Notas de Aula
Exemplos em C

Routo Terada

www.ime.usp.br/~rt

Depto. C. da Computação - USP

Programa com inteiro

```
#include <stdio.h> // biblioteca standard de Input/Output
main(){
        int j; // declara var. j inteira
        printf("Digite um valor inteiro e ENTER\n"); // mostra na tela
        scanf("%d", &j); // ler j, NAO digitar NADA entre %d e "
        printf("Valor de j e' %d\n", j); // mostra na tela valor de j lido
}  // fim main
```

Programa com char

```
#include <stdio.h> // biblioteca standard de Input/Output
main(){
        char car; // declara var. car do tipo char
        printf("Digite um caractere e ENTER\n"); // mostra na tela
        scanf("%c", &car);  // %c para ler ou mostrar char 1 char
        printf("Caractere digitado e' %c\n", car);
        // mostrou na tela valor de car lido
}  // fim main
```

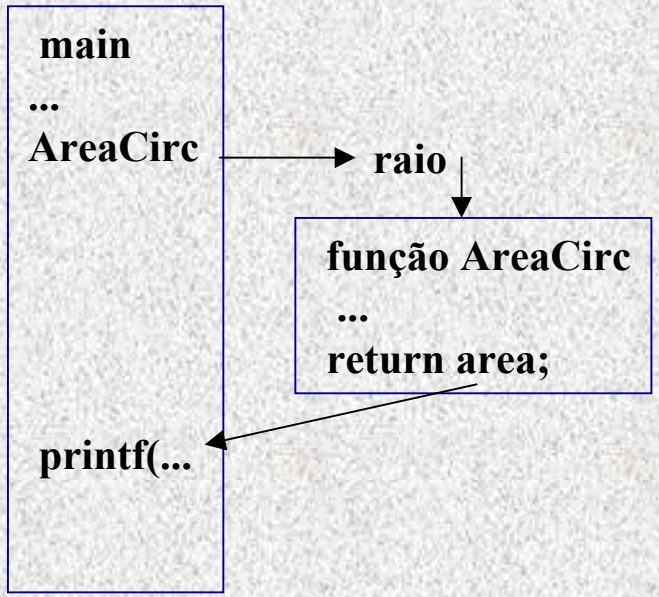
```
Digite um caractere e ENTER
 ABC
Caractere digitado e' A
```

Programa com float

```
// Programa p/ calcular area, dado o raio
// Verifique se no compilador a "Option/Use floating point library"
//                esta' LIGADA
#include <stdio.h>  // biblioteca standard de Input/Output
#define PI 3.14159  // constante pi
// a seguir a funcao main()
main(){
        float raio, area; // duas vars. declaradas

        // mostra na tela uma mensagem com printf
        printf("Digite o valor do raio do circulo, e ENTER\n");

        // le do teclado com scanf; note o & em &raio
        scanf("%f", &raio); // exemplos: 1.23  12.3e-1

        // a seguir mostra na tela o valor do raio lido
        // usando 12 colunas, e 2 decimais arredondados
        // por ex. 1.576 e' arredondado para 1.58, so' na tela
        printf("Raio digitado e' %12.2f\n", raio);

        area= PI*raio*raio;

        printf("Area do circulo com raio %f e' %f\n", raio, area);
}// fim main
```

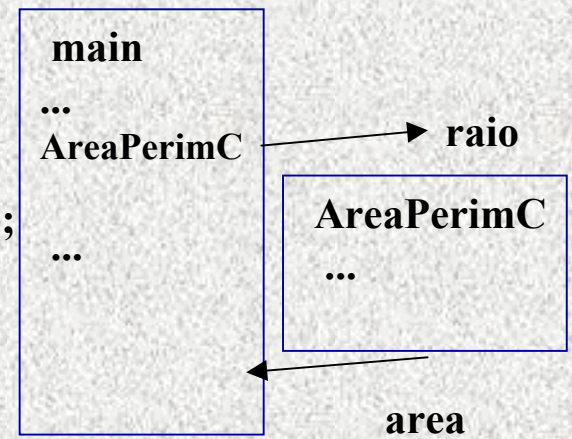
Programa com função

```
// Funcao p/ calcular area, dado o raio
// Verifique se no compilador a "Option/Use floating point library"
//                esta' LIGADA
#include <stdio.h>  // biblioteca standard de I/O
#define PI 3.14159  // constante pi
// a seguir a funcao AreaCirc
float AreaCirc(float R){ // cabecalho da funcao, parametro R
                float area; // variavel so'  existe dentro da funcao, local
                area=PI*R*R;
                return area; // retorna para main o valor area, do tipo float (único valor)
} // fim da funcao AreaCirc

// a seguir a funcao main()
main(){
                float raio; // declaracao
                // mostra na tela uma mensagem com printf
                printf("Digite o valor do raio do circulo, e ENTER\n");

                // le do teclado com scanf; note o & em &raio
                scanf("%f", &raio); // exemplos: 1.23  12.3e-1

                // a seguir mostra na tela o valor do raio lido
                // usando 12 colunas, e 2 decimais arredondados
                // por ex. 1.576 e' arredondado para 1.58, so' na tela
                printf("Raio digitado e' %12.2f\n", raio);
                                // a seguir, main recebe de AreaCirc o valor da area
                printf("Area do circulo com raio %f e' %f\n", raio, AreaCirc(raio)); // chamada da funcao
}// fim main
```



```
// Funcao p/ calcular area e perimetro, dado o raio
// Verifique se no compilador a "Option/Use floating point library"
//                esta' LIGADA
#include <stdio.h>  // biblioteca standard de I/O
#define PI 3.14159  // constante pi
// a seguir a funcao AreaPerimCirc
void AreaPerimCirc(float R, float *area, float *perimetro){ // duas saídas
            *area=PI*R*R;  // note * em *area e *perimetro
            *perimetro=2.0*PI*R;
            // sem return
} // fim da funcao AreaPerimCirc
main(){float raio,area,perimetro; // declaracao
            // mostra na tela uma mensagem com printf
            printf("Digite o valor do raio do circulo, e ENTER\n");
            // le do teclado com scanf; note o & em &raio
            scanf("%f", &raio); // exemplos: 1.23  12.3e-1
            // a seguir mostra na tela o valor do raio lido
            printf("Raio digitado e' %12.2f\n", raio);

            AreaPerimCirc(raio,&area,&perimetro);
            printf("Area do circulo com raio %f e' %f, e o perimetro e' %f\n",
                        raio, area, perimetro);
}// fim main
```

