição ser verdadeira ou falsa, está dependente de um conjunto de expressões ou valores
- Tomada de decisões quando o número de possibilidades é elevado (em geral maior do que 2...), de forma a reduzir a complexidade de vários *if-else* consecutivos encadeados

APROG 2023/2024 © DEI/ISEP

# Introdução ao C



- Seleção – Sintaxe

```
switch (expressão)
{
    case constante1: instruções1;
    case constante2: instruções2;
    ...
    case constanteN: instruçõesN;
    [ default : instruções; ]
}
```

APROG 2023/2024 © DEI/ISEP

# Introdução ao C



- Seleção – funcionamento
  - Se o valor da expressão for igual a alguma das constantes que seguem os vários **case** então são executadas as instruções correspondentes
  - Se o valor da expressão não for igual a nenhuma das constantes apresentadas pelos **case** então são executadas as instruções que seguem o **default**
  - O **default** é opcional
    - Caso não exista **default** e a expressão não for igual a nenhum **case**, nada é executado, terminando o **switch**

APROG 2023/2024 © DEI/ISEP

## Seleção – Exercício 1



- Escreva um programa que indique o estado civil correspondente a um carácter introduzido em letras maiúsculas
  - Ler carácter
  - Caso carácter lido seja
    - C escrever “Casado”
    - S escrever “Solteiro”
    - D escrever “Divorciado”
    - V escrever “Viúvo”
    - Outros casos escrever “estado civil incorrecto”

APROG 2023/2024 © DEI/ISEP

# Seleção – Exercício 1



```
Algoritmo estado_civil
ED: estado caracter
Início
    Escrever("Indique o seu estado civil:")
    Ler(estado)
    Selecionar estado
        Caso 'C'
            Escrever("Casado")
        Caso 'S'
            Escrever ("Solteiro")
        Caso 'D'
            Escrever("Divorciado")
        Caso 'V'
            Escrever ("Viuvo")
        Outros Casos
            Escrever ("Estado incorrecto")
    FimSelecionar
Fim
```

APROG 2023/2024 © DEI/ISEP

# Seleção – Exercício 1



```
#include <stdio.h>

void main(){
    char estado;

    printf("Qual o estado civil.\n");
    scanf("%c", &estado);

    switch(estado){
        case 'C': printf("Casado\n"); break;
        case 'S': printf("Solteiro\n"); break;
        case 'D': printf("Divorciado\n"); break;
        case 'V': printf("Viuvo\n"); break;
        default : printf("Estado incorrecto\n");
    }
}
```

APROG 2023/2024 © DEI/ISEP

# Repetição – Enquanto ... Fazer ...

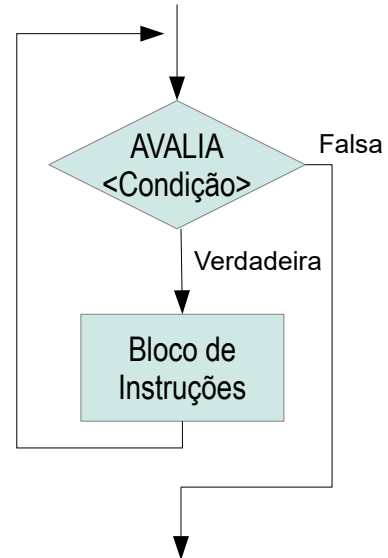


## Pseudo-código

```
ENQUANTO <Condição> FAZER  
    <Bloco de Instruções>  
FIMENQUANTO
```

```
while (<Condição>)  
{  
    <Bloco de Instruções>  
}
```

## Representação Gráfica



APROG 2023/2024 © DEI/ISEP

## Repetição – Exercício 1



- Descreva um algoritmo capaz de ler e escrever uma sequência de números inteiros terminada por zero

```
Algoritmo le_sequencia  
ED: NUM inteiro  
Início  
    Escrever("Introduza um número:")  
    Ler(NUM)  
    Enquanto NUM <> 0 Fazer  
        Escrever(NUM)  
        Escrever("Introduza próximo número:")  
        Ler(NUM)  
    FimEnquanto  
Fim
```

