

VETORES

Bibliografia:

Cap 6 - Estudo Dirigido de Linguagem C

José Augusto Manzano, Ed. Erica

Aula 1 de Vetores

Objetivos

- Entender o conceito de vetores
- Aplicar comandos para manipulação de vetores

Conceito de Vetores

- O vetor permite trabalhar com “n” dados, do mesmo tipo, em uma mesma variável.
- É também conhecido como matriz unidimensional

Sintaxe

tipo MATRIZ[dimensão];

<tipo> O tipo de dado a ser guardado

<matriz> Nome atribuído à matriz

<dimensão> Tamanho da matriz

Exemplo:

```
float MediaAluno[40];
```

Na definição acima é criada uma variável para armazenar a Média de 40 alunos.

A referência é sempre feito de 0 até $n - 1$, assim sendo MediaAluno[0] corresponde à média do primeiro aluno e MediaAluno[39] corresponde à média do último aluno

Programa Exemplo e Atividade

- Programa: 4.1.Leitura10notas

O programa lê e exibe notas de 10 alunos.

Procure entender linha a linha o que está acontecendo com o programa

Atividade

Programa: 4.2.MediaMaiorMenorNota

Após entrar com as 10 notas, o programa deve calcular a média da sala, localizar a menor nota e a maior nota.

Exibir a Média, Maior Nota e o número do aluno que tirou a maior nota, Menor Nota e o número do aluno que tirou a menor nota. Não exibir as notas individuais dos alunos.

Exercício Extra

- Comparação de médias

Permitir que o programa leia a nota de duas turmas (turma A e turma B), ambas com 10 alunos.

Calcular a média.

Identificar qual a sala que obteve maior média, qual foi a maior nota entre as duas salas, e a menor nota entre as duas salas. Identifique também o número do aluno que obteve a maior e a menor nota.

AULA 2 de Vetores

Objetivo

- Entender e exercitar como a linguagem C trabalha com strings (vetor ou matriz de uma dimensão)

Caracteres

- Vimos que caracter armazena letras, porém, isto é parcialmente verdade.
- O caracter armazena um número, em um byte, que é o código corresponde a uma letra.
- Na tabela ASCII abaixo, 33 corresponde a !, 65 a letra “A”, 97 à letra “a” e assim por diante.

	0	1	2	3	4	5	6	7	8	9
30			sp	!	"	#	\$	%	&	'
40	()	*	+	,	-	.	/	0	1
50	2	3	4	5	6	7	8	9	:	;
60	<	=	>	?	@	A	B	C	D	E
70	F	G	H	I	J	K	L	M	N	O
80	P	Q	R	S	T	U	V	W	X	Y
90	Z	[\]	^	_	`	a	b	c
100	d	e	f	g	h	i	j	k	l	m
110	n	o	p	q	r	s	t	u	v	w
120	x	y	z	{		}	~			

Caracteres

Experimente digitar e executar o programa abaixo:

```
#include <stdio.h>
```

```
#include <stdlib.h>
```

```
int main(void)
```

```
{
```

```
    char TESTE=65;
```

```
    printf("%d -- %c\n", TESTE,TESTE);
```

```
    TESTE='B';
```

```
    printf("%d -- %c\n", TESTE,TESTE);
```

```
    system("pause");
```

```
    return(0);
```

```
}
```

Na tabela ASCII, existem alguns outros caracteres de controle:

0: *null* (nulo)

7: *bell* (campainha do micro)

8: *backspace*

9: *tab*

10: *newline*, ou mudança de linha

13: *carriage return*, quebra e volta ao início da linha

127: *delete*

String

- Em C, a string é um vetor de caracteres, terminado com caracter “\0”, que é associado quando se tecla o enter do teclado

`char NOME[10]`

A definição acima indica um vetor, com 10 posições, das quais 9 podem conter caracteres, pois a última posição armazena o “\0”

Atenção: se o usuário digitar mais de 9 posições, e isto não for controlado pelo programador, o conteúdo digitado irá invadir a área de memória de outras variáveis, com resultados imprevisíveis.

Programa:

4.4.a.NomeSobrenome

Experimente digitar no nome: “Maria Sharapova”

Verá que o sistema não pede a entrada do sobrenome. Isto porque a função `scanf` é limitada, e não aceita caracter em branco. Neste caso, ele assume que após o caracter em branco, a segunda parte do que foi digitado é a entrada da segunda variável

String

Para resolver o problema de entrada de caracter branco, vamos utilizar a função `fgets()`. O programa abaixo também usa a função `fputs()` (procure na Internet para ver para que serve esta função).

Observe a sintaxe: `fgets(nome_da_variável, quantidade de caracteres, buffer utilizado);`

Sendo que o buffer utilizado é o `stdin` (standard input), que é o teclado do micro

Programa:

4.4.b.NomeSobrenome2

Exercício

- Programa: 4.5.VetorCaracteres

Como a string é um vetor de caracteres, é possível fazer referência uma a uma.

Crie uma variável `CIDADE[15]` e `BAIRRO[15]` e peça para o usuário digitar.

Depois, faça um laço `for`, e mostre o conteúdo de `CIDADE[0]` até `CIDADE[14]` e veja o resultado.

Exiba normalmente `CIDADE` e `BAIRRO`

Experimente digitar na cidade: `sao jose do rio preto`

Verá que o campo cidade fica com o conteúdo “sao jose do ri”, e que “o preto” invade o espaço da variável bairro.

Para resolver este problema, pesquise na internet “como limpar o buffer de entrada em linguagem C”.

Contando caracteres

- Programa: 4.6.QuantCaracteresString
- Este programa varre a variável até encontrar o “\0” que indica que o usuário teclou enter
- Depois exibe a quantidade de caracteres tem as duas strings digitadas

Atividade extra

- Faça um programa que concatene uma string.
- Dado um Nome e um Sobrenome, crie uma variável NomeCompleto, que receba o conteúdo de Nome e Sobrenome.