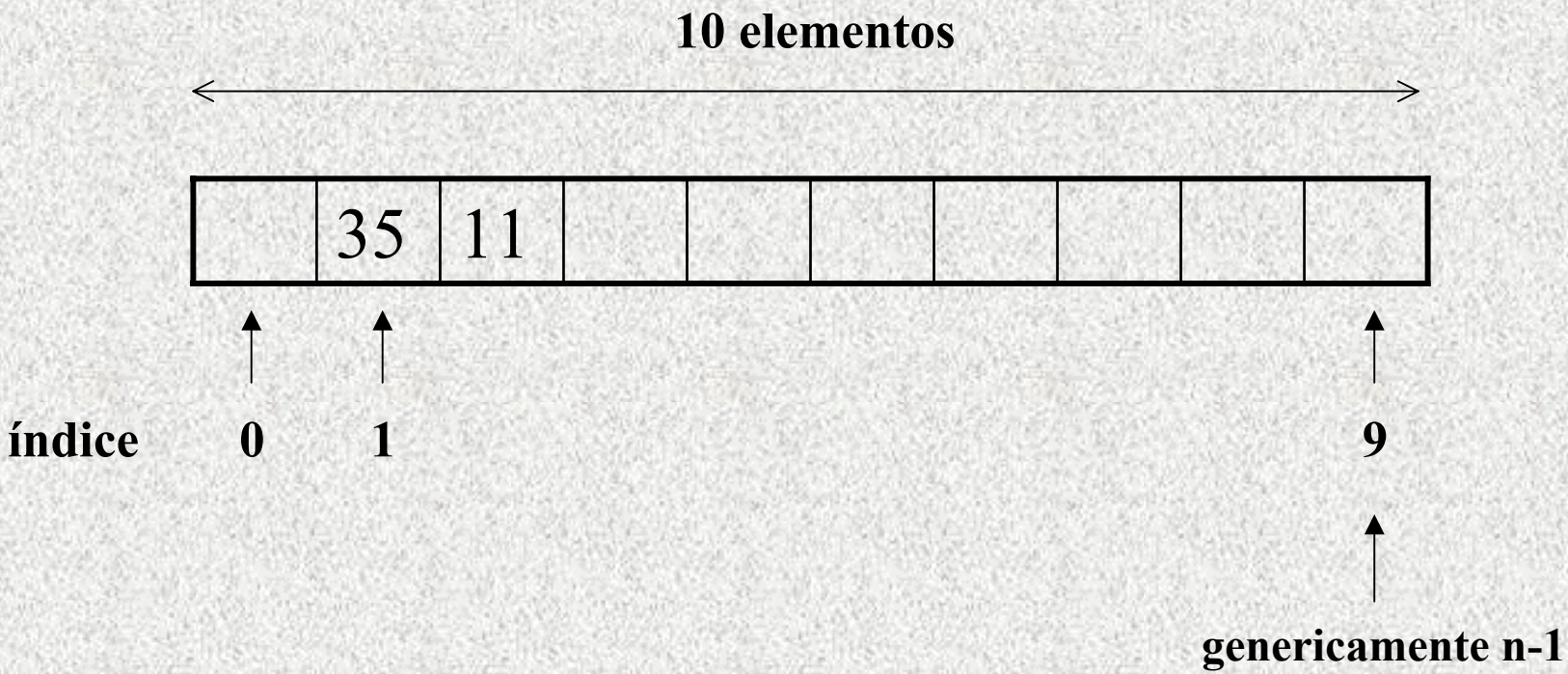
Programa com função com mais de 1 saída



duas saídas

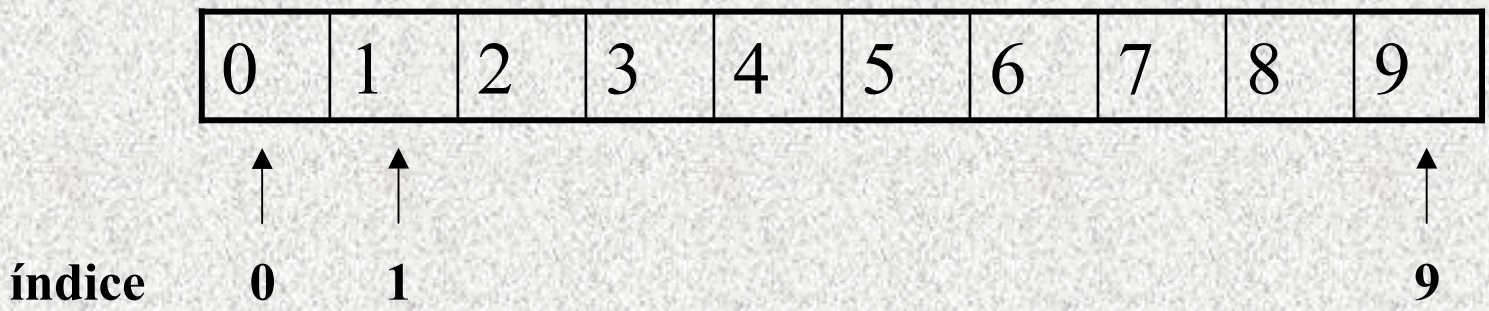
Vetor: seqüência linear de dados do mesmo tipo

```
int V[10];  // declaração de vetor V de int
V[1]=35;   // referência a um elemento
V[2]=11;
```

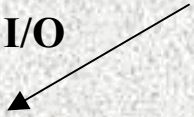


Percorrimento de um vetor:

```
conta = 0;
// índice j = 0,1,2,…9
for (j=0; j<10; j++) { // abre
    V[j] = conta++; // atribui e soma 1
    } // fecha
```



