

MC-102 — Aula 14

Strings e Matrizes

Instituto de Computação – Unicamp

30 de Abril de 2013

Roteiro

1 Strings

- Strings: Exemplos

2 Matrizes

- Exemplos com Matrizes

3 Exercícios

Strings

- A linguagem C não possui o tipo *string* explicitamente, mas podemos considerar um vetor de caracteres como uma *string*.
- Em C uma string é sempre terminada pelo caracter especial: `'\0'`
- **Portanto sempre declare uma string com um caracter a mais do que precisa!**
 - ▶ Se por exemplo estivermos trabalhando com strings de 10 caracteres:
`char st[11];`

Strings

- Para ler ou imprimir uma string do teclado usamos o operador especial `%s`.

```
int main(){
    char st[80];
    int a;

    printf("\nEntre com nome:");
    scanf("%s",st);
    printf("\nEntre com idade:");
    scanf("%d",&a);
    printf("\n Digitado: %s e %d\n",st,a);
}
```

- **Note que para strings não é utilizado o `&` no comando `scanf`.**

Strings

- Para ler strings **incluindo espaços** use a opção: `%[ ^ \n ]`.

```
int main(){
    char st[80];
    int a;

    printf("\nEntre com nome:");
    scanf("%[^\n]",st);
    printf("\nEntre com idade:");
    scanf("%d",&a);
    printf("\n Digitado: %s e %d\n",st,a);
}
```

Strings: Exemplos

- Ler uma string de até 80 caracteres e salvar a inversa desta em um vetor.
- Imprimir a inversa da string lida.

Strings: Exemplos

```
int main(){
    char st[80], stInv[80];
    int tam, i, j;

    printf("Entre com a string: ");
    scanf("%s",st);

    tam = 0;
    while(st[tam] != '\0'  && tam < 80){
        tam++;
    }

    stInv[tam] = '\0';

    j = tam-1;
    i = 0;
    while(i<tam){
        stInv[j] = st[i];
        i++;
        j--;
    }

    printf("A inversa e: %s\n",stInv);
}
```

Strings: Exemplos

A mesma coisa mas com laço **for**:

```
int main(){
    char st1[80], stInversa[80];
    int i, j , tam;

    printf("Digite um texto (max. 80):");
    scanf("%s",st1);

    for(tam=0; (st1[tam] != '\0') && (tam < 80) ; tam++)
        ;

    stInversa[tam] = '\0';
    for(j = tam-1, i = 0 ; j >= 0 ; j--, i++){
        stInversa[j] = st1[i];
    }

    printf("A inversa e: %s\n", stInversa);
}
```


Strings: Exemplos

- Ler uma string de até 80 caracteres e salvar a inversa desta em um vetor.
- Imprimir a inversa da string lida.
- **Não usar vetor adicional!**

Strings: Exemplos

```
int main(){
    char st1[80], aux;
    int i, j, tam;

    printf("Digite um texto (max. 80):");
    scanf("%s",st1);

    tam=0;
    while(st1[tam] != '\0' && tam < 80){
        tam++;
    }

    i = 0;
    j = tam -1;
    while(i < j){
        aux = st1[i];
        st1[i] = st1[j];
        st1[j] = aux;
        i++; j--;
    }

    printf("A inversa:%s\n",st1);
}
```

Matrizes

Suponha que queremos ler as notas de 4 provas para cada aluno e então calcular a média do aluno e a média da classe. O tamanho máximo da turma é de 50 alunos.

Solução

Criar 4 vetores de tamanho 50 cada. Cada vetor representa as notas dos alunos em uma prova.

```
float nota0[50],nota1[50],nota2[50],nota3[50];
```

Matrizes

- Agora suponha que estamos trabalhando com no máximo 100 provas. Seria muito cansativo criar 100 vetores, um para cada prova.
- Para resolver esse problema podemos utilizar matrizes. Uma matriz é um vetor (ou seja, um conjunto de variáveis de mesmo tipo) que possui duas ou mais dimensões, resolvendo para sempre essa questão.

Declarando uma matriz

```
<tipo> nome_da_matriz [<linhas>] [<colunas>]
```

- Uma matriz possui $linhas \times colunas$ variáveis do tipo `<tipo>`.
- As linhas são numeradas de 0 a $linhas - 1$.
- As colunas são numeradas de 0 a $colunas - 1$.

