Linguagem de Programação II

Prof. Mario Bessa

Aula 20

http://mariobessa.info

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’ (null character).
  - **“A” é diferente de ‘A’**
- Na linguagem C as **strings** são armazenadas em vetores do tipo **char**.

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

Tabela ASCII

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

SIMB	DEC	BINÁRIO
3	51	00110011
4	52	00110100
5	53	00110101
6	54	00110110
7	55	00110111
8	56	00111000
9	57	00111001
:	58	00111010
;	59	00111011

SIMB	DEC	BINÁRIO
<	60	00111100
=	61	00111101
>	62	00111110
?	63	00111111
@	64	01000000
A	65	01000001
B	66	01000010
C	67	01000011
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.
    - **stdin** dispositivo de entrada padrão (geralmente o teclado).
  - `int getchar(void)`
    - Lê um caracter.
  - **O caracter '\0' é inserido automaticamente no final do vetor `texto` após a leitura em todos os casos.**

Cadeias de caracteres (string)

- **Escrevendo na entrada padrão:**
  - `printf("%s",texto);`
    - Imprime, com formatação, cadeias de caracteres até encontrar o **EOF** (fim de arquivo).
  - `puts(texto);`
    - Imprime, sem formatação, cadeias de caracteres até encontrar o **EOF** (fim de arquivo).
  - `int putchar(int c);`
    - Imprime um caracter.

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("String: %s\n",texto);
  puts(texto);

  return 0;
}
```

Cadeias de caracteres (string)

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

```
#include <stdio.h>

int main(){
   int n;
   char letra;
   putchar('?');
   n=45;
   putchar(n-2);
   letra=getchar();
   putchar(letra);
   putchar('\n');
}
```

Cadeias de caracteres (string)

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

```c
#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);
  puts(texto);
  return 0;
}
```

Funções de manipulação de string

- #include <string.h>

Function **Operation**

strcpy(s1, s2)
Copies string s2 into string s1.

strcat(s1, s2)
Concatenates string s2 onto the end of string s1.

strlen(s1)
Returns the length of string s1.

strcmp(s1, s2)
Returns 0 if s1 and s2 are the same;
Less than 0 if s1<s2;
Greater than 0 if s1>s2.

strchr(s1, c)
Returns a pointer to the first occurrence of character c in string s1.

strstr(s1, s2);
Returns a pointer to the first occurrence of string s2 in string s1.

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 Espaco 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 para strlen:**

```
#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 subsequências contíguas de caracteres não brancos, uma por linha. Exemplo:

Entrada:

```
O   provedor 123  oferece     acesso!
```

Saída:

```
O
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 c=1;
  do{
    scanf("%s",name);
    if(strcmp(name, "end")==0) break;
    c++;
  }while(1);

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

  return 0;
}
```