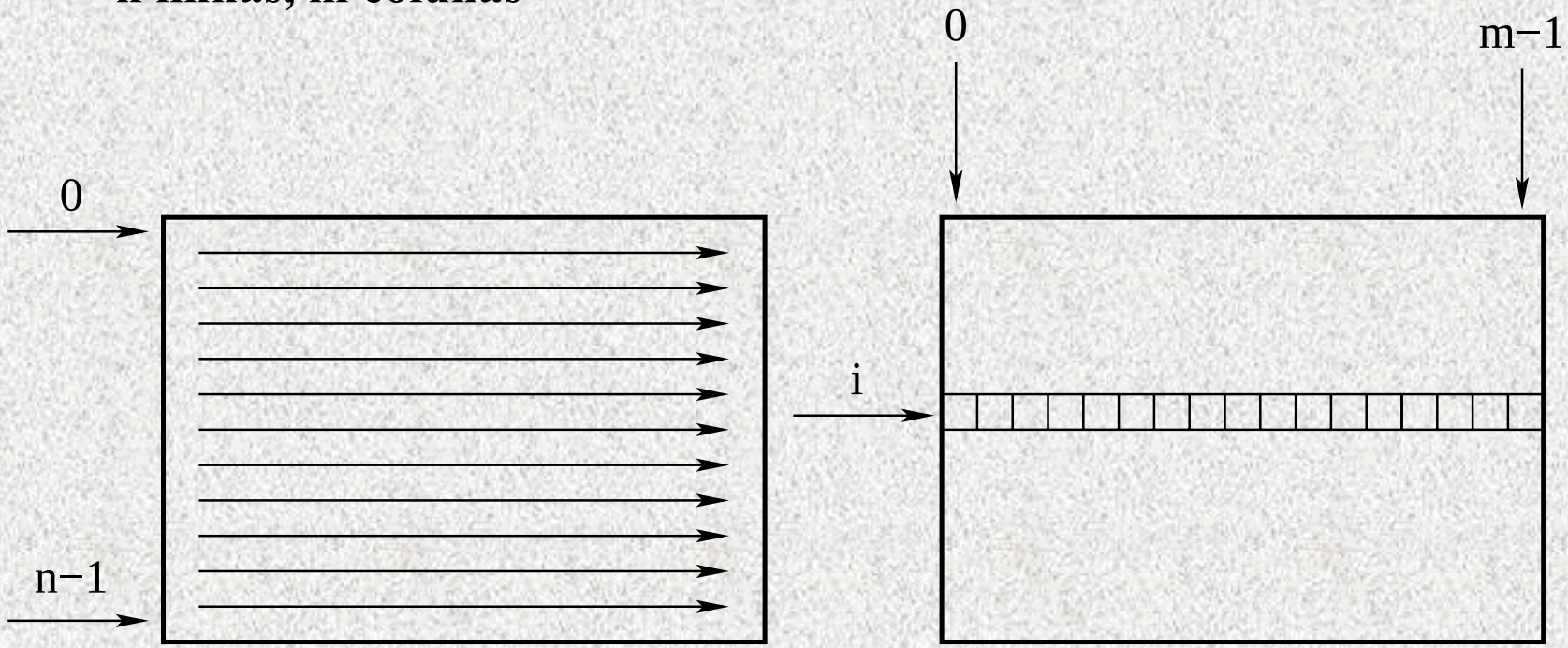
Função com vetor como parâmetro



```
// Funcao p/ copiar vetor
#include <stdio.h>  // biblioteca standard de I/O
// a seguir a funcao CopiaVetor
void CopiaVetor(int n, int Vorig[ ], int Vdest[ ]){ // dois vetores
   int j; // j=0,1,2,…n-1
   for(j=0,j<n;j++){ // j=0,1,2,..n-1
      Vdest[j]=Vorig[j];
      } // fim for j
   // sem return
} // fim da funcao CopiaVetor

// a seguir a funcao main()
main(){
   int j,conta, Vorig[100],Vdest[100]; // declarações
   conta=0;
   for(j=0;j<100;j++){ // inicializa vetor
      Vorig[j]=conta++;
      } // fim for j
   CopiaVetor(100,Vorig,Vdest); // chama função
   for(j=0;j<100;j++){
      printf("%d ", Vorig[j],Vdest[j]);
   } // fim for j
}// fim main
```

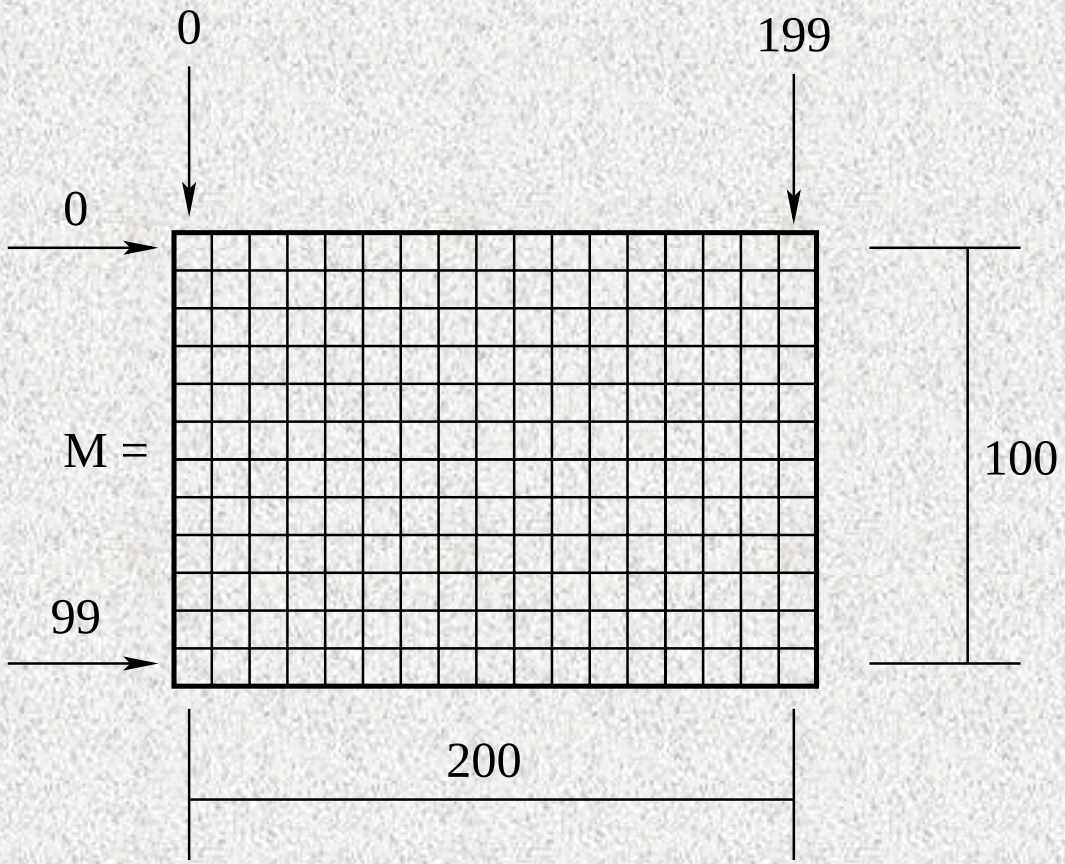
Matriz: grupo retangular de dados do mesmo tipo