APROG 2023/2024 © DEI/ISEP

# Repetição – Exercício 1



- Escreva um programa capaz de ler e escrever uma sequência de números inteiros terminada por zero

```
#include <stdio.h>

void main() {
    int num;

    printf("Introduza um numero.\n");
    scanf("%d", &num);

    while (num!=0) {
        printf("Inseriu o valor: %d", num);
        printf("Introduza proximo numero. \n");
        scanf("%d", &num);
    }
}
```

APROG 2023/2024 © DEI/ISEP

## Repetição – Fazer... Enquanto...

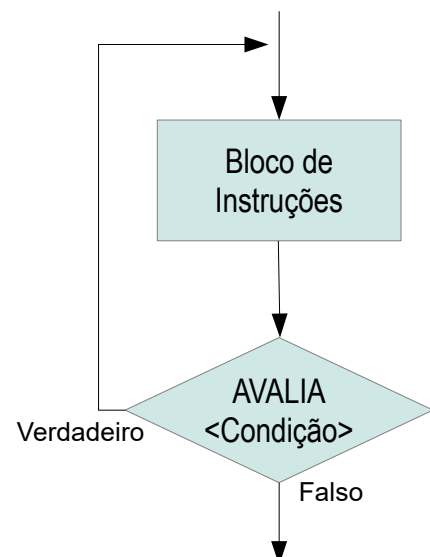


### Pseudo-código

REPETIR  
    <Bloco de Instruções>  
ENQUANTO <Condição>

```
do
{
    <Bloco de Instruções>
} while (<Condição>)
```

### Representação Gráfica



APROG 2023/2024 © DEI/ISEP



# Repetição – Exercício 2



- Descreva um algoritmo capaz de ler e escrever uma sequência de números inteiros terminada por zero

```
Algoritmo le_sequencia2
ED: NUM inteiro
Início
    Repetir
        Escrever("Introduza um número:")
        Ler(NUM)
        Se NUM <> 0 Então
            Escrever(NUM)
        FimSe
    Enquanto NUM <> 0
Fim
```

APROG 2023/2024 © DEI/ISEP

# Repetição – Exercício 2



- Escreva um programa capaz de ler e escrever uma sequência de números inteiros terminada por zero

```
#include <stdio.h>

void main() {
    int num;

    do {
        printf("Introduza um numero:\n");
        scanf("%d", &num);

        if (num != 0)
            printf("Numero inserido: %d\n", num);
    } while(num != 0);
}
```

APROG 2023/2024 © DEI/ISEP

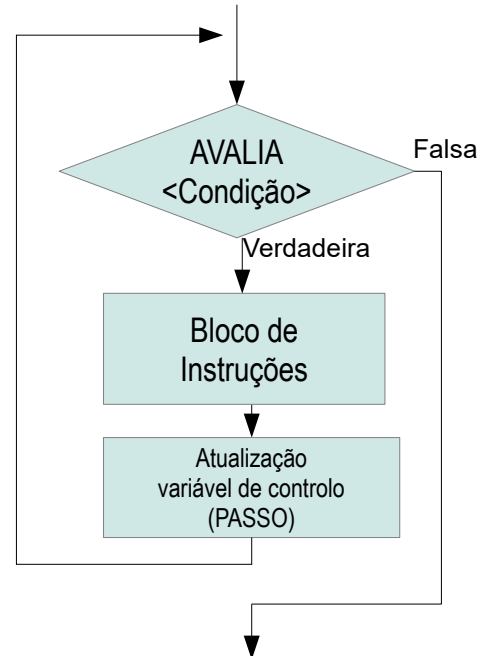
# Repetição – Para ... Fazer ...



