



# Programação

# C++

Aula 04  
*Jorgiano Vidal*



- ❖ O tipo `char`
- ❖ Tabela ASCII
- ❖ Um texto/*string*: um *array* de `char`
- ❖ A classe `std::string` do C++
- ❖ Comentários sobre codificação de texto



# 0 tipo char

- ❖ Representa um caractere

- ❖ 8 bits

```
#include <iostream>
```

```
int main() {  
    char letra;  
    std::cin >> letra;  
    std::cout << "Você digitou o caractere " << letra << std::endl;  
    return 0;  
}
```



## ❖ Aspas simples

```
#include <iostream>

int main() {
    char letra;
    std::cin >> letra;
    if (letra == 'z') {
        std::cout << "z não é uma letra válida\n";
    } else {
        std::cout << "Você digitou o caractere " << letra << std::endl;
    }
}
```



# Tabela ASCII

Dec	Hex	Oct	Char	Dec	Hex	Oct	Char	Dec	Hex	Oct	Char	Dec	Hex	Oct	Char
0	0	0		32	20	40	[space]	64	40	100	@	96	60	140	`
1	1	1		33	21	41	!	65	41	101	A	97	61	141	a
2	2	2		34	22	42	"	66	42	102	B	98	62	142	b
3	3	3		35	23	43	#	67	43	103	C	99	63	143	c
4	4	4		36	24	44	\$	68	44	104	D	100	64	144	d
5	5	5		37	25	45	%	69	45	105	E	101	65	145	e
6	6	6		38	26	46	&	70	46	106	F	102	66	146	f
7	7	7		39	27	47	'	71	47	107	G	103	67	147	g
8	8	10		40	28	50	(	72	48	110	H	104	68	150	h
9	9	11		41	29	51	)	73	49	111	I	105	69	151	i
10	A	12		42	2A	52	*	74	4A	112	J	106	6A	152	j
11	B	13		43	2B	53	+	75	4B	113	K	107	6B	153	k
12	C	14		44	2C	54	,	76	4C	114	L	108	6C	154	l
13	D	15		45	2D	55	-	77	4D	115	M	109	6D	155	m
14	E	16		46	2E	56	.	78	4E	116	N	110	6E	156	n
15	F	17		47	2F	57	/	79	4F	117	O	111	6F	157	o
16	10	20		48	30	60	0	80	50	120	P	112	70	160	p
17	11	21		49	31	61	1	81	51	121	Q	113	71	161	q
18	12	22		50	32	62	2	82	52	122	R	114	72	162	r
19	13	23		51	33	63	3	83	53	123	S	115	73	163	s
20	14	24		52	34	64	4	84	54	124	T	116	74	164	t
21	15	25		53	35	65	5	85	55	125	U	117	75	165	u
22	16	26		54	36	66	6	86	56	126	V	118	76	166	v
23	17	27		55	37	67	7	87	57	127	W	119	77	167	w
24	18	30		56	38	70	8	88	58	130	X	120	78	170	x
25	19	31		57	39	71	9	89	59	131	Y	121	79	171	y
26	1A	32		58	3A	72	:	90	5A	132	Z	122	7A	172	z
27	1B	33		59	3B	73	;	91	5B	133	[	123	7B	173	{
28	1C	34		60	3C	74	<	92	5C	134	\	124	7C	174	
29	1D	35		61	3D	75	=	93	5D	135	]	125	7D	175	}
30	1E	36		62	3E	76	>	94	5E	136	^	126	7E	176	~
31	1F	37		63	3F	77	?	95	5F	137	_	127	7F	177	



