

Laboratório de Introdução à Ciência da Computação I

Aula de Vetores e Matrizes

Professor: Jó Ueyama

Estagiária PAE: Aurea Soriano

Vetores em C

- Variável para armazenar valores homogêneos e é unidimensional
- Declaração em C

<tipo> <nome_variável> [<tamanho>]

Ex:

```
int x[10];
```

- Como trabalhar com “string” em C?

Vetores em C

- Um vetor do tipo “char” pode armazenar “string”
- Note que uma string sempre termina com o caracter *null* (“\0”)
- Exemplo em C

```
char nome[3] = "Ana";
```

```
char sobrenome[] = {'H','i','t','s'};
```

```
printf("%s,%d\n",nome,strlen(nome));
```

```
printf("%s,%d\n",sobrenome,strlen(sobrenome));
```

```
char sobrenome[] = {'H','i','t','s','\0'};
```

Funções para strings

- Manipulação de strings:

`strlen();`

`strcat();`

`strcmp();`

`strcpy();`

- Entrada e saída

`gets();`

`puts();`

Matrizes

- Modos de inicialização:

```
int mat[2][3];
```

```
int mat[2][3] = {{1,2,3},{4,5,6}};
```

```
int mat[][3] = {{1,2,3},{4,5,6}};
```

Passagem de parâmetros

- Variável simples

```
main()
{
    int x = 5;
    func_a(x); //chamada da função
}
```

```
func_a(int y){ //função que recebe parâmetro
//func_a(y) int y;{ //outra forma de receber parâmetro
```

```
    ....
    ....
}
```

Passagem de parâmetros

- Variável simples (mais de 1 parâmetro)

```
main()
```

```
{
```

```
    int x = 5;
```

```
    int y = 7;
```

```
    func_a(x, y); //chamada da função
```

```
}
```

```
func_a(int x, int y){ //função que recebe parâmetro
```

```
//func_a(x, y) int x, y;{ //outra forma de receber parâmetro
```

```
    ....
```

```
    ....
```

```
}
```

Passagem de parâmetros - vetor

```
main() {  
    int x[3] = {1,2,3};  
    func_a(x,3); //chamada da função  
}
```

```
func_a(int y[],int tam){ //função que recebe  
    parâmetro  
    ....  
    ....  
}
```


Passagem de parâmetros - matriz

```
main()
{
    int x[2][3] = {1,2,3};
    func_a(x,6); //chamada da função
}

func_a(int y[],int tam){ //função que recebe
    parâmetro
    ....
    ....
}
```

Passagem de parâmetros

- Matriz (variação)

```
main()
{
    int x[2][3] = {1,2,3};
    func_a(x,2,3); //chamada da função
}
```

```
func_a(int y[][3], int l, int c)
{ //função que recebe parâmetro
    ....
    ....
}
```

Exercício 1: vetor e função

- Escreva um programa que identifica o menor valor e o maior valor de um vetor unidimensional
- Implemente cada operação (maior e menor) numa função independente, passando o conteúdo do vetor por parâmetro

Para facilitar a depuração do programa, imprima o conteúdo e o resultados das operações

Exercício 2: matriz e função

- Escreva um programa que gere duas matrizes A e B quadradas de ordem 10 e que realize as seguintes operações:
 - transposta de A (A^t);
 - produto de $A \times B$;
 - calcule o determinante de A;

Implemente cada operação numa função independente, passando o conteúdo da matriz por parâmetro

Para facilitar a depuração do programa, imprima o conteúdo de A e B e o resultados das operações

Exemplo com vetor

```
#include <stdio.h>
#include <stdlib.h>
#define MAX 3
int main(int argc, char *argv[])
{
    int nota[MAX]; nota[0] = 10; nota[1] = 20; nota[2] = 30;
    //int nota[MAX] = {10,20,30};
    //int nota[] = {10,20,30};
    int media,x,acc=0;
    for(x=0; x<MAX;x++){
        acc += nota[x];
        printf("%d\n",nota[x]);
    }
    media = acc/MAX;
    printf("%d\n",media);
    system("PAUSE");
    return 0;
}
```

Note que em C o primeiro elemento do vetor é o índice [0]

Exercício

- Escreva um programa que encontre o maior e o menor número de um vetor de inteiros de 5 elementos e mostre as suas respectivas posições;
 - Inicialize o vetor com números randômicos
 - Ex:

```
int x;
```

```
x = rand();
```

Exercício

- Faça um programa que preencha um primeiro vetor com dez números inteiros e um segundo vetor com cinco números inteiros. O programa deverá mostrar uma lista dos números do primeiro vetor com seus respectivos divisores armazenados no segundo vetor, bem como suas posições
 - Para saber se um número inteiro é divisível por outro, deve-se calcular o resto da divisão
 - Exemplo em C

```
int x = 10, y = 2;  
printf("%d\n",x%y);
```

Exercício

- Faça um programa que realize as seguintes operações com a string “ano2010”:
 - Converta o “ano” para caracteres maiúsculos
 - Saída: ANO2010
 - Insira um caracter de espaço em branco na quarta posição sem que haja perda de conteúdo
 - Saída: ANO 2010
 - Inverta o ordem da string
 - Saída: 0102ONA

Exercício

- Escreva um programa para calcular o fatorial de um número inteiro digitado pelo usuário
- Escreva um programa para converter um valor informado pelo usuário da moeda no padrão USD para EU. O valor deverá ser armazenado no formato de string
 - Padrão:
 - USD 1,234.56 para Eu 1.234,56

Solução: fatorial

```
int main()
{
    int num;
    long resposta;
    while(1)
    {
        printf("\n Digite o número: ", 163);
        scanf("%d", &num);
        resposta=1;
        while(num > 1)
            resposta *= num--;
        printf("O fatorial %c: %1d\n", 130, resposta);
        break;
    }
    system("pause");
}
```

Referências

Ascencio AFG, Campos EAV. Fundamentos de programação de computadores. São Paulo : Pearson Prentice Hall, 2006. 385 p.

VICTORINE VIVIANE MIZRAHI, Treinamento em Linguagem C – Módulo 1 e Módulo 2, Makron Books, 1990.