## Pseudo-código

```
PARA <Variável de controlo> = <Valor1> ATÉ <Valor2> PASSO <N> FAZER  
    <Bloco de Instruções>  
FIMPARA
```

## Representação Gráfica



APROG 2023/2024 © DEI/ISEP

# Repetição – Para ... Fazer ...



```
PARA <Variável de controlo>=<Valor1> ATÉ <Valor2> PASSO <N> FAZER  
    <Bloco de Instruções>  
FIMPARA
```

```
for(inicializações; condição; passo)  
{  
    <Bloco de Instruções>  
}
```

APROG 2023/2024 © DEI/ISEP

# Repetição – Exercício 3



- Descreva um algoritmo capaz de ler e mostrar uma sequência de 10 números reais

```
Algoritmo sequencia3
ED: NUM real
    I inteiro
Início
    Para I ← 1 Até 10 Passo 1 Fazer
        Escrever("Introduza um número:")
        Ler(NUM)
        Escrever(NUM)
    FimPara
Fim
```

APROG 2023/2024 © DEI/ISEP

# Repetição – Exercício 3



- Escreva um programa capaz de ler e mostrar uma sequência de 10 números reais

```
#include <stdio.h>

void main(){
    int num, i;

    for(i = 0; i < 10; i++){
        printf("Introduza um numero:\n");
        scanf("%d", &num);
        printf("O numero inserido foi: %d", num);
    }
}
```

APROG 2023/2024 © DEI/ISEP



# Tipos de Dados Simples

- Numéricos
  - char, int, float, double
- Apontadores
  - Definidos pelo utilizador
  - Requer a utilização do símbolo \*
- Enumerados
  - Definidos pelo utilizador
  - Requer a utilização da palavra reservada “enum”

APROG 2023/2024 © DEI/ISEP



## Tipos de Dados char

- char
  - Armazena um valor inteiro de 1 byte
  - Guardar um carácter ou um número inteiro
  - Formato de leitura e escrita: %c
  - Exemplos

```
char c;  
c = 'A';    /* instruções equivalentes */  
c = 65;     /* 65 é o código ASCII do carácter A */  
  
char a, b;  
a = 5;      /* a contém o código ascii 5 */  
b = '5';    /* b contém o código ASCII do caracter 5 (este  
             corresponde ao código ascii 53)*/
```

APROG 2023/2024 © DEI/ISEP

# Exemplo



- Programa que lê e escreve um carácter

```
#include <stdio.h>
void main()
{
    char c;
    printf("Escreva um caracter\n");
    scanf("%c", &c);
    printf("O caracter inserido foi: %c\n", c);
}
```



- Programa que lê um carácter e escreve o seu código ASCII

```
#include <stdio.h>
void main()
{
    char c;
    printf("Escreva um caracter\n");
    scanf("%c", &c);
    printf("O caracter inserido foi: %d\n", c);
}
```



APROG 2023/2024 © DEI/ISEP

# Instruções



- Instruções simples não incluem sub-instruções
- Exemplos
  - Instrução de atribuição
    - `y = 5;`
  - Instrução de Saída
    - `printf("Algoritmia e Programacao");`
  - Instrução de Entrada
    - `scanf("%d", &num);`

APROG 2023/2024 © DEI/ISEP

# Tipos de Dados Reais



- Armazenar valores numéricos com parte fracionaria

- float
  - Números reais de precisão simples
  - 4 bytes
  - Formato de leitura e escrita: %f ou %e
- double
  - Números reais de dupla precisão
  - 8 bytes
  - Formato de leitura e escrita: %f ou %e
  - Exemplos

```
float num1, num2, num3;  
num1 = 0.99;  
num2 = 34;  
num3 = 6.2E+5; /* notação científica, representa 6.2×10^5
```

APROG 2023/2024 © DEI/ISEP

## Dados Numéricos – Ex: Máquina/Compilador 32bit



```
#include <stdio.h>  
#include <limits.h>  
#include <float.h>  
  
int main()  
{  
    printf("int\n");  
    printf("%d\n", INT_MIN);  
    printf("%d\n", INT_MAX);  
  
    printf("\nfloat\n");  
    printf("%e\n", FLT_MIN);  
    printf("%e\n", FLT_MAX);  
  
    printf("\ndouble\n");  
    printf("%le\n", DBL_MIN);  
    printf("%le\n", DBL_MAX);  
}
```

```
int  
-2147483648  
2147483647  
  
float  
1.175494e-038  
3.402823e+038  
  
double  
2.225074e-308  
1.797693e+308  
  
Process returned 14 (0xE)  
execution time : 0.042 s  
Press any key to continue.
```