# Tabela ASCII

Dec	Hex	Oct	Char	Dec	Hex	Oct	Char	Dec	Hex	Oct	Char	Dec	Hex	Oct	Char
0	0	0		32	20	40	[space]	64	40	100	@	96	60	140	`
1	1	1		33	21	41	!	65	41	101	A	97	61	141	a
2	2	2		34	22	42	"	66	42	102	B	98	62	142	b
3	3	3		35	23	43	#	67	43	103	C	99	63	143	c
4	4	4		36	24	44	\$	68	44	104	D	100	64	144	d
5	5	5		37	25	45	%	69	45	105	E	101	65	145	e
6	6	6		38	26	46	&	70	46	106	F	102	66	146	f
7	7	7		39	27	47	'	71	47	107	G	103	67	147	g
8	8	10		40	28	50	(	72	48	110	H	104	68	150	h
9	9	11		41	29	51	)	73	49	111	I	105	69	151	i
10	A	12		42	2A	52	*	74	4A	112	J	106	70	152	j
11	B	13		43	2B	53	+	75	4B	113	K	107	71	153	k
12	C	14		44	2C	54	,	76	4C	114	L	108	72	154	l
13	D	15		45	2D	55	-	77	4D	115	M	109	73	155	m
14	E	16		46	2E	56	.	78	4E	116	N	110	74	156	n
15	F	17		47	2F	57	/	79	4F	117	O	111	75	157	o
16	10	20		48	30	60	0	80	50	120	P	112	76	158	p
17	11	21		49	31	61	1	81	51	121	Q	113	77	159	q
18	12	22		50	32	62	2	82	52	122	R	114	78	160	r
19	13	23		51	33	63	3	83	53	123	S	115	79	161	s
20	14	24		52	34	64	4	84	54	124	T	116	7A	162	t
21	15	25		53	35	65	5	85	55	125	U	117	7B	163	u
22	16	26		54	36	66	6	86	56	126	V	118	7C	164	v
23	17	27		55	37	67	7	87	57	127	W	119	7D	165	w
24	18	30		56	38	70	8	88	58	130	X	120	7E	166	x
25	19	31		57	39	71	9	89	59	131	Y	121	7F	167	y
26	1A	32		58	3A	72	:	90	5A	132	Z	122	80	168	z
27	1B	33		59	3B	73	;	91	5B	133	[	123	81	169	{
28	1C	34		60	3C	74	<	92	5C	134	\	124	82	170	
29	1D	35		61	3D	75	=	93	5D	135	]	125	83	171	}
30	1E	36		62	3E	76	>	94	5E	136	^	126	84	172	~
31	1F	37		63	3F	77	?	95	5F	137	_	127	85	173	

'A'==65



# Tabela ASCII

Dec	Hex	Oct	Char	Dec	Hex	Oct	Char	Dec	Hex	Oct	Char	Dec	Hex	Oct	Char
0	0	0		32	20	40	[space]	64	40	100	@	96	60	140	`
1	1	1		33	21	41	!	65	41	101	A	97	61	141	a
2	2	2		34	22	42	"	66	42	102	B	98	62	142	b
3	3	3		35	23	43	#	67	43	103	C	99	63	143	c
4	4	4		36	24	44	\$	68	44	104	D	100	64	144	d
5	5	5		37	25	45	%	69	45	105	E	101	65	145	e
6	6	6		38	26	46	&	70	46	106	F	102	66	146	f
7	7	7		39	27	47	'	71	47	107	G	103	67	147	g
8	8	10		40	28	50	(	72	48	110	H	104	68	150	h
9	9	11		41	29	51	)	73	49	111	I	105	69	151	i
10	A	12		42	2A	52	*	74	4A	112	J	106	6A	152	j
11	B	13		43	2B	53	+	75	4B	113	K	107	6B	153	k
12	C	14		44	2C	54	,	76	4C	114	L	108	6C	154	l
13	D	15		45	2D	55	-	77	4D	115	M	109	6D	155	m
14	E	16		46	2E	56	.	78	4E	116	N	110	6E	156	n
15	F	17		47	2F	57	/	79	4F	117	O	111	6F	157	o
16	10	20		48	30	60	0	80	50	120	P	112	70	160	p
17	11	21		49	31	61	1	81	51	121	Q	113	71	161	q
18	12	22		50	32	62	2	82	52	122	R	114	72	162	r
19	13	23		51	33	63	3	83	53	123	S	115	73	163	s
20	14	24		52	34	64	4	84	54	124	T	116	74	164	t
21	15	25		53	35	65	5	85	55	125	U	117	75	165	u
22	16	26		54	36	66	6	86	56	126	V	118	76	166	v
23	17	27		55	37	67	7	87	57	127	W	119	77	167	w
24	18	30		56	38	70	8	88	58	130	X	120	78	170	x
25	19	31		57	39	71	9	89	59	131	Y	121	79	171	y
26	1A	32		58	3A	72	:	90	5A	132	Z	122	7A	172	z
27	1B	33		59	3B	73	;	91	5B	133	[	123	7B	173	{
28	1C	34		60	3C	74	<	92	5C	134	\	124	7C	174	
29	1D	35		61	3D	75	=	93	5D	135	]	125	7D	175	}
30	1E	36		62	3E	76	>	94	5E	136	^	126	7E	176	~
31	1F	37		63	3F	77	?	95	5F	137	_	127	7F	177	



