

IP - Aula 21/11/2023

Matrizes II

Hebert Coelho

Instituto de Informática
Universidade Federal de Goiás

Roteiro

- Matrizes de Strings
- Inicialização de Matrizes
- Parâmetros da função main

Matrizes de Strings

- Utiliza-se uma matriz bidimensional de caracteres;
- O tamanho do índice esquerdo indica o número de strings;
- O tamanho do índice direito indica o comprimento máximo de cada string.

Exemplo:

```
char nomes[30][20];
```

Matrizes de Strings

- Para acessar uma string individual basta especificar o índice esquerdo

Exemplo:

m	a	r	i	a		a	l	i	c	e
p	e	d	r	o						
h	e	n	r	i	q	u	e			

```
puts(M[1]);  
ou  
printf("%s",M[1]);  
imprimirá a string pedro.
```

Matrizes de Strings

Inserindo informações em uma matriz de strings

```
gets(matriz[índice]);  
ou  
scanf("%s",matriz[índice]);
```

Matrizes de Strings

Exercício:

Faça um programa para ler o nome e a idade de $N \leq 10$ pessoas e imprima o nome da pessoa mais velha. O número N é fornecido pelo usuário e cada pessoa tem nome com no máximo 20 caracteres.

Inicialização de Matrizes

- Pode-se inicializar matrizes no momento da declaração.

Sintaxe:

```
tipo nome_da_matriz[tamanho1]...[tamanhoN] =  
{lista_valores};
```

Exemplo:

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

Inicialização de Matrizes de Strings

- Pode-se inicializar matrizes de strings de forma abreviada.

Sintaxe:

```
char nome_da_matriz[tamanho] = "string";
```

Exemplo:

```
char str[14] = "Eu gosto de C";
```

Exemplo:

```
char str[14] = {'E', 'u', ' ', 'g', 'o', 's', 't', 'o', ' ',  
'd', 'e', ' ', 'C', '\0'};
```


Inicialização de Matrizes Bidimensionais

Exemplo:

```
int str[10][2]  
={{1,1},{2,4},{3,9},{4,16},{5,25},{6,36},  
{7,49},{8,64},{9,81},{10,100}};
```

Os Argumentos argc e argv

A função 'main()' pode, de fato, aceitar parâmetros formais, contudo, é importante observar que a escolha desses parâmetros não está sob o controle direto do programador. A declaração mais abrangente para a função main() é a seguinte:

Exemplo:

```
int main (int argc, char *argv[]);
```

Os Argumentos argc e argv

- Os parâmetros 'argc' e 'argv' fornecem ao programador acesso à linha de comando utilizada para chamar o programa.
- O 'argc' (argument count) é um valor inteiro que indica o número de argumentos pelos quais a função main() foi invocada na linha de comando.
- O argv (argument values) é um vetor de strings que contém os parâmetros específicos passados pela linha de comando. Cada string dentro desse vetor representa um dos argumentos fornecidos durante a chamada do programa. O valor de argc indica o número total de argumentos na linha de comando, e cada elemento de argv representa um desses argumentos.

Exercício 1 sobre argc e argv

Escreva um programa que faz uso dos parâmetros 'argv' e 'argc'. O programa recebe da linha de comando o dia, mês e ano correntes, e imprime a data no formato indicado no exemplo. Veja o exemplo, supondo que o executável se chame "data" e seja dado a seguinte linha de comando para execução no linux:

```
./data 19 04 06
```

O programa imprimirá:

```
19 de abril de 2006
```

Resposta para o exercício

```
#include <stdio.h>
int main(int argc, char *argv[]){
    int mes;
    char *nomemes[] = {"Janeiro", "Fevereiro", "Março",
        "Abril", "Maio", "Junho", "Julho", "Agosto", "Setembro",
        "Outubro", "Novembro", "Dezembro"};
    if (argc == 4){
        mes = atoi(argv[2]);
        if (mes < 1 || mes > 12) printf("Erro!\nMes invalido!")
        else
            printf("\n%s de %s de 20%s", argv[1],
                nomemes[mes - 1], argv[3]);    }
    else
        printf("Erro!\nUso: data dia mes ano, todos inteiros");
}
```

Exercício 2 sobre argc e argv

Desenvolva um programa em linguagem C que aceite como entrada, através da linha de comando durante a execução, um número natural e exiba o seu fatorial na saída padrão.

Exercício 3 sobre argc e argv

Desenvolva uma calculadora de operações matemáticas básicas. Utilize a linha de comando para receber dois operandos e o tipo de operação a ser realizada. Os operandos e a operação devem ser passados como argumentos da linha de comando (utilize argc e argv). O programa deve exibir o resultado na saída padrão. O programa deve suportar as operações de adição (+), subtração (-), multiplicação (*), e divisão (/). Além disso, deve validar a entrada do usuário, exibindo uma mensagem de erro caso a quantidade de argumentos seja inadequada ou caso os operandos não sejam números válidos. Considere a possibilidade de divisão por zero e trate-a adequadamente. Inclua mensagens informativas para orientar o usuário sobre o formato correto de entrada.

Exemplo:

```
./calculadora 5 + 3
```

```
Resultado: 8
```

Mini projeto

- Escolha os grupos para implementação do mini projeto 2.
- Escolha os temas.