



INSTITUTO FEDERAL

Bahia

Campus Santo Antônio de Jesus

<http://www.portal.ifba.edu.br/santoantonio>

LINGUAGEM DE PROGRAMAÇÃO

Prof. George Pacheco Pinto

AGENDA

- ❑ Linguagem C
 - ❑ Strings
 - ❑ Funções Matemática

STRING

- ❑ Tipo de dado estruturado que representa uma sequência de zero ou mais caracteres (letras, números e símbolos especiais);
- ❑ Vetor de caracteres. Denotadas em C entre aspas duplas, ex: “Linguagem”
- ❑ Coleção de bytes terminados por um caractere NULL (`\0`)

STRING

- ❑ Como nos vetores, os caracteres de uma cadeia podem ser individualizados através de um índice que indica a posição do caractere na cadeia:
 `s[i]` - referência ao i-ésimo caractere da cadeia “s”

char	char	char	char	char	char	char	char	char
L	I	N	G	U	A	G	E	M \0

STRING

- ❑ Em C para se trabalhar com cadeias de caracteres(String), utilizamos a biblioteca representada pela `#include <string.h>`.
- ❑ A biblioteca disponibiliza uma série de funções, como: comparação, concatenação dentre outras operações.

DECLARANDO STRINGS

- ❑ `char nome_variavel[TAMANHO_MAX];`

- ❑ Exemplo:

- ❑ `char sexo;`

- ❑ é uma variável que irá armazenar apenas um caractere denotado por apóstrofes ou aspas simples (ex. 'M' ou 'F').

- ❑ `char descricao[50];`

- ❑ é uma variável string que sustenta um máximo de 50 caracteres.

Obs.:

É preciso definir o tamanho máximo de caracteres que a String deverá comportar no momento de inicialização.

LEITURA DE STRING

- ❑ gets

- ❑ Função que efetua a leitura de uma String.
Aceita caracteres em branco no meio da string;

- ❑ `gets(nome_variavel);`

- ❑ Não verifica o limite máximo de caracteres de entrada, a qualquer momento compilador pode retornar erro de estouro de buffer.

- ❑ Exemplo:

- `gets(nome);`

LEITURA DE STRING

❑ fgets

- ❑ Lê uma string de tamanho TAM do dispositivo de entrada padrão e armazena esta string em s;
- ❑ É mais seguro, pois checa o tamanho máximo do vetor;
- ❑ fgets (s, TAM, stdin);
- ❑ fgets (nome, 10, stdin);

Obs: Antes de utilizar o gets e fgets, deve ser utilizado o **fflush(stdin);** para limpar o buffer. Isso é muito importante caso antes tenha sido utilizado o scanf.

IMPRIMIR STRINGS

`printf + operador %s`

`printf(“%s”, nome_variavel);`

Exemplo:

`printf(“O nome é %s”, nome);`

ACESSAR CARACTERE ESPECÍFICO

nome_string[posição]

Exemplo:

```
char nome[10];  
fflush(stdin);  
gets(nome);  
printf("%c", nome[0]); //G  
printf("%c", nome[1]); //e  
printf("%c", nome[2]); //o  
printf("%c", nome[3]); //r
```

G	e	o	r	g	e				
0	1	2	3	4	5	6	7	8	9

EXEMPLO

Escrever um programa que solicite o nome de uma pessoa e o imprima na tela, informando se a inicial do mesmo é uma vogal ou consoante.

SOLUÇÃO

```
#include <stdio.h>
int main()
{
    char nome[20];

    printf("Informe o seu nome: ");
    fflush(stdin);
    gets(nome);
    switch(nome[0])
    {
        case 'a':
        case 'e':
        case 'i':
        case 'o':
```

```
        case 'u':
            printf("%s inicia com vogal\n", nome);
            break;
        default:
            printf("%s inicia com consoante\n", nome);
            break;
    }
    return 0;
}
```

PERCORRENDO CARACTERES DE UMA CADEIA

```
#include <stdio.h>
int main() {
    int i, n;
    char nome[35];

    printf("Informe o seu nome:\n");
    gets(nome);
    printf("\n");

    for (i=0; nome[i] != NULL; i++) {
        printf("Posicao %d - %c\n", i, nome[i]);
    }
}
```