Exemplo de declaração de matriz

```
int matriz [4] [4];
```

	0	1	2	3
0				
1				
2				
3				

Acessando uma matriz

- Em qualquer lugar onde você escreveria uma variável no seu programa, você pode usar um elemento de sua matriz, da seguinte forma:

```
nome_da_matriz [<linha>] [<coluna>]
```

Ex: `matriz [1] [10]` — Refere-se a variável na 2ª linha e na 11ª coluna da matriz.

- Lembre-se que, assim como vetores, a primeira posição em uma determinada dimensão começa no índice 0.**
- O compilador não verifica se você utilizou valores válidos para a linha e para a coluna.

Declarando uma matriz de múltiplas dimensões

```
<tipo> nome_da_matriz [< dim1 >] [< dim2 >] ... [< dimN >]
```

- Essa matriz possui $dim_1 \times dim_2 \times \dots \times dim_N$ variáveis do tipo `<tipo>`
- Cada dimensão é numerada de 0 a $dim_i - 1$

Declarando uma matriz de múltiplas dimensões

- Você pode criar por exemplo uma matriz para armazenar a quantidade de chuva em um dado dia, mês e ano:

```
double chuva[31][12][3000];
```

```
chuva[23][3][1979] = 6.0;
```

Exemplos com Matrizes

Lendo uma matriz 4×4 do teclado:

```
/*Leitura*/  
for (i = 0; i < 4; i++)  
    for (j = 0; j < 4; j++) {  
        printf ("Matriz[%d][%d]: ", i, j);  
        scanf ("%d", &matriz[i][j]);  
    }
```

Exemplos com Matrizes

Escrevendo uma matriz 4×4 na tela:

```
/*Escrita*/  
for (i = 0; i < 4; i++) {  
    for (j = 0; j < 4; j++)  
        printf ("%d ", matriz[i][j]);  
    printf ("\n");  
}
```

Exemplos com Matrizes

- Ler duas matrizes 4×4 e calcular a soma das duas.

Exemplos com Matrizes

```
int main(){
    double mat1[3][3], mat2[3][3], mat3[3][3];
    int i,j;

    printf("\n **** Dados da Matriz 1 ****\n");
    for(i=0; i<3; i++){
        for(j=0; j<3; j++){
            printf("Entre com dado da linha %d - coluna %d: ", i, j);
            scanf("%lf", &mat1[i][j]);
        }
    }
    printf("\n **** Dados da Matriz 2 ****\n");
    for(i=0; i<3; i++){
        for(j=0; j<3; j++){
            printf("Entre com dado da linha %d - coluna %d: ", i, j);
            scanf("%lf", &mat2[i][j]);
        }
    }
    .....
    .....
```

Exemplos com Matrizes

```
int main(){
    double mat1[3][3], mat2[3][3], mat3[3][3];
    int i,j;

    .....
    .....
    .....

    for(i=0; i<3; i++){
        for(j=0; j<3; j++){
            mat3[i][j] = mat1[i][j] + mat2[i][j];
        }

    printf("\n **** Dados da Matriz 3 ****\n");
    for(i=0; i<3; i++){
        for(j=0; j<3; j++){
            printf("%lf, ", mat3[i][j]);
        }
        printf("\n");
    }
}
```

Exercício

- Escreva um programa que lê uma string de até 50 caracteres, e imprime “Palindromo” caso a string seja um palindromo e “Nao Palindromo” caso contrário.
- OBS: Um palindromo é uma palavra ou frase, que é igual quando lida da esquerda para a direita ou da direita para a esquerda (acentos e espaços em brancos são descartados).
- Exemplo de palindromo: Saudável leva duas.

Exercícios

Escreva um programa que leia todas as posições de uma matriz 10×10 . O programa deve então exibir o número de posições não nulas na matriz.

Exercícios

- Escreva um programa que lê todos os elementos de uma matriz 4×4 e mostra a matriz e a sua transposta na tela.

Matriz

$$\begin{bmatrix} 0 & 1 & 0 & 2 \\ 0 & 1 & 0 & 2 \\ 0 & 1 & 0 & 2 \\ 0 & 1 & 0 & 2 \end{bmatrix}$$

Transposta

$$\begin{bmatrix} 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 \\ 2 & 2 & 2 & 2 \end{bmatrix}$$