

Vetores, Matrizes e Funções

Disciplina de Programação de Computadores I
Universidade Federal de Ouro Preto

Vetores como parâmetro de sub-rotinas

- Vetores podem ser passados como parâmetros em sub-rotinas.
- Ao se passar um vetor como parâmetro, deve-se passar, também, o seu tamanho.
- Ao se passar um vetor como parâmetro de sub-rotina, **não é criado um novo vetor** local na sub-rotina.
- Isto significa que **os valores de um vetor são alterados dentro de uma sub-rotina!**

Vetores como parâmetros de sub-rotinas: exemplo

```
void proc5(int vet[], int tam){
    int i;
    for(i=0; i<tam; i++)
        vet[i]=5;
}
int main(void){
    int i, tamanho = 10, x[tamanho];
    for(i=0; i< tamanho; i++)
        x[i]=8;
    proc5(x, tamanho);
    for(i=0; i< tamanho; i++)
        printf("%d\n", x[i]);
    return 0;
}
```

Vetores e retornos de sub-rotinas

- Vetores não podem ser retornados por sub-rotinas
- Pode-se utilizar o fato de que vetores são alterados dentro de sub-rotinas para simular o retorno de um vetor por uma sub-rotina
- Basta que um procedimento receba o vetor e o altere o seu conteúdo.
- O conteúdo será visto por quem chamou com as alterações.

Vetores como parâmetros de sub-rotinas: exemplo

```
void leVet(int vet[], int tam){
    int i;
    printf("Digite %d numeros: ",
tam);
    for(i = 0; i < tam; i++)
        scanf("%d", &vet[i]);
}
```

```
void escreveVet(int vet[], int tam){
    int i;
    for(i=0; i< tam; i++)
        printf("vet[%d] = %d\n", i,
vet[i]);
}
```

```
int main(int){
```

```
    int vet1[10], vet2[20];
    leVet(vet1,10);
    leVet(vet2,20);
    escreveVet(vet1,10);
    escreveVet(vet2,20);
    return 0;
}
```

Vetores e Funções- Exemplo

Faça um procedimento que faça a leitura um vetor de 10 elementos inteiros e imprima somente os valores armazenados nos índices pares (considere o índice 0 (zero) como sendo “par”).

Vetores e Funções- Exemplo

```
void imprime_par() {
    int vetor[10];
    int i;
    for (i=0; i<10; i++) {
        printf("Digite o valor da posição %i do vetor: ", i);
        scanf("%i", &vetor[i]);
    }
    for (i=0; i<10; i=i+2) {
        printf("Posição %i - Valor: %i\n", i, vetor[i]);
    }
}
```

```
int main() {
    imprime_par();
    return 0;
}
```

Vetor multidimensional como parâmetro de sub-rotina

- Ao passarmos um vetor de qualquer dimensão como parâmetro de sub-rotina, o mesmo poderá ser alterado dentro da sub-rotina.

```
void zeraMatriz(int matriz[2][2]) {  
    int i, j;  
    for (i = 0; i < 2; i++)  
        for (j = 0; j < 2; j++)  
            matriz[i][j] = 0;  
}
```

```
int main(){  
    int mat[2][2] = { {0,1} , {2,3} };  
    zeraMatriz(mat);  
    return 0;  
}
```


Matrizes e Funções - Exemplo

Faça um programa que peça ao usuário para digitar os valores de uma matriz quadrada de números inteiros de dimensão 4. Posteriormente, crie uma função que encontre o maior valor desta matriz e outra função que encontre o menor valor.

Imprima os valores encontrados na função principal (main).

Matrizes e Funções - Exemplo

```
int encontra_maior (int matriz[4][4]){  
    int i, j, maior = matriz[0][0];  
    for (i=0; i<4; i++)  
        for (j=0; j<4; j++)  
            if (matriz[i][j] > maior)  
                maior = matriz[i][j];  
    return maior;  
}
```

```
int encontra_menor (int matriz[4][4]){  
    int i, j, menor = matriz[0][0];  
    for (i=0; i<4; i++)  
        for (j=0; j<4; j++)  
            if (matriz[i][j] < menor)  
                menor = matriz[i][j];  
    return menor;  
}
```

Matrizes e Funções - Exemplo

```
int main()
{
    int i, j , matriz[4][4];
    for (i=0; i<4; i++){
        for (j=0; j<4; j++) {
            printf("Digite a posicao [%i,%i]:", i+1, j+1);
            scanf("%i", &matriz[i][j]);}}

    int maior, menor;
    maior = encontra_maior(matriz);
    menor = encontra_menor(matriz);
    printf("O maior valor da matriz é %i", maior);
    printf("O menor valor da matriz é %i", menor);
    return 0;
}
```

Referências Bibliográficas

- Material de aula da disciplina Algoritmos, UFJF:
<https://sites.google.com/site/algoritmosufjf>
- Material de aula do Prof. Ricardo Anido, da UNICAMP:
<http://www.ic.unicamp.br/~ranido/mc102/>
- Material de aula da Profa. Virgínia F. Mota:
<https://sites.google.com/site/virginiaferm/home/disciplinas>
- DEITEL, P; DEITEL, H. *C How to Program*. 6a Ed. Pearson, 2010.

Agradecimentos

- Professores do Departamento de Ciência da Computação da UFJF que gentilmente permitiram a utilização das videoaulas elaboradas por eles.