## ❖ Ler um caractere e mostrar seu valor na tabela ASCII

```
#include <iostream>

int main(){
    char l;
    std::cin >> l;
    std::cout << "O valor ASCII de \'\"
                << l << "\' é \" << (int)l <<
                std::endl;
    return 0;
}
```





## ❖ Ler um caractere e mostrar seu valor na tabela ASCII

```
#include <iostream>

int main(){
    char l;
    std::cin >> l;
    std::cout << "O valor ASCII de \''
                << l << '\'' é " << static_cast<int>(l) <<
                std::endl;
    return 0;
}
```



## ❖ Soma de caracteres

```
#include <iostream>
```

```
void print_char(char c) {  
    std::cout << "0 caractere \" << c  
                << "\" tem código ASCII "  
                << static_cast<int>(c) << std::endl;  
    return;  
}
```

```
int main() {  
    char c1, c2, c3;  
    c1 = '1';  
    c2 = '2';  
    c3 = c1 + c2;  
    print_char(c1);  
    print_char(c2);  
    print_char(c3);  
}
```



## ❖ Soma de caracteres

```
#include <iostream>
```

```
void print_char(char c) {  
    std::cout << "O caractere \" << c  
                << "\" tem código ASCII "  
                << static_cast<int>(c) << std::endl;  
    return;  
}
```

```
int main() {  
    char c1, c2, c3;  
    c1 = '1';  
    c2 = '2';  
    c3 = c1 + c2;  
    print_char(c1);  
    print_char(c2);  
    print_char(c3);  
}
```

Saída do programa

0 caractere '1' tem código ASCII 49
0 caractere '2' tem código ASCII 50
0 caractere 'c' tem código ASCII 99



## ❖ Soma de caracteres

```
#include <iostream>
```

```
void print_char(char c) {  
    std::cout << "O caractere \" << c  
                << "\" tem código ASCII "  
                << static_cast<int>(c) << std::endl;  
    return;  
}
```

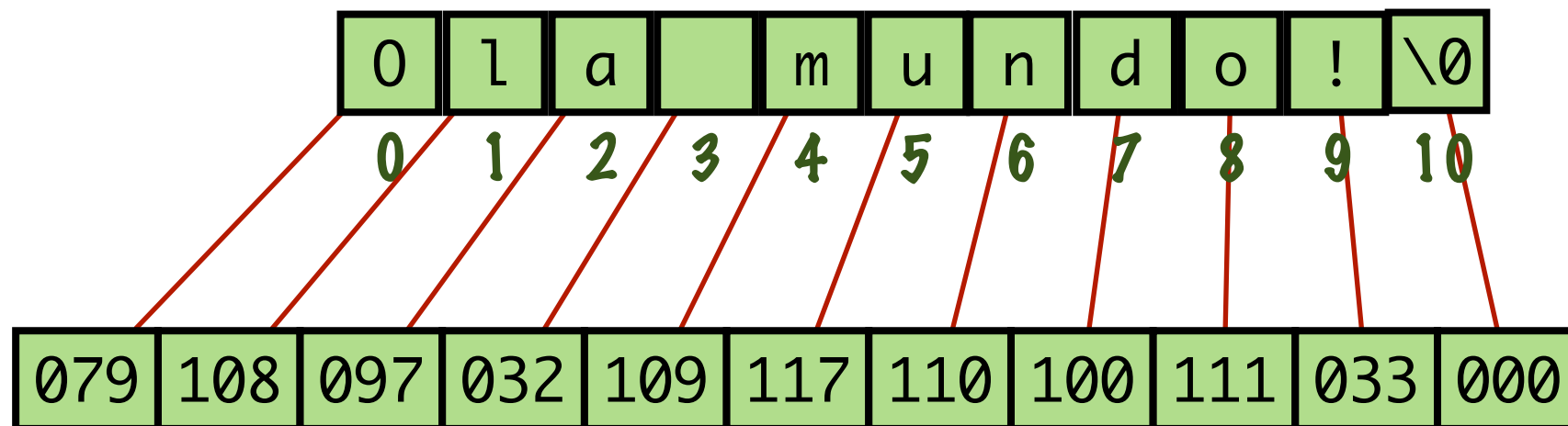
```
int main() {  
    char c1, c2, c3;  
    c1 = 50;  
    c2 = 52;  
    c3 = c1 + c2;  
    print_char(c1);  
    print_char(c2);  
    print_char(c3);  
}
```

