

Prática sobre array de caracteres: Strings

```
// Exemplo de como usar o getchar

char nome[50]; //guarda o nome
char c;
int i, pos; // pos guarda a posição atual no vetor

pos = 0;
c = getchar(); // busca uma letra no buffer
while (c != '\n') { // fica no loop enquanto não encontrar um ENTER
    nome[pos] = c;
    pos++;
    c = getchar();
}

for (i=0; i < pos; i++){
    printf("%c", nome[i]);
}
```

- 1) TABELA ASCII: o tipo **char** pode receber valores numéricos. Como ele ocupa 1 byte, os valores variam de -128 .. +127. Já o tipo **unsigned char**, que também ocupa 1 byte, representa números de 0 .. 255. Cada valor de um tipo **char** tem um caractere associado na tabela ASCII (a tabela original tem 7 bits, indo de 0 a 127)

No comando printf, pode-se utilizar **%d** para mostrar uma variável **unsigned char**, será mostrado um número. Caso queira mostrar o caractere ASCII correspondente ao número armazenado na variável, utilize **%c**.

Monte um programa que mostra os valores e os caracteres correspondentes da tabela ASCII, para valores de 32 à 126. Mostrar 7 caracteres por linha. Para padronização, utilize três caracteres para representar o valor numérico, separe o número da letra com espaço, e dada par número/letra com tabulação (/t)

Exemplo de saída:

32	33 !	34 "	35 #	36 \$	37 %	38 &
39 '	40 (41)	42 *	43 +	44 ,	45 -
46 .	47 /	48 0	49 1	50 2	51 3	52 4
53 5	54 6	55 7	56 8	57 9	58 :	59 ;
60 <	61 =	62 >	63 ?	64 @	65 A	66 B
67 C	68 D	69 E	70 F	71 G	72 H	73 I
74 J	75 K	76 L	77 M	78 N	79 O	80 P
81 Q	82 R	83 S	84 T	85 U	86 V	87 W
88 X	89 Y	90 Z	91 [92 \	93]	94 ^
95 _	96 `	97 a	98 b	99 c	100 d	101 e
102 f	103 g	104 h	105 i	106 j	107 k	108 l
109 m	110 n	111 o	112 p	113 q	114 r	115 s
116 t	117 u	118 v	119 w	120 x	121 y	122 z
123 {	124	125 }	126 ~			

- 2) Utilizando os conceitos aprendidos até o momento (vetores e tipo **char**), crie um programa que leia o nome de uma pessoa e, em seguida, mostre o nome lido. Utilizar `getchar()` para entrada dos dados (veja acima um exemplo de como usar o `getchar()`). O nome pode ter no máximo 50 caracteres. Não usar as funções de `string.h`

Exemplo de saída:

```
<< Vetor de char>>
Digite um nome: Bruno
O nome digitado é: Bruno
```

- 3) Faça um programa semelhante ao anterior, só que o nome digitado deve ser mostrado em letras maiúsculas (pense como fazer isso olhando a diferença entre os valores dos caracteres minúsculos e maiúsculos na tabela ASCII). Não usar as funções de `string.h`

Exemplo de saída:

```
<< Vetor de char>>
Digite um nome: Bruno
O nome digitado é: BRUNO
```

- 4) Faça um programa em que o usuário digita dois nomes, e o programa informa se os nomes são iguais. Não usar as funções de `string.h`

Exemplo de saída:

```
<< Compara nomes>>

Digite o nome 1: Bruno
Digite o nome 2: Bruno

Os nomes digitados são iguais
```

Exemplo de saída:

```
<< Compara nomes>>

Digite o nome 1: Bruno
Digite o nome 2: BrunO

Os nomes digitados são diferentes
```

- 5) Refaça o exercício anterior, para que agora não haja diferenciação entre caracteres maiúsculos e minúsculos.

Exemplo de saída:

```
<< Compara nomes>>
Digite o nome 1: Bruno
Digite o nome 2: BrunO
Os nomes digitados são iguais
```

```
<< Compara nomes>>
Digite o nome 1: Bruno
Digite o nome 2: Carlos
Os nomes digitados são diferentes
```

- 6) Faça um programa que recebe o nome e sobrenome de uma pessoa (máximo 40 caracteres), copie o nome para um vetor e o sobrenome para outro. Mostre os dois vetores copiados.

```
<< Nomes >>

Digite o primeiro nome e último nome: Carlos Silva
Nome: Carlos
Sobrenome: Silva
```

- 7) O especificador de tipo `%s` pode ser usado para mostrar *strings* (vetores de caracteres). O programa abaixo mostra um texto na tela com a palavra “Brasil”. Modifique o programa abaixo para exibir o mesmo vetor usando `%c` e `%s`. A saída deve ser como a indicada mais abaixo (obrigatório uma saída com `%c` e outra com `%s`).

```
int main(void) {
    char nome[10] = "0123456789";

    nome[0] = 'B';
    nome[1] = 'r';
    nome[2] = 'a';
    nome[3] = 's';
    nome[4] = 'i';
    nome[5] = 'l';
    int i,tamanho = 6;

    for (i=0; i < tamanho; i++){
        printf("%c",nome[i]);
    }

    return 0;
```

```
}
```

Saída esperada

```
Usando %c: Brasil  
Usando %s: Brasil
```