Aula 13

MC 102 - Algoritmos e Programação de Computadores

Matrizes e vetores multidimensionais.

Múltiplos Vetores

Às vezes temos a necessidade de armazenar vários vetores para um mesmo dado.

Voltando ao exemplo da aula passada, um vetor pode armazenar as notas da primeira prova de uma turma. Mas como armazenar as notas das 4 provas de uma turma e depois calcular a média final?

Múltiplos Vetores

Criamos 4 vetores?

```
float nota_P1[65],nota_P2[65],nota_P3[65],nota_P4[65];
```

Mas e se forem 100 notas? Criamos 100 vetores?

Múltiplos Vetores

Criamos 4 vetores?

```
float nota_P1[65],nota_P2[65],nota_P3[65],nota_P4[65];
```

Mas e se forem 100 notas? Criamos 100 vetores?

De novo, não há nada melhor?

Matrizes

Uma matriz representa um conjunto de vetores de mesmo tamanho.

- Podemos dizer que um **vetor simples** é um **vetor monodimensional**.
- Uma **matriz** é considerada um **vetor de duas dimensões**.

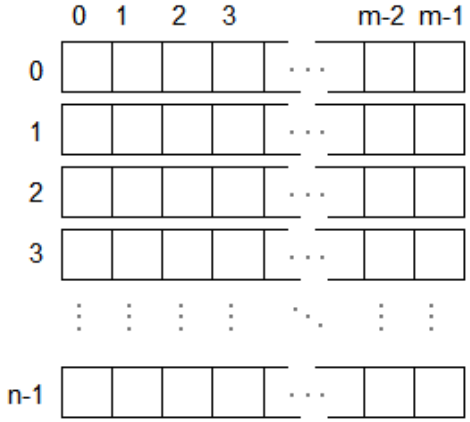
Uma matriz possui n linhas e m colunas:

- as linhas são numeradas de 0 a n -1
- as colunas são numeradas de 0 a m -1

Uma matriz pode armazenar (linhas * colunas) elementos de um mesmo tipo

Matrizes

Uma matriz de ***n*** linhas e ***m*** colunas



Matrizes

Sintaxe:

tipo_de_dado identificador[numero_de_linhas][numero_de_colunas];

tipo_de_dado:
char, int, short, float, etc...

identificador
Nome da variável para referência

numero_de_linhas
Quantidade de linhas que serão utilizadas

numero_de_colunas
Quantidade de colunas que serão utilizadas

Exemplo:

```
float nota[4][65];
```


Matrizes

Para acessar cada valor de uma matriz, é necessário:

- primeiro acessar um de seus vetores
- em seguida acessar a posição do vetor

Exemplos:

Acessar a 15a. posição do vetor 3o. vetor:

```
printf("%f\n", nota[2][14]);
```

Atribuir um valor:

```
i=2;
j=10;
nota[i][j] = 8.5;
```

Inicialização de Matrizes

Inicializando na declaração:

```
int identidade[4][4] = { {1, 0, 0, 0},
                         {0, 1, 0, 0},
                         {0, 0, 1, 0},
                         {0, 0, 0, 1} };
```

Inicialização de grandes matrizes:

```
for(i=0;i<5;i++)
   for(j=0;j<5;j++)
      nota[i][j] = 0.0;
```


Vetores Multidimensionais

Há casos em que uma matriz é insuficiente para armazenar um conjunto de dados para um determinado programa. Nestes casos, é necessário definir um vetor de ***d***-dimensões.

Para um vetor tridimensional:

```
int matriz[10][10][10]; /* cubo 10x10x10 */
```

Quatro dimensões:

```
int matriz[5][10][10][10]; /* 5 cubos 10x10x10 */
```


Programa 1

Faça um programa que leia duas matrizes 4 x 4 e mostre a multiplicação entre elas.

Entrada de dados
- Duas matrizes 4x4

Algoritmo
- Informe a matriz A
- Informe a matriz B
- Para cada i de 1 a 4 (Linhas de A e C)
  - Para cada j de 1 a 4 (Colunas de B e C)
    - Para cada k de 1 a 4 (Linhas de A e Colunas de B)
      - $C_{ij} = C_{ij} + A_{ik} * B_{kj}$
- Imprima a matriz C

Programa 1 - Código

```
#include <stdio.h>

#define LIM 4

int main(int argc, char **argv){
   int matA[LIM][LIM];
   int matB[LIM][LIM];
   int matC[LIM][LIM];
   int i,j,k;

   for (i=0; i<LIM; i++) {
     for (j=0; j<LIM; j++) {
       printf("Digite o valor da
         posicao (%d,%d) da
         matriz A: ",i,j);
       scanf("%d",&matA[i][j]);
       printf("Digite o valor da
         posicao (%d,%d) da
         matriz B: ",i,j);
       scanf("%d",&matB[i][j]);
     }
   }
```

Programa 1 - Código

```
#include <stdio.h>

#define LIM 4

int main(int argc, char **argv){
  int matA[LIM][LIM];
  int matB[LIM][LIM];
  int matC[LIM][LIM];
  int i,j,k;

  for (i=0; i<LIM; i++) {
    for (j=0; j<LIM; j++) {
      printf("Digite o valor da
        posicao (%d,%d) da
        matriz A: ",i,j);
      scanf("%d",&matA[i][j]);
      printf("Digite o valor da
        posicao (%d,%d) da
        matriz B: ",i,j);
      scanf("%d",&matB[i][j]);
    }
  }

  /* Inicializacao da Matriz C */
  for (i=0; i<LIM; i++)
    for (j=0; j<LIM; j++)
      matC[i][j]=0;

  for (i=0; i<LIM; i++)
    for (j=0; j<LIM; j++)
      for (k=0; k<LIM; k++)
        matC[i][j] = matC[i][j] +
          matA[i][k] * matB[k][j];

  printf("\n Matriz A * Matriz B\n");
  for (i=0; i<LIM; i++){
    for (j=0; j<LIM; j++)
      printf("%d ", matC[i][j]);
    printf("\n");
  }
  return 0;
}
```


Exercício para entrega

Escreva um programa que, dadas duas matrizes A e B, com dimensões de no máximo 5x5 elementos, retorne:

1) a soma destas duas matrizes

2) a soma da diagonal principal das duas