Saída do programa

O caractere '2' tem código ASCII 50
O caractere '4' tem código ASCII 52
O caractere 'f' tem código ASCII 102



- ❖ Array de caracteres  
`char mensagem[11];`
- ❖ Termina com o caractere nulo
  - ❖ `'\0'` OU `0`
  - ❖ Ocupa espaço





```
char mensagem[15];
```

0	l	a		m	u	n	d	o	!	\0	?	?	?	?
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14

- ❖ Espaço limitado
- ❖ String é considerada até o '**\0**'



```
char mensagem[15];
```

0	l	a		m	u	n	d	o	!	\0	?	?	?	?
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14

❖ Espaço limitado

❖ String é considerada até o '**\0**'

```
#include <iostream>
```

```
int main() {  
    char ola[15] = "Ola mundo!";  
    std::cout << ola << std::endl;  
    return 0;  
}
```



```
char mensagem[15];
```

0	l	a		m	u	n	d	o	!	\0	?	?	?	?
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14

❖ Espaço limitado

❖ String é considerada até o '**\0**'

```
#include <iostream>
```

```
int main() {  
    char ola[15] = "Ola mundo!";  
    std::cout << ola << std::endl;  
    ola[3] = 0; // Pode ser ola[3] = '\0'  
    std::cout << ola << std::endl;  
    return 0;  
}
```





```
char mensagem[15];
```

0	l	a		m	u	n	d	o	!	\0	?	?	?	?
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14

❖ Espaço limitado

❖ String é considerada até o '**\0**'

```
#include <iostream>
```

```
int main() {  
    char ola[15] = "Ola mundo!";  
    std::cout << ola << std::endl;  
    ola[3] = 0; // Pode ser ola[3] = '\0'  
    std::cout << ola << std::endl;  
    ola[3] = 32; ola[4] = 77;  
    std::cout << ola << std::endl;  
    return 0;  
}
```



```
char mensagem[15];
```

0	l	a		m	u	n	d	o	!	\0	?	?	?	?
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14

❖ Espaço limitado

❖ String é considerada até o '**\0**'

```
#include <iostream>
```

```
int main() {  
    char ola[15] = "Ola mundo!";  
    std::cout << ola << std::endl;  
    ola[3] = 0; // Pode ser ola[3] = '\0'  
    std::cout << ola << std::endl;  
    ola[3] = 32; ola[4] = 77;  
    std::cout << ola << std::endl;  
    return 0;  
}
```

Execução do programa

```
Ola mundo!  
Ola  
Ola Mundo!
```



```
char mensagem[15];
```

0	l	a		m	u	n	d	o	!	\0	?	?	?	?
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14

❖ Espaço limitado

❖ String é considerada até o '**\0**'

```
#include <iostream>
```

```
int main() {  
    char ola[15] = "Ola mundo!";  
    std::cout << ola << std::endl;  
    ola[3] = 0; // Pode ser ola[3] = '\0'  
    std::cout << ola << std::endl;  
    ola[3] = 32; ola[4] = 77;  
    std::cout << ola << std::endl;  
    return 0;  
}
```

Execução do programa

```
Ola mundo!  
Ola  
Ola Mundo!
```



```
char mensagem[15];
```

0	l	a		m	u	n	d	o	!	\0	?	?	?	?
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14

❖ Espaço limitado

❖ String é considerada até o '**\0**'

```
#include <iostream>
```

```
int main() {  
    char ola[15] = "Ola mundo!";  
    std::cout << ola << std::endl;  
    ola[3] = 0; // Pode ser ola[3] = '\0'  
    std::cout << ola << std::endl;  
    ola[3] = 32; ola[4] = 77;  
    std::cout << ola << std::endl;  
    return 0;  
}
```

Execução do programa

```
Ola mundo!  
Ola  
Ola Mundo!
```



- ❖ Classe que encapsula operações em texto
- ❖ Definida no arquivo de cabeçalho `string`
  - ❖ `#include <string>`
  - ❖ Obs.: O arquivo `iostream` inclui o `string`

```
#include <iostream>
```

```
int main(){  
    std::string nome;  
    std::cin >> nome;  
    std::cout << "Olá "  
                << nome  
                << "!\n";  
    return 0;  
}
```



- ❖ Classe que encapsula operações em texto
- ❖ Definida no arquivo de cabeçalho `string`
  - ❖ `#include <string>`
  - ❖ Obs.: O arquivo `iostream` inclui o `string`

```
#include <iostream>
```

```
int main(){  
    std::string nome;  
    std::cin >> nome;  
    std::cout << "Olá "  
                << nome  
                << "!\n";  
    return 0;  
}
```