n linhas, m colunas

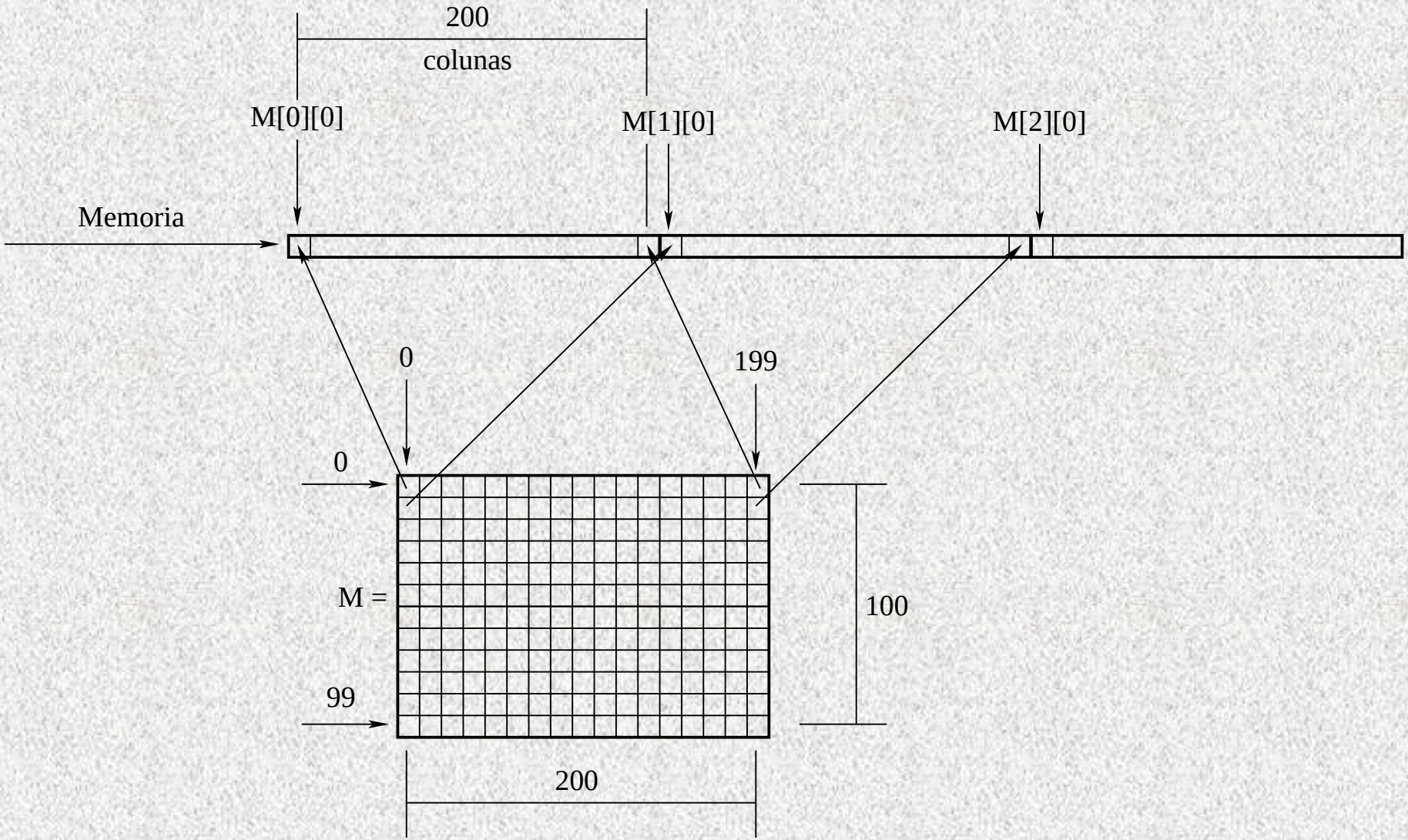


Matriz: grupo retangular de dados do mesmo tipo

int M[100][200]; // declaração da matriz M de 2 índices
a = M[2][3]; a = M[i][j]; // referência a elemento da matriz

