

Laboratório de Programação I (MATA57)



Prof.: Claudio Junior N. da Silva (claudiojns@ufba.br)

Strings

2023.1

Introdução

- Definição:
 - *String* é uma cadeia de caracteres;
 - Lista linear ou vetor onde cada elemento desse vetor é um caractere;
 - Agrupamento desses caracteres representam uma informação;
 - ✓ Palavra, frase, artigo, capítulo, livro,...

Representação

- Exemplo de representação de uma *string* como vetor ou cadeia de caracteres:

- "STRING"

	índice					
	0	1	2	3	4	5
variável	S	T	R	I	N	G

- Strings são frequentemente usadas para representar sequências de caracteres
 - `char vet[6] = {'s', 't', 'r', 'i', 'n', 'g'};`
 - C++: tipo **string**
- Caracteres da sequência pertencem ao **alfabeto ASCII**
 - 9 e '9' tem valores diferentes
 - 9 (ASCII): valor numérico para o tab ('\t')
 - '9': valor numérico 57 (ASCII)

ASCII TABLE

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	!	65	41	A	97	61	a
2	2	[START OF TEXT]	34	22	"	66	42	B	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	e
6	6	[ACKNOWLEDGE]	38	26	&	70	46	F	102	66	f
7	7	[BELL]	39	27	'	71	47	G	103	67	g
8	8	[BACKSPACE]	40	28	(72	48	H	104	68	h
9	9	[HORIZONTAL TAB]	41	29)	73	49	I	105	69	i
10	A	[LINE FEED]	42	2A	*	74	4A	J	106	6A	j
11	B	[VERTICAL TAB]	43	2B	+	75	4B	K	107	6B	k
12	C	[FORM FEED]	44	2C	,	76	4C	L	108	6C	l
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	/	79	4F	O	111	6F	o
16	10	[DATA LINK ESCAPE]	48	30	0	80	50	P	112	70	p
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	Y	121	79	y
26	1A	[SUBSTITUTE]	58	3A	:	90	5A	Z	122	7A	z
27	1B	[ESCAPE]	59	3B	;	91	5B	[123	7B	{
28	1C	[FILE SEPARATOR]	60	3C	<	92	5C	\	124	7C	
29	1D	[GROUP SEPARATOR]	61	3D	=	93	5D]	125	7D	}
30	1E	[RECORD SEPARATOR]	62	3E	>	94	5E	^	126	7E	~
31	1F	[UNIT SEPARATOR]	63	3F	?	95	5F	_	127	7F	[DEL]

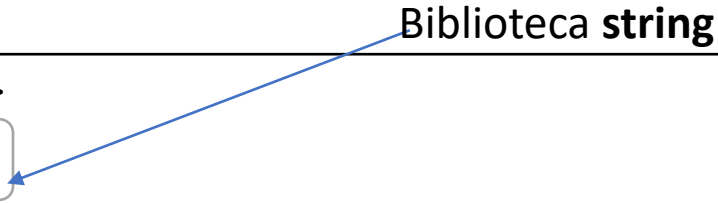
Declaração

- Como declarar uma string?
 - `string nome1 = "Fulano";`
 - `string nome2("Beltrano");`

Biblioteca **string**

```
#include <iostream>
#include <string>
using namespace std;

int main() {
    string str1 = "Boa noite!";
    string str2("Bonner");
}
```



Leitura

- Como ler uma string?
 - string nome;
 - cin >> nome;
 - getline(cin, nome);

```
#include <iostream>
```

```
#include <string>
```

```
using namespace std;
```

```
int main() {
```

```
    string str1, str2;
```

```
    cin >> str1; // lê uma palavra (para na leitura de espaço, tab ou enter  
                // ' ', '\t', '\n' não farão parte de str1
```

```
    getline (cin, str2); // lê uma linha inteira (para na leitura de enter)  
                        // '\n' (ENTER) não fará parte do conteúdo de str2
```

```
}
```



Impressão

- Como imprimir uma string?
 - string nome;
 - cout

```
#include <iostream>
#include <string>
using namespace std;

int main() {
    string nome;
    cin >> nome;
    cout << nome << endl; //imprime o conteúdo da string
}
```

Acesso

- Como acessar uma string?
 - `cout << nome[n] << endl;`
 - Acessar o n-ésimo caracter de uma string
 - `int tamanho = nome.size();`
 - Retorna o tamanho da string
 - Impressão sequencial de todos os caracteres?
`for(int i=0; i<tamanho; i++)`
 `cout << nome[i];`
 `cout << endl;`

Manipulação

- `string str1 = "Boa noite", str2, str3 = " senhores";`
 - a. `str2 = str1;`
 - copia `str1` para `str2`
 - b. `str2 = str2 + '!';`
 - adiciona o caractere `!` ao fim de `str2`
 - c. `str3 = str3 + "empresários";`
 - adiciona a string `"empresários"` ao fim de `str3`

`if (str1 == str2)`

`cout << "As strings são iguais" << endl;`

`else`

`cout << "As strings são diferentes" << endl;`

- compara `str1` e `str2` (`<`, `<=`, `>`, `>=` e `!=` também é possível)

Manipulação

```
#include<iostream>
#include <string>
using namespace std;

int main()
{
    string s1 = "Praticar";
    string s2 = "Prgramacao";
    string s3 = s1 + s2;

    if(s1 != s2)
        cout << s1 << " nao eh igual a " << s2 << endl;

    if(s1 > s2)
        cout << s1 << " eh maior que " << s2 << endl;

    else if(s1 < s2)
        cout << s1 << " eh menor que " << s2 << endl;

    if(s3 == s1 + s2)
        cout << s3 << " eh igual a " << s1 + s2 << endl;

    return 0;
}
```

Exercícios de sala

11. Faça um programa que peça para o usuário inserir uma frase e uma letra e que mostre na tela a quantidade de vezes que a letra inserida apareceu na frase.
12. O pessoal do RH está aplicando um teste para avaliar a capacidade de leitura dos candidatos. Para isso, foi solicitado para você um programa que receba como entrada uma frase e em seguida a frase seja impressa na ordem inversa. Exemplo:
 1. Entrada: o cacau esta caindo
 2. Saída: odniac atse uacac o
 3. Por fim, verifique se a frase se trata de um palíndromo (ex: Anotaram a data da maratona).
13. Os usuários descobriram que ocorre um erro no sistema toda vez que é digitada uma palavra com que contenha letra com acento (á,é,í,ó,ú,â,ê,ô,ã). O chamado foi direcionado para você. Sua tarefa é desenvolver um programa em C++ que solicite a entrada(leitura) de uma String qualquer, que verifique se existe alguma letra com acento e caso exista que substitua a substitua pela correspondente sem acento. Exemplo: á será substituída por a; é por e...
14. Para efeito de padronização, o Gerente de TI solicitou que você desenvolvesse um programa que, dada uma string, com letras maiúsculas ou não, transformasse tudo para letras minúsculas. (Verificar tabela com código ASCII)