





Aula 20: Array de caracteres e strings Introdução a Programação

Túlio Toffolo & Puca Huachi http://www.toffolo.com.br

BCC201 – 2019/1 Departamento de Computação – UFOP

Aulas anteriores

- Vetores
- Matrizes

Aula de hoje

- Cadeias de caracteres
- Parâmetros da função main
- 3 Exercícios
- Próxima aula

Aula de hoje

- Cadeias de caracteres
- Parâmetros da função mair
- 3 Exercícios
- Próxima aula

Revisão

- Em C uma cadeia de caracteres (string) é implementada como um vetor do tipo char.
- Variáveis do tipo char são usadas para armazenar um caracter (tamanho = 1 byte).

Revisão

Lembrando que...

Caracteres literais são representados por aspas simples:

```
char c1 = 'a';
char c2 = 'A';
```

- Variáveis do tipo char podem receber valores literais do tipo caractere ou também valores inteiros
 - Que representam o caractere correspondente, conforme o sistema de codificação adotado
 - Lembram da tabela ASCII?
- Assim, variáveis do tipo char podem também ter seu valor comparado com inteiros.

Tabela ASCII

Decimal	Hex	Char	Decimal	Hex	Char	Decimal	Hex	Char	Decimal	Hex	Char
0	0	[NULL]	32	20	[SPACE]	64	40	@	96	60	`
1	1	[START OF HEADING]	33	21	1	65	41	A	97	61	a
2	2	[START OF TEXT]	34	22		66	42	В	98	62	b
3	3	[END OF TEXT]	35	23	#	67	43	C	99	63	C
4	4	[END OF TRANSMISSION]	36	24	\$	68	44	D	100	64	d
5	5	[ENQUIRY]	37	25	%	69	45	E	101	65	е
6	6	[ACKNOWLEDGE]	38	26	&	70	46	F	102	66	f
7	7	[BELL]	39	27	1	71	47	G	103	67	g
8	8	[BACKSPACE]	40	28	(72	48	H	104	68	ĥ
9	9	[HORIZONTAL TAB]	41	29)	73	49	1	105	69	i .
10	Α	[LINE FEED]	42	2A	*	74	4A	J	106	6A	j
11	В	[VERTICAL TAB]	43	2B	+	75	4B	K	107	6B	k
12	С	[FORM FEED]	44	2C	,	76	4C	L	108	6C	1
13	D	[CARRIAGE RETURN]	45	2D		77	4D	M	109	6D	m
14	E	[SHIFT OUT]	46	2E		78	4E	N	110	6E	n
15	F	[SHIFT IN]	47	2F	1	79	4F	0	111	6F	0
16	10	[DATA LINK ESCAPE]	48	30	0	80	50	P	112	70	р
17	11	[DEVICE CONTROL 1]	49	31	1	81	51	Q	113	71	q
18	12	[DEVICE CONTROL 2]	50	32	2	82	52	R	114	72	r
19	13	[DEVICE CONTROL 3]	51	33	3	83	53	S	115	73	s
20	14	[DEVICE CONTROL 4]	52	34	4	84	54	T	116	74	t
21	15	[NEGATIVE ACKNOWLEDGE]	53	35	5	85	55	U	117	75	u
22	16	[SYNCHRONOUS IDLE]	54	36	6	86	56	V	118	76	v
23	17	[ENG OF TRANS. BLOCK]	55	37	7	87	57	W	119	77	w
24	18	[CANCEL]	56	38	8	88	58	X	120	78	X
25	19	[END OF MEDIUM]	57	39	9	89	59	Υ	121	79	y
26	1A	[SUBSTITUTE]	58	3A	1	90	5A	Z	122	7A	z
27	1B	[ESCAPE]	59	3B	;	91	5B	1	123	7B	{
28	1C	[FILE SEPARATOR]	60	3C	<	92	5C	\	124	7C	Í
29	1D	[GROUP SEPARATOR]	61	3D	=	93	5D	1	125	7D	}
30	1E	[RECORD SEPARATOR]	62	3E	>	94	5E	^	126	7E	~
31	1F	[UNIT SEPARATOR]	63	3F	?	95	5F	_	127	7F	[DEL]

Exemplo

Crie um programa em C/C++ que imprime o código (em decimal) relativo a um caractere digitado pelo usuário.

Exemplo

```
#include <stdio.h>
int main()
{
    char c;
    scanf("%c", &c);
    printf("Código (decimal) de '%c' é %d\n", c, (int) c);
    return 0;
}
```

Cadeia de caracteres

Cadeias de caracteres são simplesmente arrays/vetores de caracteres que terminam com o caractere '\0':

- O caractere especial '\0' indica o final da cadeia de caracteres
- Note que para armazenar 10 caracteres precisamos de 11 posições
 - Uma posição adicional para o caractere '\0'
- Estas cadeias são também chamadas de strings

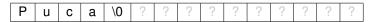
Exemplos

Suponha um array de 15 caracteres

• char nome[15]:



scanf("%s", nome);(suponha que o usuário digitou Puca)



Operações em cadeia de caracteres

A função strlen() (abreviação de *string length*) é utilizada para calcular o tamanho de uma *string*.

