

Programação de Computadores – Aula 5

Prof. Msc. Carlos Alberto

Carlosqueiroz.fate@gmail.com

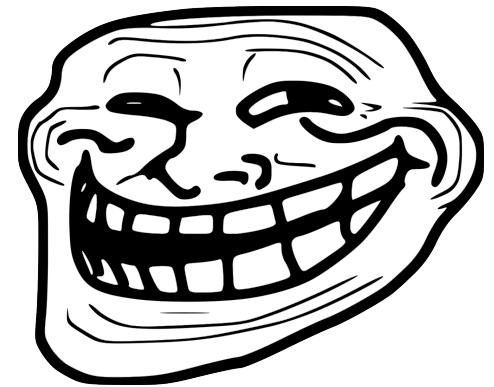
Vetores, Matrizes e Strings

Exercício

- Escreva um programa que recebe as notas de cinco estudantes. Em seguida, mostre as notas cujo o valor está acima da média das notas digitadas.

Exercício

- Vamos agora adaptar o programa para lidar com uma turma de 100 alunos...



Vetores, Matrizes e Strings

- Coleção de variáveis do mesmo tipo referenciadas por um mesmo nome
- Um elemento específico é acessado por meio de um índice
- Constituem em posições contíguas na memória
- Podem ter uma ou várias dimensões

Vetores, Matrizes e Strings

- String é um array unidimensional de `char` terminado com `'\0'` (caractere nulo)

Vetores

■ Forma Geral

Declaração

<Tipo da variável> <nome da variável> [tamanho];

Vetores

■ Exemplo

```
char alfabeto[27];
```

```
alfabeto[0] // Começa do zero
```

```
alfabeto[1]
```

```
alfabeto[2]
```

```
...
```


Vetores

- Atenção, ninguém impede que você faça isso...

alfabeto[50]

alfabeto[51]

...

Em C não há verificação de limites no acesso

Matrizes e Strings

- Matrizes unidimensionais
 - Tipo nome_var[tamanho];

```
main() {  
    int i, x[100];  
    for(i = 0; i < 100; i++) x[i] = i;  
}
```

```
x[-1] = -1;  
x[100] = 100;
```

```
short int i, x[5];  
for(i = 0; i < 5; i++)    x[i] = i;
```

```
short *x0, *x1, *x2;  
x0 = &x[0]; //00ff0000  
x1 = &x[1]; //00ff0002  
x2 = &x[2]; //00ff0004  
x0 = &x;    //00ff0000  
x0 = x;     //00ff0000
```

| End. | Valor |
|----------|----------|
| 00FF0000 | 00000000 |
| 00FF0001 | 00000000 |
| 00FF0002 | 00000000 |
| 00FF0003 | 00000001 |
| 00FF0004 | 00000000 |
| 00FF0005 | 00000010 |
| 00FF0006 | 00000000 |
| 00FF0007 | 00000011 |
| 00FF0008 | 00000000 |
| 00FF0009 | 00000100 |
| ... | |
| 00FF2056 | 00000000 |

Matrizes e Strings

```
void funtion(int arg[5]) {  
    ...;  
}
```

```
void funtion(int arg[]) {  
    ...;  
}
```

```
void funtion(int *arg) {  
    ...;  
}
```

```

short int x[5];
short *px = x; //00ff0000
*(px) = 0;
*(px + 1) = 1; //x[1]
*(px + 2) = 2; //x[2]
*((*int)px + 1) = 2;

```

```

void funtion(int *arg) {
    x[3] = 3; /*(arg + 3)
    x[4] = 4; /*(arg + 4)
}

```

```

funtion(x);
funtion(&x);
funtion(&x[0]);

```

| End. | Valor |
|----------|----------|
| 00FF0000 | 00000000 |
| 00FF0001 | 00000000 |
| 00FF0002 | 00000000 |
| 00FF0003 | 00000001 |
| 00FF0004 | 00000000 |
| 00FF0005 | 00000010 |
| 00FF0006 | 00000000 |
| 00FF0007 | 00000011 |
| 00FF0008 | 00000000 |
| 00FF0009 | 00000100 |
| ... | |
| 00FF2056 | 00000000 |

Matrizes e Strings

■ Matrizes multidimensionais

```
tipo nome[tamanho][tamanho]...
```

```
int matrix[10][3][4];  
func(matrix);
```

```
void func(int d[][3][4]){
```

```
void func(int*** d){}
```

Matrizes e Strings

■ Inicialização de matriz

```
int i[10] = {1,2,3,4,5,6,7,8,9,0};
```

```
char i[2] = "C é legal";
```

```
char i[2] = {'C','é',' ','l'};
```

```
int i[2][5] = {1,2,3,4,5,  
               6,7,8,9,0};
```

```
char i[] = "C é legal";
```

```
int i[] = {1,2,3,4,5,6,7,8,9,0};
```

```
int i[][5] = {1,2,3,4,5,  
              6,7,8,9,0};
```

String

- `char nome_da_string[tamanho];`
- O final deStrings
- Atenção, não faça isso...
 - `string1 = string2 ; // Errado!!`

String

■ Gets(string);