MC-102 — Aula 09
Laços Encaixados e Matrizes

Instituto de Computação – Unicamp

28 de Março de 2012

Roteiro

Laços Encaixados

- Muitas vezes é necessário implementar um laço dentro de outro laço.
- Estes são laços encaixados.

```
int i,j;

for(i=1;i<=10;i++){
  for(j=1;j<=5;j++){
     printf("\n i:%d j:%d",i,j);
  }
}
```

Laços Encaixados

- Já sabemos testar se um determinado número é ou não primo.
- Imagine que agora queremos imprimir os n primeiros números primos.
- O que podemos fazer?

Laços Encaixados

- O programa abaixo verifica se valor na variável "candidato" corresponde a um primo:

```
divisor = 2;
eprimo = 1;
while( (divisor <= candidato/2) && (eprimo) ){
  if(candidato % divisor == 0)
     eprimo = 0;
  divisor++;
}
if(eprimo){
  printf("%d, ", candidato);
}
```

Laços Encaixados

Podemos usar o trecho de código anterior para imprimir os *n* primeiros números primos:

```
int main(){
  int divisor, candidato, primosImpressos, n, eprimo;
  printf("\n Digite um numero inteiro positivo:");
  scanf("%d",&n);

  candidato = 2;
  primosImpressos = 0;
  while(primosImpressos < n){

    //trecho do código anterior que
    //checa se candidato é ou não é primo

    if(eprimo){
      printf("%d, ", candidato);
      primosImpressos++;
    }
    candidato++;//Testa proximo numero candidato
  }
}
```

Laços Encaixados

Código completo:

```
int main(){
  int divisor, candidato, primosImpressos, n, eprimo;
  printf("\n Digite um numero inteiro positivo:");
  scanf("%d",&n);
  candidato = 2;
  primosImpressos = 0;
  while(primosImpressos < n){
    divisor = 2;
    eprimo=1;
    while( (divisor <= candidato/2) && (eprimo) ){
      if(candidato % divisor == 0)
         eprimo = 0;
      divisor++;
    }
    if(eprimo){
      printf("%d, ", candidato); primosImpressos++;
    }
    candidato++;//Testa proximo numero candidato
  }
}
```

Laços Encaixados

- Note que o número 2 é o único número par que é primo.
- Podemos alterar o programa para sempre imprimir o número 2:

```
int main(){
  int divisor, candidato, primosImpressos, n, eprimo;

  printf("\n Digite um numero inteiro positivo:");
  scanf("%d",&n);

  if(n > 0){
    printf("%d, ", 2);
     .....
  }
```

Laços Encaixados

- Podemos alterar o programa para testar apenas números ímpares depois:

```
candidato = 3;
primosImpressos = 1;
while(primosImpressos < n){
  divisor = 2;
  eprimo=1;
  while( (divisor <= candidato/2) && (eprimo) ){
     if(candidato % divisor == 0)
          eprimo = 0;
     divisor++;
  }
  if(eprimo){
     printf("%d, ", candidato);
     primosImpressos++;
  }
  candidato = candidato + 2;//Testa proximo numero candidato
}
```

Laços Encaixados

```
int main(){
  int divisor, candidato, primosImpressos, n, eprimo;
  printf("\n Digite um numero inteiro positivo:");
  scanf("%d",&n);
  if(n > 0){
    printf("%d, ", 2);
    candidato = 3; primosImpressos = 1;
    while(primosImpressos < n){
      divisor = 2;  eprimo=1;
      while( (divisor <= candidato/2) && (eprimo) ){
         if(candidato % divisor == 0)
              eprimo = 0;
         divisor++;
      }
      if(eprimo){
         printf("%d, ", candidato); primosImpressos++;
      }
      candidato = candidato + 2;//Testa proximo numero candidato
    }
  }
}
```

Laços Encaixados

- Suponha que queremos imprimir todas as possibilidades de resultados ao se jogar 4 dados de 6 faces.
- Para cada possibilidade do primeiro dado, devemos imprimir todas as possibilidades dos 3 dados restantes.
- Para cada possibilidade do primeiro e segundo dado, devemos imprimir todas as possibilidades dos 2 dados restantes....
- Você consegue pensar em uma solução com laços aninhados?

Laços Encaixados

```
int main(){
  int d1, d2, d3, d4;

  printf("\nD1  D2  D3  D4\n");
  for(d1 = 1; d1 <= 6; d1++)
    for(d2 = 1; d2 <= 6; d2++)
      for(d3 = 1; d3 <= 6; d3++)
        for(d4 = 1; d4 <= 6; d4++)
           printf("%d   %d   %d   %d\n",d1,d2,d3,d4);

}
```

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 de 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* × *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 2^a linha e na 11^a 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

$$\texttt{<tipo>}\ \texttt{nome\_da\_matriz}\ [<\ dim_1\ >]\ [<\ dim_2\ >]\ \ldots\ [<\ dim_N\ >]$$

- Essa matriz possui $dim_1 \times dim_2 \times \cdots \times \dim_N$ 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 3×3 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");
  }
}
```

Exercícios

Escreva um programa que leia todas as posições de uma matriz 10×10. Em seguida, mostra o índice da linha e o índice da coluna e o valor das posições não nulas. No final, exibe o número de posições não nulas.

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

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