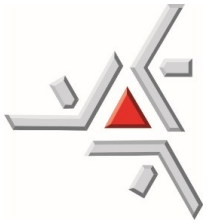


Introdução à Linguagem C

Disciplina: Estruturas de Dados

**Departamento de Informática
Centro de Tecnologia
Universidade Estadual de Maringá**



Introdução



- ❑ A linguagem C foi criada por Dennis Ritchie, em 1972, no centro de Pesquisas da Bell Laboratories.
- ❑ É uma linguagem de propósito geral, sendo adequada à programação estruturada.



Sintaxe



- ☐ Todo programa em C inicia sua execução chamando a função **main()**, sendo obrigatória a sua declaração no programa principal.
- ☐ Comentários no código:
 - `/* comentário de mais
de uma linha */.`
 - `// comentário de uma linha`
- ☐ Cada instrução encerra com ; (ponto e vírgula), que faz parte do comando.



Sintaxe Exemplo



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

```
// Função obrigatória em todos os programas escritos em C
```

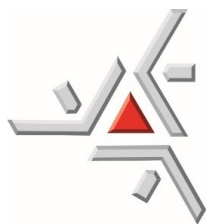
```
void main() {
```

```
    // comando 1;
```

```
    // comando 2;
```

```
}
```

Observação: a função **main()** também pode retornar um **int**.



Tipos de Dados

Principais tipos primitivos



Tipo de dado	Significado	Tamanho (em bytes)	Intervalo de valores aceitos
<i>char</i>	Caractere (código ASCII)	1	de -128 a 127
<i>short int</i>	Inteiro curto	2	-de 32 768 a 32 767
<i>unsigned short int</i>	Inteiro curto sem sinal	2	de 0 a 65 535
<i>int</i>	Inteiro	4	de -2 147 483 648 a 2 147 483 647
<i>unsigned int</i>	Inteiro sem sinal	4	de 0 a 4 294 967 295
<i>long int</i>	Inteiro longo	4	-de 2 147 483 648 a 2 147 483 647
<i>unsigned long int</i>	Inteiro longo sem sinal	4	de 0 a 4 294 967 295
<i>float</i>	Flutuante (real)	4	$3.4 \cdot 10^{-38}$ à $3.4 \cdot 10^{38}$
<i>double</i>	Flutuante duplo	8	de $1.7 \cdot 10^{-308}$ a $1.7 \cdot 10^{308}$
<i>long double</i>	Flutuante duplo longo	10	$3.4 \cdot 10^{-4932}$ à $3.4 \cdot 10^{4932}$



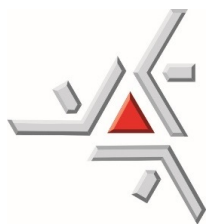
Operadores Aritméticos e de Atribuição



Operador	Sintaxe
Adição	$a + b$
Subtração	$a - b$
Incremento	$a++$
Decremento	$a--$

Operador	Sintaxe
Divisão	a / b
Multiplicação	$a * b$
Módulo (resto)	$a \% b$

Operador	Sintaxe
Atribuição	$a = b$



Operadores Comparativos



Operador	Sintaxe
Menor que	$a < b$
Menor ou igual que	$a \leq b$
Maior que	$a > b$
Maior ou igual que	$a \geq b$
Diferente de	$a \neq b$
Igual a	$a == b$
<i>Não</i> lógico	$!a$
<i>E</i> lógico	$a \&\& b$
<i>Ou</i> lógico	$a b$

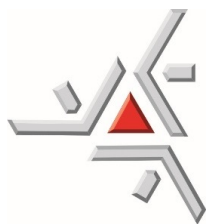


Função printf()



`printf("expressão de controle", argumentos);`

- ❑ É uma função de *Input/Output*, que permite escrever na tela.
- ❑ A expressão de controle pode conter:
 - caracteres que serão exibidos na tela; e
 - códigos de formatação que indicam o formato em que os argumentos devem ser impressos.
- ❑ Cada argumento deve ser separado por vírgula.

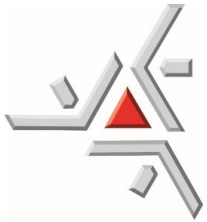


Função printf()



Símbolo	Efeito
\n	Nova linha
\t	Tab
\b	Retrocesso
\"	Aspas

Símbolo	Significado
%c	Caractere simples
%d	Decimal
%f	Ponto flutuante
%s	Cadeia de caracteres



Função printf()

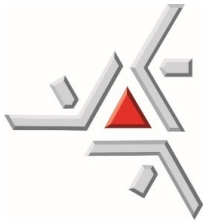


Exemplo

```
#include <stdio.h>

int main() {
    printf("Este e o numero dois: %d", 2);
    printf(" \n%s esta a %d milhoes de milhas
           \ndo sol", "Venus", 67);

    return 0;
}
```



Função printf()



arredondamento

```
#include <stdio.h>

int main() {
    printf("\n%4.2f", 3456.78);
    printf("\n%4.1f", 3456.78);
    printf("\n%4.3f", 3456.78);
    return 0;
}
```

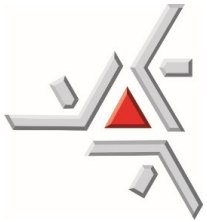


Função scanf()



- ❑ É o complemento de printf() e nos permite ler dados formatados da entrada padrão (teclado).
- ❑ Sua sintaxe é similar a printf().