PERCORRENDO CARACTERES DE UMA CADEIA

```
char s[80];  
int ctmaius = 0, ctminus = 0, ctalfa = 0;
```

1) verificando se é um caractere alfabético maiúsculo

```
if ((s[i] >= 'A') && (s[i] <= 'Z'))  
    ctmaius = ctmaius + 1;
```

2) verificando se é um caractere alfabético minúsculo

```
if ((s[i] >= 'a') && (s[i] <= 'z'))  
    ctminus = ctminus + 1;
```

3) verificando se é um caractere alfabético

```
if (((s[i] >= 'A') && (s[i] <= 'Z')) ||  
    ((s[i] >= 'a') && (s[i] <= 'z')))  
    ctalfa = ctalfa + 1;
```

PERCORRENDO CARACTERES DE UMA CADEIA

```
char s[80];  
int ctnum = 0, cta = 0, ctbit = 0, ctespaco = 0;
```

```
4) verificando se é um caractere numérico  
  if ((s[i] >= '0') && (s[i] <= '9'))  
    ctnum = ctnum + 1;
```

```
5) verificando se é a vogal 'a'  
  if ((s[i] == 'A') || (s[i] == 'a'))  
    cta = cta + 1;
```

```
6) verificando se é um bit ('0' ou '1')  
  if ((s[i] == '0') || (s[i] == '1'))  
    ctbit = ctbit + 1;
```

```
7) verificando se é o espaço em branco  
  if (s[i] == ' ')  
    ctespaco = ctespaco + 1;
```

PERCORRENDO CARACTERES DE UMA CADEIA

```
char s[80];  
int cta = 0, cte = 0, cti = 0, cto = 0, ctu = 0;
```

8) verificando as vogais separadamente

```
if ((s[i] == 'A') || (s[i] == 'a'))  
    cta = cta + 1;  
else if ((s[i] == 'E') || (s[i] == 'e'))  
    cte = cte + 1;  
else if ((s[i] == 'I') || (s[i] == 'i'))  
    cti = cti + 1;  
else if ((s[i] == 'O') || (s[i] == 'o'))  
    cto = cto + 1;  
else if ((s[i] == 'U') || (s[i] == 'u'))  
    ctu = ctu + 1;
```


PERCORRENDO CARACTERES DE UMA CADEIA

```
char s[80];  
int ctcons = 0;
```

9) verificando se é uma consoante:

```
if (((s[i] >= 'A') && (s[i] <= 'Z')) ||  
    ((s[i] >= 'a') && (s[i] <= 'z')))) {  
  
    if ((s[i] != 'A') && (s[i] != 'a') &&  
        (s[i] != 'E') && (s[i] != 'e') &&  
        (s[i] != 'I') && (s[i] != 'i') &&  
        (s[i] != 'O') && (s[i] != 'o') &&  
        (s[i] != 'U') && (s[i] != 'u')) {  
        ctcons = ctcons + 1;  
    }  
}
```

FUNÇÕES DE MANIPULAÇÃO

- ❑ strcpy
- ❑ strcat
- ❑ strlen
- ❑strupr
- ❑strlwr
- ❑ strcmp

STRCPY

- ❑ Copia o conteúdo de um String para outra String
 - ❑ `strcpy(<variavel_destino>,<variavel_origem>);`

- ❑ Exemplo:
 - ❑ `strcpy(string1,string2);`

Obs.:

Em C é incorreto executar a seguinte instrução:

`string1 = string 2; ou string1 = "CARLOS";`

Não existe atribuição direta.

STRCAT

- ❑ Concatena o conteúdo de duas Strings
 - ❑ `strcat(<variavel_destino>,<variavel_origem>);`

- ❑ Exemplo:

```
string1 -> "Engenharia" e string2->"Computação"  
strcat(string1,string2);  
printf("%s",string1); //EngenhariaComputação  
printf("%s",string2); // Computação
```

Obs.:

A string de origem permanecerá inalterada e será anexada ao fim da string de destino.

STRLEN

- ❑ Retorna o tamanho de String

- ❑ `strlen(<variavel>);`

- ❑ Exemplo:

```
string1 -> "Linguagem de Programação"  
int size = 0;  
size = strlen(string1);  
printf("%d", size); //Imprime o valor 24
```

Obs.:

A função retorna um valor inteiro contendo o tamanho do vetor de caracteres.