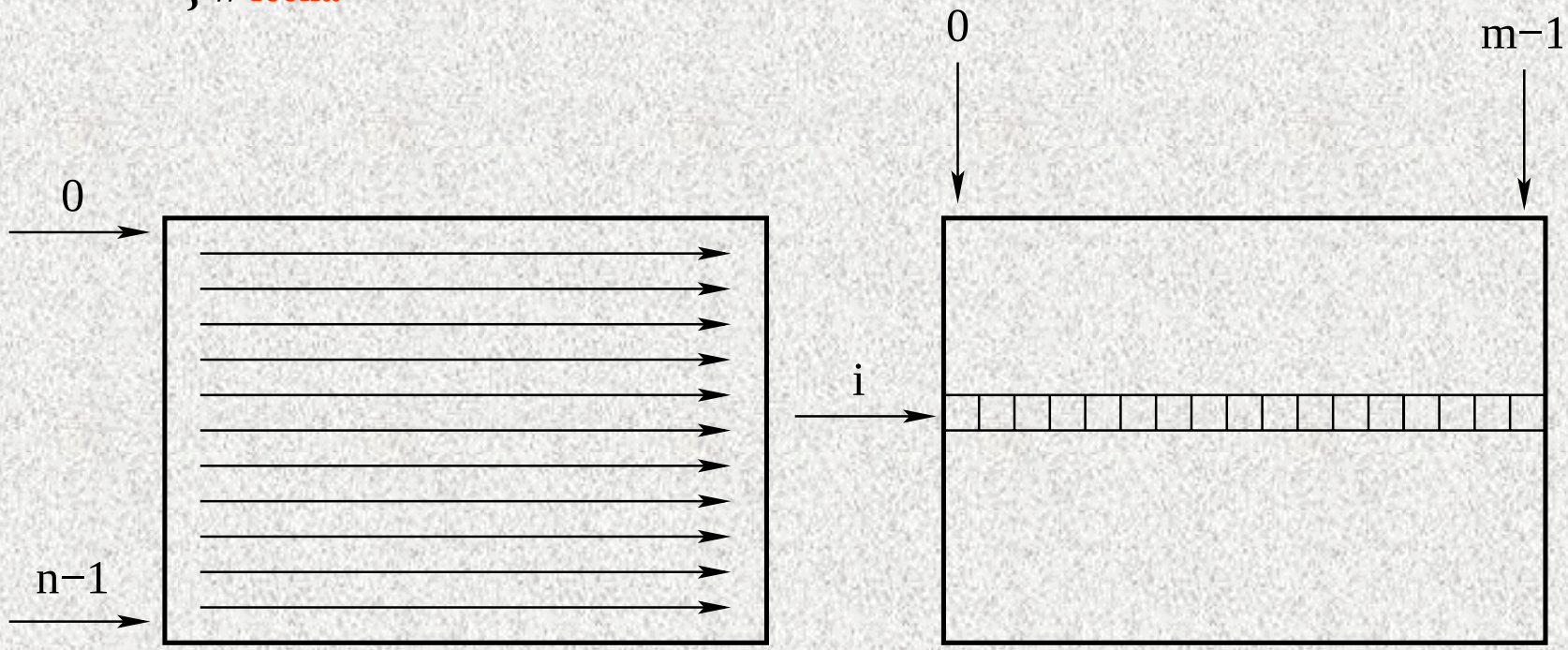


Armazenamento na memória: elementos de uma linha em posições consecutivas



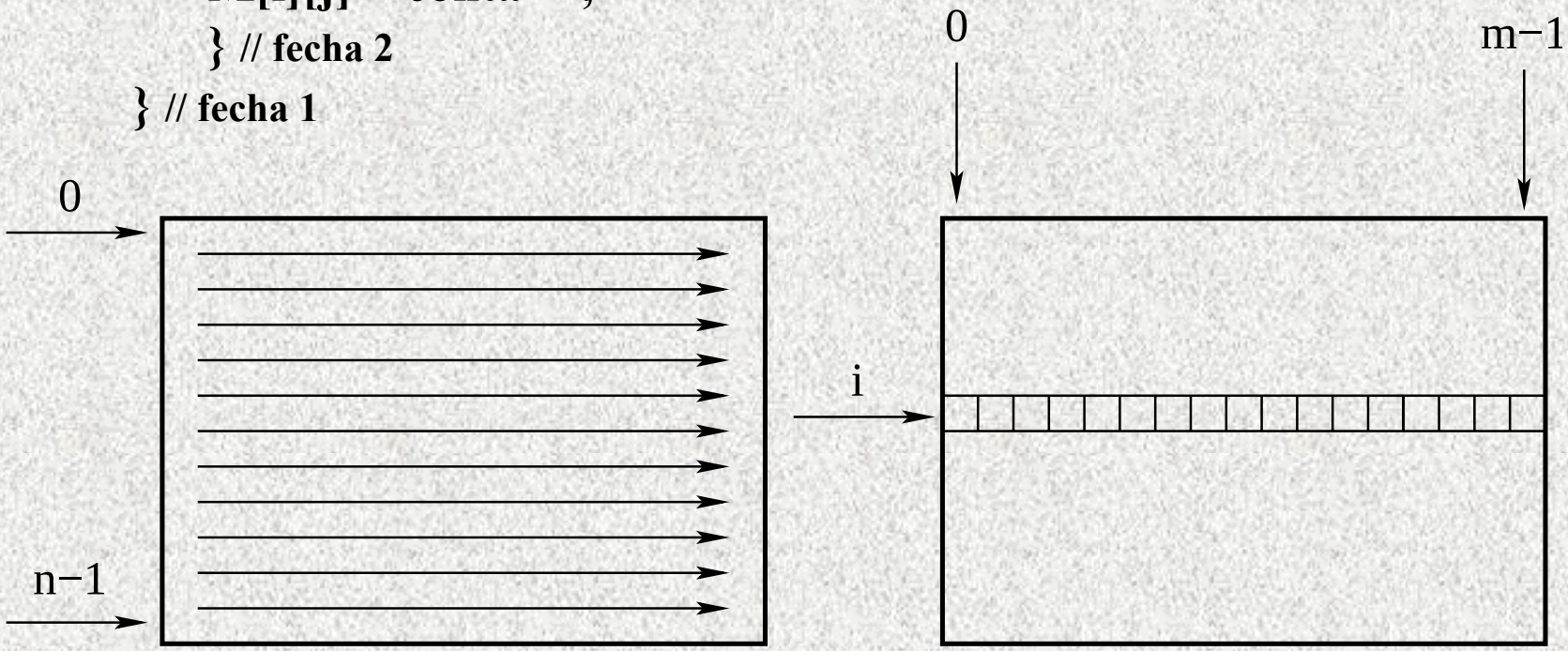
Percorrimento de uma linha:

```
conta = 0; // supor índice i definida
/* para uma linha fixa i */
for (j=0; j<m; j++) { // abre
    M[i][j] = conta++;
    } // fecha
```



Percorrimento linha por linha:

```
conta= 0;
/* define uma linha fixa i */
 for(i=0;i<n;i++){ // abre 1
    for (j=0; j<m; j++) { // abre 2
       M[i][j] = conta++;
       } // fecha 2
 } // fecha 1
```



Leitura: elemento a elemento

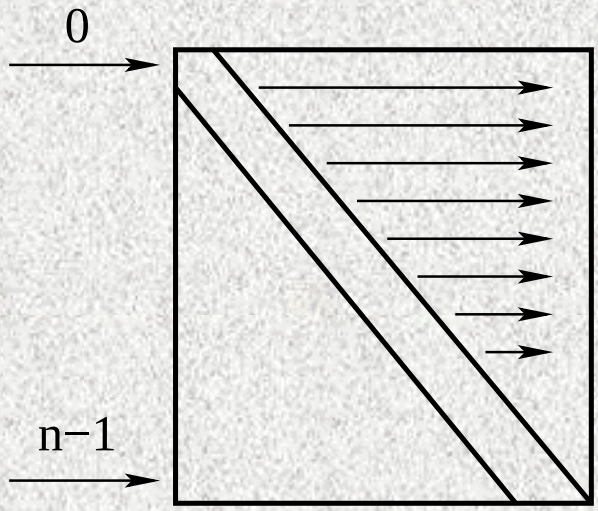
```
/* supor feita a leitura de n e m */
for (i=0; i<n; i++){
  for (j=0 j<m; j++) {
    printf (“Digite M[%d][%d]: ", i, j);
    scanf ("%d", &M[i][j]);
    } // fim for j
} // fim for i
```

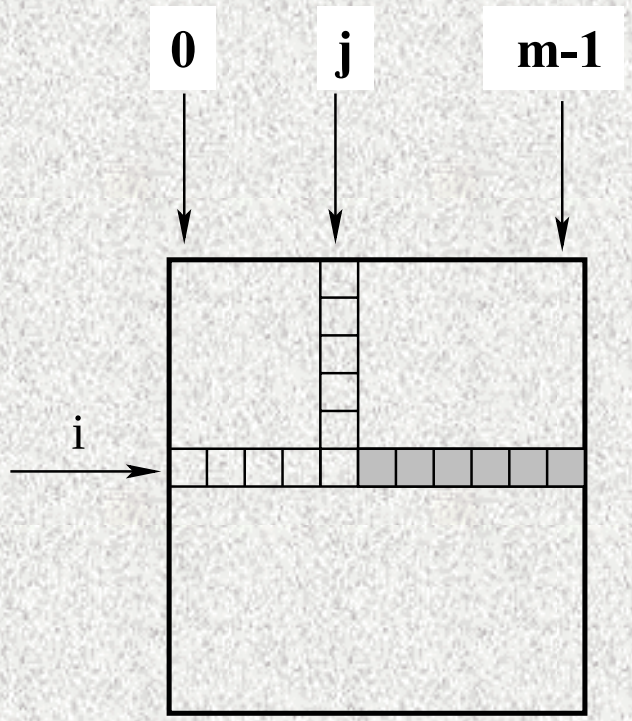
Mostra na tela: elemento a elemento

```
for (i=0; i<n; i++) {
  for (j=0 j<m; j++) {
     printf ("%d ", M[i][j]);
     } // fim for j
 printf ("\n"); // linha nova
 } // fim for i
```

Percorrer parte superior de uma linha

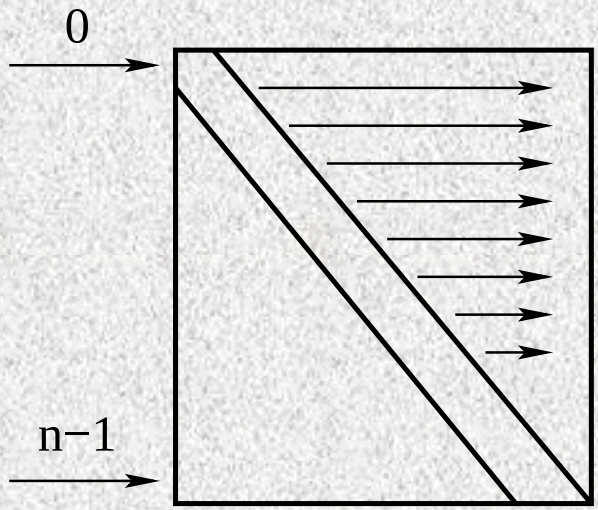
```
conta = 0;
/* para uma linha fixa i */
for (j=i+1; j<m; j++) {
      M[i][j] = conta++;
 } // fim for j
```