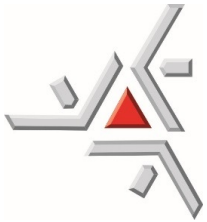
`scanf("expressão de controle", argumentos);`



Função scanf()



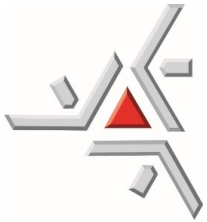
- ☐ A lista de argumentos deve consistir nos **endereços das variáveis**.
- ☐ A linguagem C oferece um operador para tipos básicos chamado operador de endereço: **&**



Função scanf()



- ❑ A memória do computador é dividida em bytes, e são numerados de 0 até o limite da memória.
- ❖ Estas posições são chamadas de endereços.
- ❖ Toda variável ocupa uma certa localização na memória, e seu endereço é o primeiro byte ocupado por ela.



Função scanf()



```
#include <stdio.h>

void main() {
    int num;

    printf("Digite um numero: ");
    scanf("%d", &num);

    printf("\nNumero: %d", num);
    printf("\nEndereco: %d", &num);
}
```



Estrutura de Controle de Fluxo - IF



```
if (condição)  
    comando;
```

```
else  
    comando;
```

```
if (condição) {  
    comando;  
} else {  
    comando;  
}
```

```
if (condição)  
{  
    comando1;  
    comando2;  
    ...
```

```
} else {  
    comando3;  
    comando4;  
    ...  
}
```

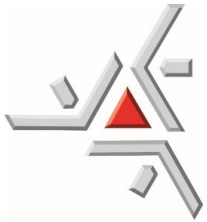
```
if (condição) {  
    comando1;  
    ...  
} else if  
    (condição) {  
        comando2;  
        ...  
} else {  
    ...  
}
```




Estrutura de Controle de Fluxo - IF



```
#include <stdlib.h>
#include <time.h>
#include <stdio.h>
void main() {
    int num, segredo;
    srand(time(NULL));
    segredo = rand() / 10;
    printf("Qual e o numero: ");
    scanf("%d", &num);
    if (segredo == num) {
        printf("Acertou!");
        printf("\nO numero e %d\n", segredo);
    } else if (segredo < num) {
        printf("Errado, muito alto!\n");
    } else {
        printf("Errado, muito baixo!\n");
    }
}
```



Loop for



for (inicialização; condição; incremento)

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

```
void main() {
```

```
    int x;
```

```
    for (x = 1; x < 100; x++) {
```

```
        printf("%d\n", x);
```

```
    }
```

```
}
```

```
for (;;) {
```

```
    printf("loop infinito\n");
```

```
}
```

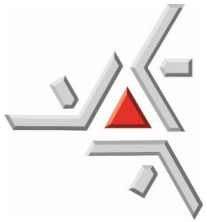


Loop for



```
#include <stdio.h>

void main() {
    int linha, coluna;
    for (linha = 1; linha <= 24; linha++){
        for (coluna = 1; coluna < 40; coluna++) {
            printf("-");
        }
        printf("\n");
    }
}
```



While



```
while (condição) {  
    comando;  
}
```

```
do {  
    comando;  
} while (condição);
```

```
void main() {  
    char ch;  
    while (ch != 'a') {  
        scanf("%c", &ch);  
    }  
}
```



Bibliografia



- ❑ KERNIGHAN, Brian W. e RITCH, Dennis M., **C: A Linguagem de Programação**, Rio de Janeiro, Campus, 1986.
- ❑ PLAUGER, P.J. e BRODIE J. **Standart C : guia de referência básica**, São Paulo, Mcgraw-Hill, 1991. 306p.
- ❑ HANCOCK, Les e KRIEGER, Morris. **Manual de Linguagem C**, Rio de Janeiro, Campus, 1985. 182p.



Bibliografia



- ❑ MIZRAHI, Viviane V. **Treinamento em Linguagem C** - módulo 1 e 2, São Paulo, McGraw-Hill, 1990, 241p.

- ❑ SCHILDT, Herbert. **Turbo C: Guia do Usuário**, São Paulo, McGraw-Hill, 1988, 414p.