

Princípios de Desenvolvimento de Algoritmos MAC122

Prof. Dr. Paulo Miranda
IME-USP

Strings

Cadeias de caracteres (string)

- Sequência de:
 - letras ('a', 'b', 'A', 'B'),
 - símbolos ('!', '?', '+', '=', '%', ...),
 - espaços em branco (' ') e/ou
 - dígitos ('0', '1', ..., '9')
 - terminada pelo caracter '\0'.
- Na linguagem C as **strings** são armazenadas em vetores do tipo **char**.

```
char texto[500];
```

Tabela ASCII

- ASCII é uma padronização onde cada carácter é manipulado sob forma de código binário.

SIMB	DEC	BINÁRIO	SIMB	DEC	BINÁRIO
3	51	00110011	<	60	00111100
4	52	00110100	=	61	00111101
5	53	00110101	>	62	00111110
6	54	00110110	?	63	00111111
7	55	00110111	@	64	01000000
8	56	00111000	A	65	01000001
9	57	00111001	B	66	01000010
:	58	00111010	C	67	01000011
;	59	00111011	D	68	01000100

Exemplo:

```
char texto[11]; /*declaração*/  
texto[0] = 'B';  
texto[1] = 'e';  
texto[2] = 'm';  
texto[3] = '-';  
texto[4] = 'v';  
texto[5] = 'i';  
texto[6] = 'n';  
texto[7] = 'd';  
texto[8] = 'o';  
texto[9] = '!';  
texto[10] = '\\0';  
printf("%s\\n", texto);
```

Representação gráfica:

'B'	'e'	'm'	'-'	'v'	'i'	'n'	'd'	'o'	'!'	'\\0'
0	1	2	3	4	5	6	7	8	9	10

Exemplo:

```
/*Inicialização durante a declaração.*/  
char texto[] = {'B', 'e', 'm', '-', 'v', 'i', 'n', 'd', 'o', '!', '\0'};  
/* ou */  
char texto[] = "Bem-vindo!";  
  
printf("%s\n", texto);
```

Representação gráfica:

'B'	'e'	'm'	'-'	'v'	'i'	'n'	'd'	'o'	'!'	'\0'
0	1	2	3	4	5	6	7	8	9	10

Cadeias de caracteres (string)

- **Lendo da entrada padrão:**
 - **scanf("%s", texto);**
 - Lê cadeias de caracteres até encontrar espaço em branco, nova linha ou **EOF** (fim de arquivo).
 - **gets(texto);**
 - Lê caracteres incluindo espaços em branco, até encontrar nova linha ou **EOF**. Não deve ser usada pois apresenta problemas quando o texto digitado excede o tamanho da string.
 - **fgets(texto, TAM, stdin);**
 - Igual ao **gets** porém mais seguro, lê no máximo **TAM** caracteres.
 - **O caracter '\0' é inserido no final do vetor texto após a leitura em todos os casos.**

Cadeias de caracteres (string)

- Lendo da entrada padrão:

```
#include <stdio.h>
#define LIM 500

int main(){
    char texto[LIM];

    printf("Digite uma string: ");

    scanf("%s", texto);
    gets(texto);
    fgets(texto, LIM-1, stdin); /*Mais seguro!*/

    printf("texto: %s\n", texto);

    return 0;
}
```

Cadeias de caracteres (string)

- Lendo da entrada padrão (um caracter por vez):

```
#include <stdio.h>

int main(){
    char texto[500];
    int c,i=0;
    while(1){
        c = getchar(); /*Lê o próximo caracter.*/
        if(c==EOF || c=='\n')
            break;

        texto[i] = (char)c;
        i++;
    }
    texto[i] = '\0';
    printf("texto: %s\n", texto);
    return 0;
}
```


Cadeias de caracteres (string)

- Imprimindo em uma string:

```
#include <stdio.h>

int main(){
    char texto[500];
    float media= 5.5;

    /*Imprimindo na saída padrão.*/
    printf("media: %.2f\n",media);
    /*Grava no vetor.*/
    sprintf(texto, "media: %.2f\n",media);
    printf("%s", texto);

    return 0;
}
```

Cadeias de caracteres (string)

- Manipulando strings: `#include <string.h>`

```
#include <stdio.h>
#include <string.h>
int main(){
    char firstname[100]="Paulo";
    char lastname[100]="Miranda";
    char name[100];

    printf("%d\n",strlen(firstname)); /* Imprime 5. */
    printf("%d\n",strlen(lastname)); /* Imprime 7. */

    strcpy(name, firstname); /*Copia firstname.*/
    strcat(name, " ");        /*Adiciona Espaço em branco.*/
    strcat(name, lastname);  /*Adiciona lastname.*/
    printf("%d\n",strlen(name)); /*Imprime 13.*/
    printf("name: %s\n",name); /*Imprime nome completo.*/
    return 0;
}
```

Cadeias de caracteres (string)

- Uma possível implementação de **strlen** para ASCII:

```
#include <stdio.h>

int main(){
    char texto[]="Paulo Miranda";
    int i;

    i = 0;
    while(texto[i]!='\0')
        i++;
    printf("String possui %d caracteres.\n",i);

    return 0;
}
```

Exercício:

- Faça um programa que recebe uma linha de texto da entrada padrão, e lendo um caractere por vez, produza as subseqüências contíguas de caracteres não brancos, uma por linha. Exemplo:

Entrada:

0 provedor 123 oferece acesso!

Saída:

**0
provedor
123
oferece
acesso!**

Solução:

```
#include <stdio.h>

int main(){
    int c = getchar();
    do{
        while(c==' ') /*Remove espaços em branco.*/
            c = getchar();

        while(c!=' ' && c!='\n' && c!=EOF){
            printf("%c", (char)c);
            c = getchar();
        }
        printf("\n");
    }while(c!=EOF && c!='\n');

    return 0;
}
```

Exercício:

- Faça em programa que conta o número de palavras de um texto terminado com a palavra “end”. Exemplo:

Entrada:

**um programa que le as
notícias em um formato especial
chamado XML end**

Saída:

13 palavras

Solução:

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

int main(){
    char name[512];
    int r,c=0;
    do{
        r = scanf(" %s",name);
        if(r==EOF) break;
        c++;
    }while(strcmp(name,"end")!=0); /*strcmp compara strings*/

    printf("%d palavras\n",c);

    return 0;
}
```