Observações da última aula

- Ponto e vírgula
- Extensão do arquivo
- Simular o compilador
  - Anotando o valor das variáveis
- Identação

Comandos de Repetição
Parte 3

Prof. Debora Medeiros

Baseado no material de:
Ciro Trindade (Unisantos)

O comando *break*

- o comando *break* permite interromper *a* execução de um laço.

```c
#include <stdio.h>
int main () {
    int i, j;

    for (i = 0; i < 4; i++){
        for (j = 0; j < 2; j++){
            if (i == 1){
               break;
            }
            else{
                printf("i: %d j: %d\n", i, j);
            }
        }
    }
    system("pause");
    return 0;
}
```

```
i: 0 j: 0
i: 0 j: 1
i: 2 j: 0
i: 2 j: 1
i: 3 j: 0
i: 3 j: 1
Press any key to continue . . .
```


O comando *continue*

- o comando *continue* faz com que o loop pule para a próxima iteração. Os comandos que sucedem continue no bloco não são executados

```c
#include <stdio.h>
int main() {
    int i;
    for (i = 0; i < 5; i++){
        if (i == 1){
            continue;
        }
        else printf("i: %d \n", i);
    }
    system("pause");
    return 0;
}
```

```
i: 0
i: 2
i: 3
i: 4
Press any key to continue . . .
```


Vetores e Matrizes

Prof. Debora Medeiros

Baseado no material de:
Ciro Trindade (Unisantos)

Introdução

- Um vetor é uma coleção de variáveis do mesmo tipo que são referenciadas por um nome comum
- Um elemento específico em um vetor é acessado através de um índice

Vetores

- A forma geral de um vetor é:

```
tipo nome-da-variável[tamanho];
```

- **tipo**: declara o tipo base do vetor
- **tamanho**: define quantos elementos o vetor conterá
- Todos os vetores têm 0 como o índice do 1º elemento
- Exemplo: `int p[10];`
- declara um vetor de inteiros que tem 10 elementos, p[0] a p[9]

Exemplo

```c
#include <stdio.h>
int main() {
    int x[10];
    int i;
    for(i = 0; i < 10; i++) {
        x[i] = i;
    }
    for(i = 0; i < 10; i++) {
        printf("%d  ",x[i]);
    }
    printf("\n");
    system("pause");
    return 0;
}
```


Exemplo

```c
#include <stdio.h>
int main() {
    int x[10];
    int i;
    for(i = 0; i < 10; i++) {
        x[i] = i;
    }
    for(i = 0; i < 10; i++) {
        printf("%d  ",x[i]);
    }
    printf("\n");
    system("pause");
    return 0;
}
```

```
0  1  2  3  4  5  6  7  8  9
Press any key to continue . . .
```


Vetores

- Em C, não é feita a verificação do tamanho do vetor
- É responsabilidade do programador incluir a verificação dos limites do vetor quando isso for necessário
- Um vetor pode armazenar qualquer tipo de dado

Inicializando Vetores

- Você pode inicializar vetores na mesma instrução de sua declaração

```
int tab[LIM]={100,50,20,10,5,2,1};
```

- A lista de valores é colocada entre chaves e os valores são separados por vírgulas
- Os valores são atribuídos na sequência
- Se nenhum número for fornecido para dimensionar o vetor, o compilador contará o número de itens da lista de inicialização e o fixará como dimensão do vetor

```
int tab[]={100,50,20,10,5,2,1};
```

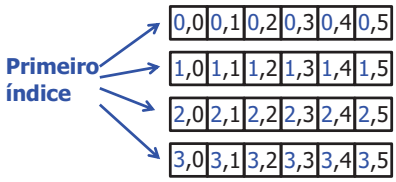

Matrizes

- A linguagem C permite matrizes de qualquer tipo, incluindo matrizes com mais de duas dimensões
- Com 2 pares de colchetes obtemos uma matriz de 2 dimensões e p/ cada par de colchetes adicionais obtemos uma matriz com uma dimensão a mais:

```
tipo nome-da-variável[tamanho 1][tamanho 2]... [tamanho n];
```

Matrizes

- Exemplo de uma matriz 4x6



Matrizes

- Exemplo de uma matriz 4x6


```
 1: #include <stdio.h>
 2: int main() {
 3:     char matriz[5][10];
 4:     int i,j,LIN=5,COL=10;
 5:     for(i = 0; i < LIN; i++) {
 6:         for(j = 0; j < COL; j++) {
 7:             matriz[i][j]='.';
 8:             printf("%c",matriz[i][j]);
 9:         }
10:         printf("\n");
11:     }
12:     printf("Digite coordenadas na forma (linha,coluna)\n");
13:     printf("Use um número negativo para sair.\n");
14:     printf("Coordenadas (linha,coluna): ");
15:     scanf("%d,%d",&i,&j);
16:     while(i >= 0) {
17:         matriz[i][j] = '*';
18:         for(i = 0; i < LIN; i++) {
19:             for(j = 0; j < COL; j++) {
20:                 printf("%c",matriz[i][j]);
21:             }
22:             printf("\n");
23:         }
24:         printf("Coordenadas (linha,coluna): ");
25:         scanf("%d,%d",&i,&j);
26:     }
27:     system("pause");
28:     return 0;
29: }
```


Exemplo

```
..........
..........
..........
..........
..........
Digite coordenadas na forma (linha,coluna)
Use um n·mero negativo para sair.
Coordenadas (linha,coluna): 2,3
..........
..........
...*......
..........
..........
Coordenadas (linha,coluna): -1
Press any key to continue . . .
```


Inicializando matrizes

- As matrizes são inicializadas da mesma maneira que os vetores

```
char inimigo[LIN][COL]={
    {0,0,0,0,0,0,0,0,0,0},
    {0,1,1,1,1,0,0,1,0,1},
    {0,0,0,0,0,0,0,1,0,1},
    {1,0,0,0,0,0,0,1,0,0},
    {1,0,1,1,1,0,0,0,0,0} };
```

- Uma matriz pode ser vista como um vetor onde seus elementos são vetores