



# **Programação de Computadores I**

## Funções Básicas da Linguagem C

PROFESSORA CINTIA CAETANO

# Função Printf()

---

- ▶ Sintaxe

`printf(“expressão de controle” , argumentos);`

- ▶ É uma função padrão para exibir uma mensagem na tela, podendo conter também formatadores padrão que indicam o tipo de variável a ser vinculada nessa mensagem.
- ▶ Os argumentos são as próprias variáveis, que podem ser inclusive impressas em forma de operações lógicas ou aritméticas. Cada argumento deve ser separado um do outro por vírgulas ( , ).

# Função Printf()

---

## Expressões de controle

<code>\n</code>	new line (pula linha)
<code>\t</code>	tab (tabulação horizontal)
<code>\b</code>	backspace (volta um caractere)
<code>\f</code>	form feed (avanço de página)
<code>\\</code>	imprimir a barra invertida
<code>\'</code>	imprimir o apóstrofe
<code>\"</code>	imprimir aspas

# Função Printf()

---

## Formatadores

`%d` ou `%i` inteiro

`%f` float

`%o` octal

`%x` hexadecimal (`%X` hexadecimal maiúsculo)

`%u` inteiro sem sinal (unsigned)

`%e` notação científica

`%s` string (seqüência de caracteres)

`%c` char

`%p` ponteiro

`%ld` ou `%li` long int

`%lf` double

# Função Printf(): Exemplos

---

## Exemplos:

```
main()
{
printf("Cinco é igual a  %d ", 5);
}
```

---

```
main()
{
int a=5 , b=10;
printf("O valor de  %d +  %d  é = %d ", a, b, a+b);
}
```

---

```
main()
{
printf("%s está a  %d está a milhares de milhas\n do sol","venus", 67);
}
```

---

```
main()
{
printf("%d  %c  %x  %o ", 'A', 'A' , 'A', 'A');
printf("%c  %c  %c  %c ", 'A', 65 , 0x41, 0101);
}
```

---

# DICAS

---

- ▶ Cuidar que a tabela ASCII possui 256 valores. Se por acaso passar desse número e tentarmos imprimir em formato caractere (%c) um número maior que 255, será impresso o resto da divisão do número por 256. Se o número for 3393 será impresso A, porque  $3393\%256$  é 65.

# Arredondamento

---

- ▶ Existe ainda uma maneira de limitar o número máximo de algarismos significativos nos formatadores.

```
#include<stdio.h>
void main()
{
    float a = 3456.78;
    printf("%4.2f\n ",a);
    printf("%3.2f\n ",a);
    printf("%3.lf\n ",a);
    printf("%10.3f\n ",a);
}
```

# Complementando com zeros à esquerda

---

```
#include<stdio.h>
void main()
{
    printf("%04d\n ",21);
    printf("%06d\n ",21);
    printf("%6.4d\n ",21);
    printf("%6.0d\n ",21);
}
```

# Função scanf()

---

- ▶ Da mesma forma que é importante imprimir dados na tela do computador, há necessidade de entrar dados pelo teclado.
- ▶ Sintaxe:  
scanf(“expressão de controle”, argumentos);

# Função scanf()

---

- ▶ Na expressão de controle valem os mesmos formatadores da função printf.
- ▶ Os argumentos são precedidos do & indicando uma operação utilizando endereço de memória para armazenar o valor da variável.

# Função scanf(): Exemplo 1

---

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

```
void main()
```

```
{
```

```
    int num;
```

```
    printf("Digite um número inteiro: ");
```

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

```
    printf("O número digitado foi %d ",num);
```

```
    printf("O endereço de memória onde foi armazenado %d  
é %u",num,&num);
```

```
}
```

## Função scanf(): Exemplo 2

---

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

```
void main(){
```

```
    int a,b;
```

```
    printf("Entre com o primeiro valor: ");
```

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

```
    printf("Entre com o segundo valor: ");
```

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

```
    printf("A soma eh %d", a+b);
```

```
}
```

# Função scanf() : Exemplo 3

---

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

```
void main()
```

```
{  
    char nome[50]; //define um vetor de caracteres com  
    tamanho máximo 50 posições  
    printf("Digite seu nome completo: ");  
    scanf("%s",&nome); // captura uma string do teclado  
    printf("O nome digitado foi %s ",nome);  
}
```

## Função scanf() : Exemplo 3

---

- ▶ Obs: Note que o scanf necessita do <enter> como confirmação do dado (int, float, char, string, etc.) digitado para continuar.
- ▶ Se o usuário digitar o nome completo, o scanf trunca a string assim que encontra o primeiro espaço em branco, porque o compilador entende que é o fim da string, imprimindo apenas o primeiro nome na tela.

# Função Gets() e Puts()

---

- ▶ Para contornar esse problema utiliza-se outra função, por exemplo: gets e puts.
- ▶ A função gets lê tudo o que foi digitado no teclado até pressionar <enter>. A função não armazena o <enter>, mas adiciona um caractere nulo(\0) para indicar o final da string.
- ▶ Sua sintaxe geral é:  
`gets(nome_variavel);`
- ▶ Se for ultrapassado o espaço reservado para a string, esta função sobrepõe os valores na memória, podendo ocasionar um erro grave.

# Função Gets() e Puts()

---

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

```
void main()
```

```
{
```

```
    char nome[50]; //define um vetor de caracteres de no máximo 50 posições
```

```
    printf("Digite seu nome completo: ");
```

```
    gets(nome); //lê uma string do teclado inclusive espaços em branco
```

```
    puts(nome); // coloca a string digitada no teclado
```

```
}
```

# Função Gets() e Puts()

---

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

```
#include <string.h>
```

```
void main ()
```

```
{
```

```
    char nome[14], nome2[14] = "Ana";
```

```
    printf("Nome: ");
```

```
    gets(nome);
```

```
    strcpy(nome2, nome); //função para realizar copiar string
```

```
    printf("Nome digitado: %s \n", nome);
```

```
    printf("Nome copiado: %s \n", nome2);
```

```
}
```

# Função Getchar()

---

- ▶ É uma função que captura um caractere do teclado (getchar), e o imprime na tela (putchar).

```
#include<stdio.h>
void main()
{
    char ch;
    printf("Pressione uma tecla qualquer ");
    ch=getchar();
    printf("A tecla pressionada foi: %c\n ",ch);
}
```

# Função Getch() e Getche()

---

- ▶ As funções `getch()` e `getche()` retornam o caracter pressionado.
- ▶ A função `getche()` imprime o caracter na tela antes de retorná-lo, enquanto que a função `getch()` apenas retorna o caracter sem imprimi-lo.
- ▶ Ambas são definidas no arquivo de cabeçalho `conio.h`, portanto, não pertencem ao padrão ANSI.
- ▶ A sintaxe destas funções é similar a da macro `getchar()`.

# Observação

---

- ▶ Há várias funções para manipulação de caracteres como: `isalpha( )`, `isupper( )`, `islower( )`, `isdigit( )`, `isspace( )`, `tolower( )`