STRUPR

- ❑ Coloca o conteúdo da String em MAIÚSCULO

- ❑ `strupr(<variavel>);`

- ❑ Exemplo:

- `string1 -> “Linguagem de Programação”`

- `strupr(string1);`

- `printf(“%s”,string1); // LINGUAGEM DE PROGRAMAÇÃO`

STRLWR

- ❑ Coloca o conteúdo da String em minúsculo

- ❑ `strlwr(<variavel>);`

- ❑ Exemplo:

- `string1 -> "LINGUAGEM DE PROGRAMAÇÃO"`

- `strlwr(string1);`

- `printf("%s",string1); // linguagem de programação`

STRCMP

- ❑ `int strcmp(s1, s2);`
 - ❑ Compara duas strings "exatamente", ou seja, caracteres minúsculos são considerados diferentes e maiores que os respectivos caracteres maiúsculos.
- ❑ `int stricmp(s1, s2);` ou `int strcmpi(s1, s2);`
 - ❑ compara duas strings sem "case sensitivity", ou seja, não diferencia caracteres minúsculos de caracteres maiúsculos.
- ❑ Estas duas rotinas retornam um valor inteiro como resultado da comparação:
 - 0 - significa que são iguais
 - Outro valor - diferentes

BIBLIOTECAS MATEMÁTICAS

❑ Em C para se trabalhar com funções matemáticas, utilizamos a biblioteca representada pela `#include <math.h>`.

- ❑ `pow`
- ❑ `sqrt`
- ❑ `log`
- ❑ `log10`
- ❑ `sin`
- ❑ `con`
- ❑ `tan`
- ❑ `acos`
- ❑ `asin`
- ❑ `atan`
- ❑ `ceil`
- ❑ `floor`
- ❑ `fabs`

BIBLIOTECAS MATEMÁTICAS

❏ pow

- ❏ Retorna um calculo exponencial
- ❏ `pow(<var_numero>, <var_expoente>);`

❏ sqtr

- ❏ Retorna o calculo de uma raiz quadrada
- ❏ `sqtr(<var_numero>);`

❏ log

- ❏ Retorna o resultado de um calculo de uma função logarítmica natural
- ❏ `log(<var_numero>);`

BIBLIOTECAS MATEMÁTICAS

❏ log10

- ❏ Retorna o resultado de um calculo de uma função logarítmica na base 10
- ❏ `log10(<var_numero>);`

❏ sin

- ❏ Realiza o calculo do seno
- ❏ `sin(<var_angulo>);`

❏ cos

- ❏ Realiza o calculo do cosseno
- ❏ `cos(<var_angulo>);`

BIBLIOTECAS MATEMÁTICAS

- ❑ `tan`
 - ❑ Realiza o calculo da tangente
 - ❑ `tan(<var_angulo>);`
- ❑ `Arco Seno`
 - ❑ `asin(<var_angulo>);`
- ❑ `Arco Cosseno`
 - ❑ `acos(<var_angulo>);`
- ❑ `Arco Tangente`
 - ❑ `atan(<var_angulo>);`

BIBLIOTECAS MATEMÁTICAS

- ❑ Arredondamento para cima

- ❑ `ceil(<var_numero>);`

- ❑ Arredondamento para baixo

- ❑ `floor(<var_numero>);`

- ❑ Valor Absoluto

- ❑ `fabs(<var_numero>);`

EXERCÍCIOS

1. Escreva um programa que solicite o nome de uma pessoa e o imprima todo em formato maiúsculo, informando se a inicial do mesmo é uma vogal ou consoante.
2. Faça um programa em C que leia nome, horas trabalhadas, salário-hora e sexo de um grupo de operários e que calcule e imprima: Salário total dos funcionários; O maior salário, juntamente com o nome de quem o recebeu; O número de funcionários do sexo masculino e feminino cadastrado; O percentual de funcionários homens e mulheres cadastradas.

EXERCÍCIOS

3. Escreva um programa em C para ler uma palavra e escrever:
A primeira letra da palavra.
A última letra da palavra.
O número de letras existente na palavra.

REFERÊNCIAS

Consultar ementário.