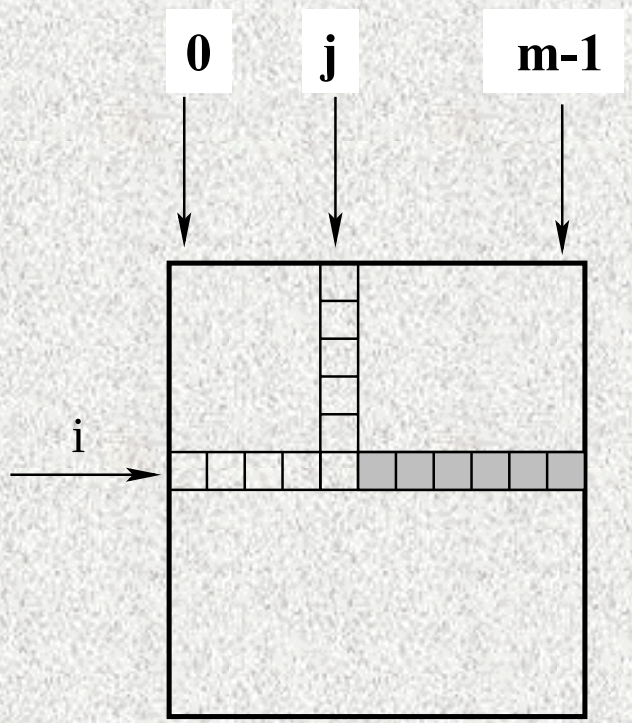




Percorrer parte superior da matriz, por linha

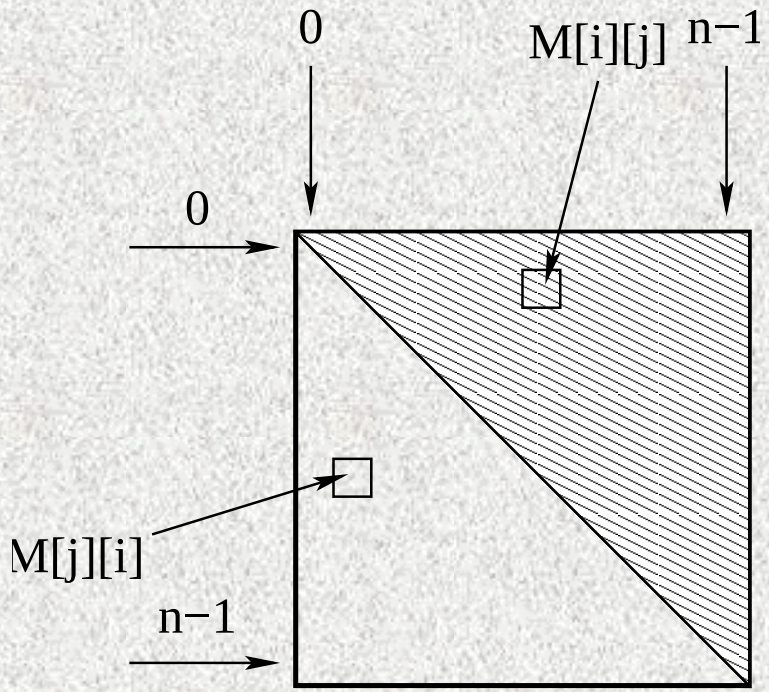
```
conta = 0;
for (i=0; i<n; ++) {
  for (j=i+1; j<m; j++) {
     M[i][j] = conta++;
     } // for j
 } // for i
```



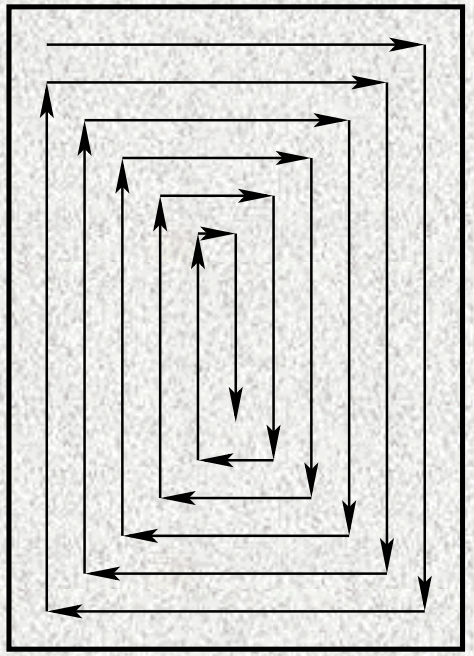


Uma matriz M, n por m, é simétrica sse M=[M transposta].

```
simetrica = 1; // indicador de passagem
for (i=0; i<n; i++) {
    for (j=i+1; j<m; j++){
        if (M[i][j] != M[j][i]) {
            simetrica = 0;
        } // fim if
    } // fim for j
} // fim for i
```



Outras formas de percorrimento



Função com Matriz

```
# include <stdio.h>
 int LUA (int A[ ][100]) {
   A[2][3] = 4;
   return 0;
   } // fim LUA

 main () {
   int M[100][100], a;
   M[2][3] = 5;
   a = LUA (M);
   } // fim main
```

Multiplicação de matriz A por vetor V.

Cálculo de um só elemento C[i]

```
C[i] = 0;
for (k=0; k<p; k++){ // 1
    C[i] += A[i][k] * V[k];
    } // 1
```

i é índice de linha
k é índice de coluna

índ.	0	1	2	3
0	A[0][0]			
1				
2				

0	V[0]
1	V[1]
2	V[2]
3	V[3]

Multiplicação de matriz A por vetor V.

Cálculo de todos os elementos C[i]

i é índice de linha

k é índice de coluna

```
C[i] = 0;
for (k=0; k<p; k++){ // 1
    C[i] += A[i][k] * V[k];
    } // 1
```

```
for (i=0; i<n; i++){ // 1
        C[i] = 0;
        for (k=0; k<p; k++){ // 2
            C[i] += A[i][k] * V[k];
            } // 2
     } // 1
```

índ.	0	1	2	3
0	A[0][0]			
1				
2				

0	V[0]
1	V[1]
2	V[2]
3	V[3]

Funções com Matriz

```
void copia (double A[ ][100], double B[ ][100], int n);
void soma  (double A[ ][100], double B[ ][100], int n);
void escalar (double A[ ][100], int n, double alpha);
void identidade (double A[ ][100], int n);
```

```
void copia (double A[ ][100], double B[ ][100], int n) {
    int i, j;
    for (i=0; i<n; i++)
      for (j=0; j<n; j++)
        A[i][j] = B[i][j];
}
```

Funções com Matriz

```
void copia (double A[ ][100], double B[ ][100], int n);
void soma  (double A[ ][100], double B[ ][100], int n);
void escalar (double A[ ][100], int n, double alpha);
void identidade (double A[ ][100], int n);
```

```
void soma  (double A[ ][100], double B[ ][100], int n) {
    int i, j;
    for (i=0; i<n; i++)
      for (j=0; j<n; j++)
        A[i][j] += B[i][j];
  }
```

Funções com Matriz

```
void copia (double A[ ][100], double B[ ][100], int n);
void soma  (double A[ ][100], double B[ ][100], int n);
void escalar (double A[ ][100], int n, double alpha);
void identidade (double A[ ][100], int n);
```

```
void escalar (double A[ ][100], int n, double alpha) {
    int i, j;
    for (i=0; i<n; i++)
      for (j=0; j<n; j++)
        A[i][j] *= alpha;
  }
```