Execução do programa

```
Apolina  
Olá Apolina!
```



- ❖ Classe que encapsula operações em texto
- ❖ Definida no arquivo de cabeçalho `string`
  - ❖ `#include <string>`
  - ❖ Obs.: O arquivo `iostream` inclui o `string`

```
#include <iostream>
```

```
int main(){  
    std::string nome;  
    std::cin >> nome;  
    std::cout << "Olá "  
                << nome  
                << "!\n";  
    return 0;  
}
```

Execução do programa

```
Apolina  
Olá Apolina!
```

Execução do programa

```
Adalgisa Creuza  
Olá Adalgisa!
```



- ❖ Classe que encapsula operações em texto
- ❖ Definida no arquivo de cabeçalho **string**
  - ❖ **#include <string>**
- ❖ Obs.: O arquivo **iostream** inclui o **string**

```
#include <iostream>
```

```
int main(){  
    std::string nome;  
    std::cin >> nome;  
    std::cout << "Olá "  
                << nome  
                << "!\n";  
    return 0;  
}
```

Execução do programa

```
Apolina  
Olá Apolina!
```

Execução do programa

```
Adalgisa Creuza  
Olá Adalgisa!
```

Separador





❖ Função para ler linha

❖ `getline(cin, string)`

```
#include <iostream>
```

```
int main() {  
    std::string nome;  
    std::getline(std::cin, nome);  
    std::cout << "Olá " << nome << "!\n";  
    return 0;  
}
```



## ❖ Cuidado ao misturar

- ❖ `std::cin` não consome o separador, o `std::getline` sim!

```
#include <iostream>
```

```
int main() {  
    int idade;  
    std::string nome, endereco;  
    std::getline(std::cin, nome);  
    std::cin >> idade;  
    std::getline(std::cin, endereco);  
    std::cout << "Olá " << nome << "!\n";  
    std::cout << "Parabéns pelos seus " << idade << " anos de vida!\n";  
    std::cout << "Enviares o presente para " << endereco << std::endl;  
    return 0;  
}
```



## ❖ Cuidado ao misturar

- ❖ std::cin não consome o separador, o std::getline sim!

```
#include <iostream>
```

```
int main() {  
    int idade;  
    std::string nome, endereco;  
    std::getline(std::cin, nome);  
    std::cin >> idade;  
    std::cin.ignore();  
    std::getline(std::cin, endereco);  
    std::cout << "Olá " << nome << "!\n";  
    std::cout << "Parabéns pelos seus " << idade << " anos de vida!\n";  
    std::cout << "Enviaremos o presente para " << endereco << std::endl;  
    return 0;  
}
```



## ❖ Cuidado ao misturar

- ❖ std::cin não consome o separador, o std::getline sim!

```
#include <iostream>
```

```
int main() {  
    int idade;  
    std::string nome, endereco;  
    std::getline(std::cin, nome);  
    std::cin >> idade;  
    std::cin.ignore();  
    std::getline(std::cin, endereco);  
    std::cout << "Olá " << nome << "!\n";  
    std::cout << "Parabéns pelos seus " << idade << " anos de vida!\n";  
    std::cout << "Enviaremos o presente para " << endereco << std::endl;  
    return 0;  
}
```

Ignora caracteres

Sem parâmetros ignorar 1 caractere (separador no nosso caso)

[https://en.cppreference.com/w/cpp/io/basic\\_istream/ignore](https://en.cppreference.com/w/cpp/io/basic_istream/ignore)



## ❖ Pode ser visto, e usado, como *array* de caracteres

```
#include <iostream>
```

```
int main() {  
    std::string s;  
    getline(std::cin, s);  
    std::cout << s << std::endl;  
    unsigned int tamanho = s.length(); // Pode ser s.size()  
    for (unsigned int i = 0; i < tamanho; ++i) {  
        if (s[i] == ' ') { // espaço  
            s[i] = '_';  
        }  
    }  
    std::cout << s << std::endl;  
    return 0;  
}
```



## ❖ Pode ser visto, e usado, como *array* de caracteres

```
#include <iostream>
```

```
int main() {  
    std::string s;  
    getline(std::cin, s);  
    std::cout << s << std::endl;  
    unsigned int tamanho = s.length(); // Pode ser s.size()  
    for (unsigned int i = 0; i < tamanho; ++i) {  
        if (s[i] == ' ') { // espaço  
            s[i] = '-';  
        }  
    }  
    std::cout << s << std::endl;  
    return 0;  
}
```

Execução do programa

Este e um exemplo de como trabalhar com string como array!  
Este e um exemplo de como trabalhar com string como array!  
Este-e-um-exemplo-de-como-trabalhar-com-string-como-array!