- Sim, calcular o tamanho: a função percorrerá o array de caracteres em busca do caractere '\0'.
- Lembre-se, portanto, que há um custo elevado em chamar essa função várias vezes.

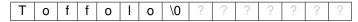
Operações em cadeia de caracteres

Exemplo:

```
char nome[15];
scanf("%s", nome); // le uma 'string' do usuário

int tamanho = strlen(nome); // calcula o tamanho da 'string'
printf("%d\n", tamanho);
```

Se o usuário digitar 'Toffolo', qual será o conteúdo do vetor?



O tamanho da cadeia de caracteres (ou seja, a 'posição' do \0):

```
1 7
```

Inicialização de strings

Uma cadeia de caracteres (*string*) pode ser inicializada facilmente:

```
char nome[] = "Toffolo";
```

Ao utilizar a construção acima, o array nome terá tamanho 7, pois o '\0'
é automaticamente inserido no final do vetor.

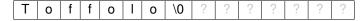
```
T o f f o I o \0
```

Inicialização de strings

Por outro lado, o código abaixo inicializa um array de 15 caracteres:

```
char nome[15] = "Toffolo";
```

- O array terá tamanho 15
- Note que o '\0' será inserido no vetor:



Exercício

Implemente a função strlen() com o seguinte protótipo:

```
1 int strlen(char str[]);
```

Biblioteca <string.h>

A biblioteca <string.h> contém algumas funções úteis:

- char *strcpy(char x[], char y[]); copia a string x (inclusive o byte \0 final) no espaço alocado para y. Cabe ao usuário garantir que o espaço alocado a y tem pelo menos strlen(x) + 1 bytes. Retorna o endereço de x.
- char *strcat(char x[], char y[]); concatena as strings x e y, isto é, acrescenta y ao final de x. Retorna o endereço da string resultante, ou seja, o endereço de x. Cabe ao usuário garantir que o espaço alocado a x é suficiente para comportar strlen(y) bytes adicionais.
- int strcmp(char x[], char y[]); compara lexicograficamente as strings x e y. Retorna um número estritamente negativo se x vem antes de y, 0 se x é igual a y e um número estritamente positivo se x vem depois de y.

Manipulação de strings

Podemos utilizar scanf para ler strings.

- No entanto, scanf finalizará a leitura quando encontrar um espaço.
- Exemplo:

```
char nomeCompleto[100];
printf("Digite o nome completo: ");
scanf("%s", nomeCompleto);
printf("Nome: %s", nomeCompleto);
```

Exemplo de execução do código acima:

```
Digite o nome completo: Tulio Angelo Machado Toffolo
Nome: Tulio
```

Manipulação de strings

Uma alternativa é utilizar a função gets ou fgets:

- gets(x): lerá da entrada o que for digitado pelo usuário até uma quebra de linha ('\n') ser detectada e armazenará na string x.
- fgets(x, n, stdin): lerá da entrada o que for digitado pelo usuário até uma quebra de linha ('\n') ser detectada ou o limite máximo de n caracteres ser atingido, e armazenará na string x.
 - stdin: constante que indica a entrada padrão (entrada do usuário). É uma abreviação para standard input.

Importante: fgets incluirá o caractere '\n' em x.

Manipulação de strings

Atenção: evite utilizar gets (função deprecated).

Mas... Porquê?

- Porque não sabemos quantos caracteres o usuário irá digitar...
- E podemos gerar inúmeros problemas por conta disso...
- Tente colocar 20 caracteres em um char nome [10];

Aula de hoje

- Parâmetros da função main

Parâmetros da função main

Até aqui, utilizamos a função main sem parâmetros:

```
int main()
{
      ...
      return 0;
}
```

Mas a função main também pode receber parâmetros:

• int argc: número de argumentos

19 / 23

- char *argv[]: valor dos argumentos
 - (poderia ser alternativamente char **argv)

Parâmetros da função main

O que o código acima irá imprimir?

Parâmetros da função main

O fato é que argv é um vetor de vetores de caracteres!

(ou um array de arrays de caracteres)

- Mas um vetor de caracteres pode ser visto como uma string...
- Portanto, argv é na prática um vetor de strings :)

Aula de hoje

- Cadeias de caracteres
- Parâmetros da função mair
- 3 Exercícios
- Próxima aula

Exercícios

Exercício 1

Crie um programa que verifica se a palavra "UFOP" (com letras maiúsculas) foi passada por argumento na linha de comando. Se sim, o programa deve imprimir:

```
Bem vindo a Ouro Preto!
```

Caso contrário, nada deve ser impresso pelo programa.

Aula de hoje

- Cadeias de caracteres
- Parâmetros da função mair
- 3 Exercícios
- Próxima aula

Próxima aula

- Aula prática: vetores, matrizes e cadeias de caracteres
- Aula teórica: revisão para a prova



Perguntas?