Funções com Matriz

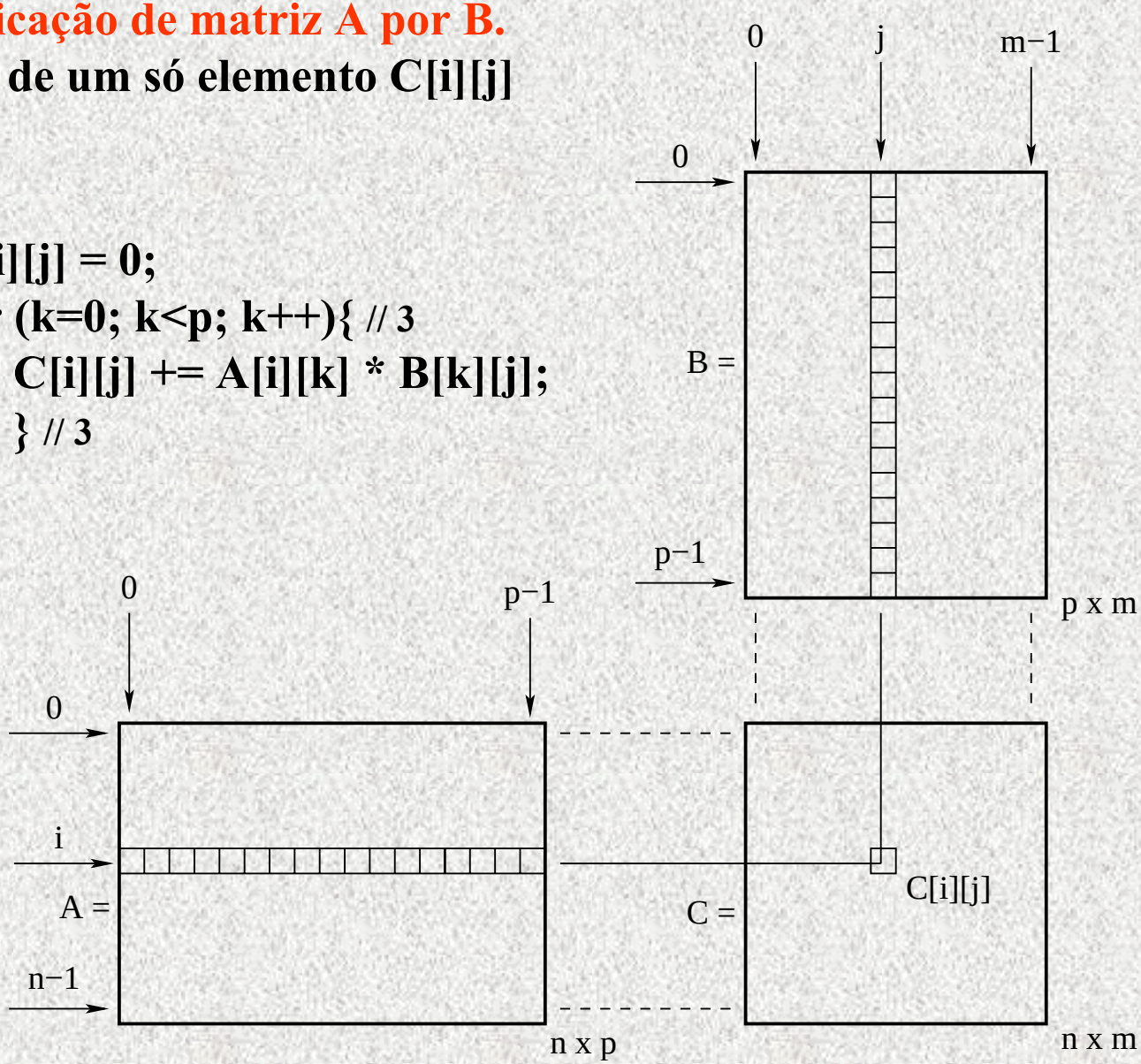
```
void copia (double A[ ][100], double B[ ][100], int n);
void soma  (double A[ ][100], double B[ ][100], int n);
void escalar(double A[ ][100], int n, double alpha);
void identidade (double A[ ][100], int n);
```

```
void identidade (double A[ ][100], int n) {
    int i, j;
    for (i=0; i<n; i++)
      for (j=0; j<n; j++)
        A[i][j] = (i == j); // 0 ou 1
  }
```

Multiplicação de matriz A por B.

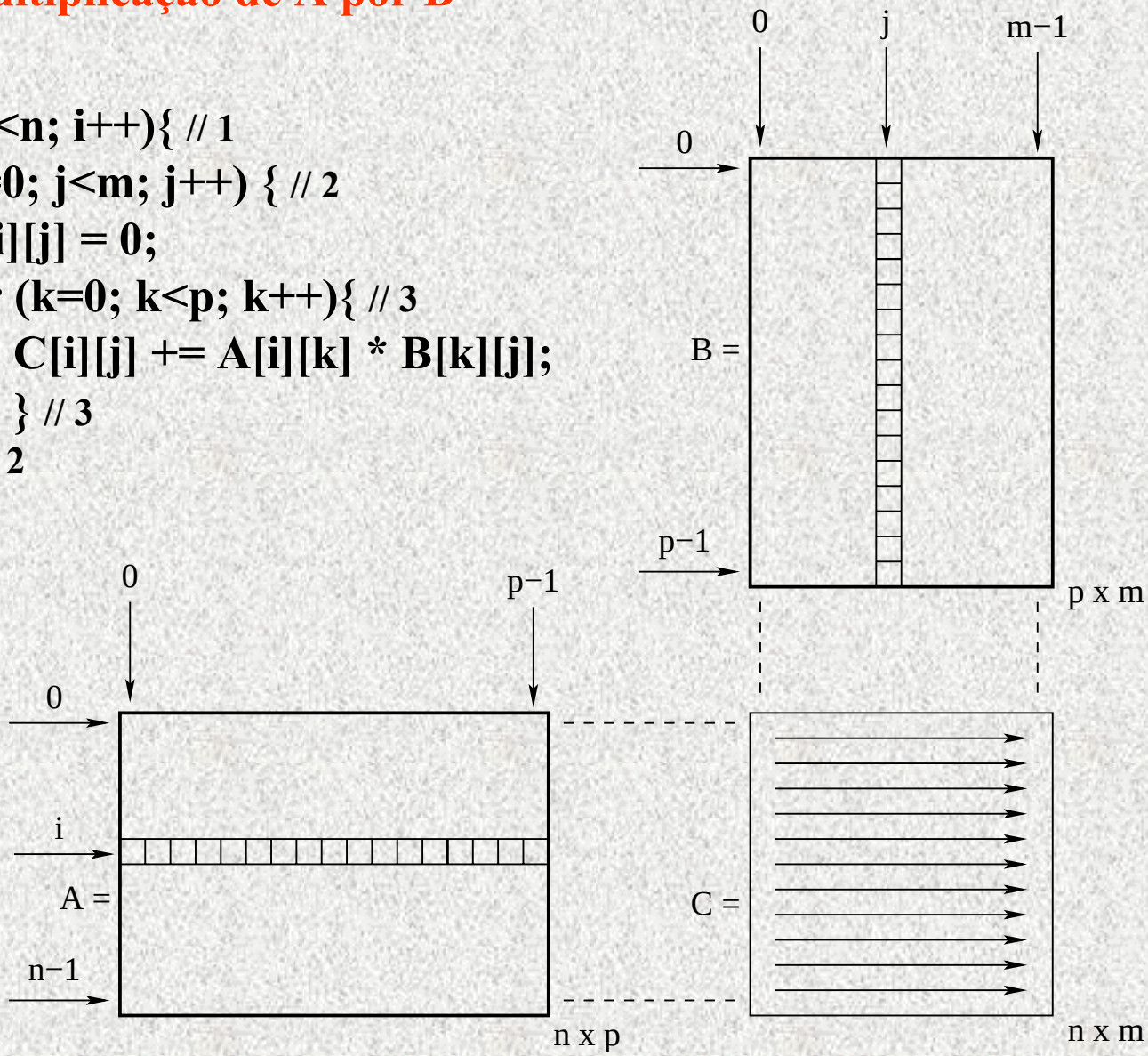
Cálculo de um só elemento C[i][j]

```
C[i][j] = 0;
for (k=0; k<p; k++){ // 3
    C[i][j] += A[i][k] * B[k][j];
    } // 3
```