## ❖ Converter tudo para maiúscula

```
#include <iostream>
```

```
void converter_para_maiuscula(std::string &s){  
    for (unsigned int i=0 ; i<s.size() ; ++i){  
        if (s[i]>='a' and s[i]<='z'){  
            s[i] = s[i] - 32; // Diferença entre minúscula e maiúscula  
                             // na tabela ASCII  
        }  
    }  
}  
  
int main(){  
    std::string texto;  
    getline(std::cin, texto);  
    converter_para_maiuscula(texto);  
    std::cout << texto << std::endl;  
    return 0;  
}
```



## ❖ Converter tudo para maiúscula

```
#include <iostream>
```

```
void converter_para_maiuscula(std::string &s){  
    for (unsigned int i=0 ; i<s.size() ; ++i){  
        if (s[i]>='a' and s[i]<='z'){  
            s[i] = s[i] - 32; // Diferença entre minúscula e maiúscula  
                             // na tabela ASCII  
        }  
    }  
}  
  
int main(){  
    std::string texto;  
    getline(std::cin, texto);  
    converter_para_maiuscula(texto);  
    std::cout << texto << std::endl;  
    return 0;  
}
```

Execução do programa

```
Isto é um texto que deve ser convertido para maiúscula!  
ISTO É UM TEXTO QUE DEVE SER CONVERTIDO PARA MAIÚSCULA!
```





## ❖ Converter tudo para maiúscula

```
#include <iostream>
```

```
void converter_para_maiuscula(std::string &s){  
    for (unsigned int i=0 ; i<s.size() ; ++i){  
        if (s[i]>='a' and s[i]<='z'){  
            s[i] = s[i] - 32; // Diferença entre minúscula e maiúscula  
                             // na tabela ASCII  
        }  
    }  
}  
  
int main(){  
    std::string texto;  
    getline(std::cin, texto);  
    converter_para_maiuscula(texto);  
    std::cout << texto << std::endl;  
    return 0;  
}
```

Execução do programa

```
Isto é um texto que deve ser convertido para maiúscula!  
ISTO é UM TEXTO QUE DEVE SER CONVERTIDO PARA MAIÚSCULA!
```



## ❖ Converter tudo para maiúscula

```
#include <iostream>
```

```
void converter_para_maiuscula(std::string &s){  
    for (unsigned int i=0 ; i<s.size() ; ++i){  
        if (s[i]>='a' and s[i]<='z'){  
            s[i] = s[i] - 32; // Diferença entre minúscula e maiúscula  
                             // na tabela ASCII  
        }  
    }  
}  
  
int main(){  
    std::string texto;  
    getline(std::cin, texto);  
    converter_para_maiuscula(texto);  
    std::cout << texto << std::endl;  
    return 0;  
}
```

Execução do programa

```
Isto é um texto que deve ser convertido para maiúscula!  
ISTO é UM TEXTO QUE DEVE SER CONVERTIDO PARA MAIÚSCULA!
```



## ❖ Caracteres acentuados

```
#include <iostream>
```

```
int main(){  
    std::string s;  
    std::cin >> s;  
    std::cout << s << " tem " << s.length() << " caracteres.\n";  
}
```



## ❖ Caracteres acentuados

```
#include <iostream>

int main(){
    std::string s;
    std::cin >> s;
    std::cout << s << " tem " << s.length() << " caracteres.\n";
}
```

Execução do programa

```
órgão
órgão tem 7 caracteres.
```



- ❖ Tabela ASCII

- ❖ 7 bits

- ❖ Extensões incluem acentos: 128 até 255 (8 bits)

- ❖ Diferentes codificações, por região/país/localidade

- ❖ UNICODE

- ❖ 16 e 32 bits: Muitos símbolos

- ❖ Java, C#: char = 16 bits

- ❖ UTF-8

- ❖ Mistura ASCII com UNICODE: Otimiza espaço

- ❖ Python usa UTF-8 por padrão

- ❖ Está se tornando padrão dos Sistemas Operacionais



❖ Dúvidas?



# Programação

# C++

Aula 04  
*Jorgiano Vidal*