APROG 2023/2024 © DEI/ISEP

# Qualificadores



- De comprimento
  - short
    - `short int i;`
  - long
    - `long int j;`
  - O formato de leitura de variáveis inteiras do tipo *short* e *long* nas funções *printf* e *scanf* deve ser precedida dos prefixos *h* (short) e *l* (long)

APROG 2023/2024 © DEI/ISEP

# Qualificadores



- De sinal
  - Unsigned
    - caso deseje que a variável contenha apenas valores positivos)
    - `unsigned int j;`
  - signed
    - `signed int i;`
    - por defeito todos os inteiros são criados com sinal...

APROG 2023/2024 © DEI/ISEP



Tipo de Variável	Nr de Bytes	Valor mínimo	Valor máximo
int	4	-2 147 483 648	2 147 483 647
short int	2	-32 768	32 767
long int	4	-2 147 483 648	2 147 483 647
unsigned int	4	0	4 294 967 295
unsigned short int	2	0	65 535
unsigned long int	4 (8 em 64bit)	0	4 294 967 295 ( $2^{64}$ )

- Para máquinas/compiladores de 32 bits – entre parêntesis máquinas/compiladores de 64 bits.
- `sizeof(<tipo de dados>)`  
é um operador que retorna o número de bytes efetivos que um dado tipo usa.

APROG 2023/2024 © DEI/ISEP

## Tipos de dados enumerados



- Definidos através do operador “**enum**”
- Têm uma ordem...
- Exemplos
  - `enum cor {red, blue, green, yellow};`
  - `enum dia {domingo, seg_feira, ter_feira, qua_feira, qui_feira, sex_feira, sábado};`
- A linguagem C, ao contrário da maioria das linguagens (como, por exemplo o C++), não tem explicitamente o tipo de dados *boolean*
  - Em C, o valor falso de uma expressão lógica é o 0 e o valor verdadeira é o 1
  - Através de dados enumerados é possível criar o tipo de dados boolean
    - `enum boolean {FALSE, TRUE};`

APROG 2023/2024 © DEI/ISEP



# Valor e endereço de uma variável



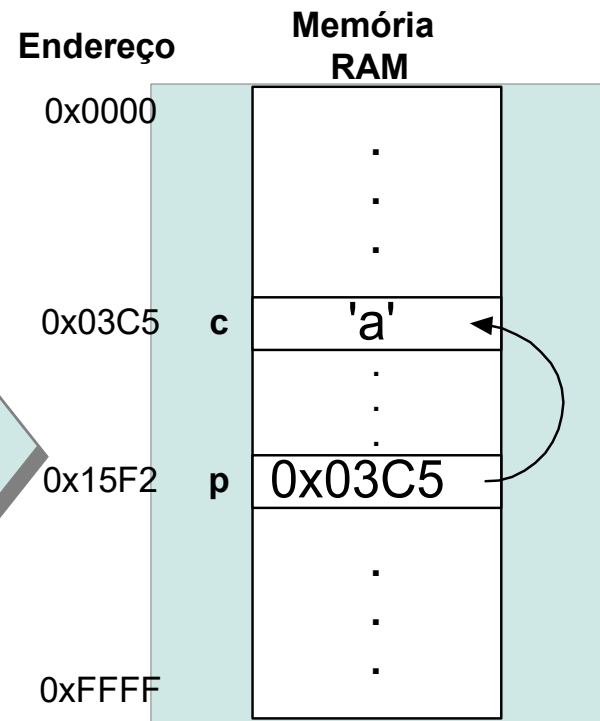
- Para ler qualquer variável do tipo *int*, *char*, *float* ou *double* utilizando a função ***scanf***, tem que preceder cada variável de um &

```
scanf("%c", &c);
```

```
char c; //c é um carácter
char *p; //p é um apontador para carácter.

p = &c; // p agora guarda o endereço de c

*p = 'a'; // c = 'a'
```



## Questões