Multiplicação de A por B

```
for (i=0; i<n; i++){ // 1
    for (j=0; j<m; j++) { // 2
        C[i][j] = 0;
        for (k=0; k<p; k++){ // 3
            C[i][j] += A[i][k] * B[k][j];
            } // 3
        } // 2
```



Multiplicação de A por B

```
void multi (double C[ ][100],
            double A[ ][100],
            double B[ ][100], int n) {
  int i, j, k;
  for (i=0; i<n; i++){ // 1
     for (j=0; j<m; j++) { // 2
          C[i][j] = 0;
          for (k=0; k<p; k++){ // 3
               C[i][j] += A[i][k] * B[k][j];
               } // 3
          } // 2
} // fim multi
```

Exponencial de matriz X, dado inteiro k>0

$$e^X = I + X + \frac{X^2}{2!} + \ldots + \frac{X^k}{k!}$$

```
void expm (double E[ ][100],
           double X[ ][100], int n, int k);
```

Exponencial de matriz X, dado inteiro k>0

Supor dadas as funções:

```
void copia (double A[ ][100], double B[ ][100], int n);
void soma  (double A[ ][100], double B[ ][100], int n);
void escalar(double A[ ][100], int n, double alpha);
void identidade(double A[ ][100], int n);
void multi (double C[ ][100],
      double A[ ][100],
      double B[ ][100], int n)
```

Utilizando estas funções, escrever:

```
void expm (double E[100][100],
            double X[100][100], int n, int k);
```

Exponencial de matriz X

$$e^X = I + X + \frac{X^2}{2!} + \ldots + \frac{X^k}{k!}$$

$$\text{Início}: E = I; T = I; i = 1$$

$$\text{enqto } i <= k \begin{cases} \text{Calcular o termo } T = \dfrac{X^i}{i!} \\ E = E_{\text{ant}} + T; \\ i++; \end{cases}$$

Exponencial de matriz X

$$e^X = I + X + \frac{X^2}{2!} + \ldots + \frac{X^k}{k!}$$

$$\text{Início}: E = I; T = I; i = 1$$

$$\text{enqto } i <= k \begin{cases} \text{Calcular o termo } T = \dfrac{X^i}{i!} = \dfrac{X^{i-1}}{(i-1)!} * \dfrac{X}{i} = T_{\text{ant}} * X * \dfrac{1}{i} \\ E = E_{\text{ant}} + T; \\ i++; \end{cases}$$

Exponencial de matriz X

$$\text{Início}: E = I; T = I; i = 1$$

$$\text{enqto } i <= k \begin{cases} \text{Calcular o termo } T = T_{\text{ant}} * X * \frac{1}{i} \\ E = E_{\text{ant}} + T; \\ i++; \end{cases}$$

$$\text{enqto } i <= k \begin{cases} B = T_{\text{ant}} * X; \text{// multi matr.} \\ B = \frac{1}{i} * B; \text{// escalar} \\ T = B; \text{// cópia matr.} \\ E = E_{\text{ant}} + B; \text{// soma matr.} \\ i++; \end{cases}$$

Exponencial de matriz X

$$\text{Início}: E = I; T = I; i = 1$$

$$\text{enqto } i <= k \begin{cases} B = T_{\text{ant}} * X; \text{// multi matr.} \\ B = \frac{1}{i} * B; \text{// escalar} \\ T = B; \text{// cópia matr.} \\ E = E_{\text{ant}} + B; \text{// soma matr.} \\ i++; \end{cases}$$

```
void expm (double E[ ][100],
           double X[ ][100],
           int n, int k) {
  int i;
  double B[ ][100], double T[ ][100];
  identidade (B, n);
  identidade (T, n);
  i = 1;
  while (i <= k) { // 1
    multi (B, T, X, n);
    escalar (B, n, 1.0/i);
    copia (T, B, n);
    soma (E, T, n);
    i++;
  } // 1
} // fim expm
```

Leitura e gravação de arquivo em disco

```
#include <stdio.h>
#define NumMax 80 // numero max de caracteres em cada linha
// Nome de um arquivo e' lido do teclado. Por ex., c:/Aentra.txt
// ler e gravar de disco
main(){
        int j;
        char lin[NumMax]; // lin e' vetor de chars p/ conter 80 chars
        char NomeArq[NumMax]; // nome do arquivo a ser lido do teclado
        char ProxLetra; // proxima letra
        FILE *ArqEntrada;
        FILE *ArqSaida;
        printf("Digite a seguir o nome do arquivo de entrada, e ENTER \n");
        scanf("%s", NomeArq);
        ArqEntrada=fopen(NomeArq, "r"); // abrir p/ ler
        ArqSaida=fopen("c:/Asai.txt", "w"); // abrir p/ gravar
```

Leitura e gravação de arquivo em disco (cont.)

```
if(ArqEntrada==NULL)printf("Arquivo .txt cujo nome V digitou nao existe");
            else{
              while( !feof(ArqEntrada) ){
                ProxLetra='x';
                // a seguir ler letra por letra ate' encontrar '\0'==fim-de-linha
                for(j=0;(j<NumMax)&&(ProxLetra!='\0');j++){
                  fscanf(ArqEntrada,"%c",&ProxLetra);
                  printf("%c",ProxLetra); // mostra na tela 1 letra de cada vez
                  fprintf(ArqSaida,"%c",ProxLetra); // grava letra/letra
                  }// fim for j
                }// fim while  !feof

            }// fim else

            fclose(ArqEntrada); // fecha
            fclose(ArqSaida); // fecha
}//fim main
```