



矩阵运算

矩阵的和与差

问题描述:

请编写程序计算两个 N 阶方阵 A、B 的和与差。

编程要求：N 阶方阵的阶数 N ($2 \leq N \leq 5$) 由键盘输入。N 阶方阵中的 $N \times N$ 个元素也由键盘输入。求 A+B 的运算用一个函数实现。求 A-B 的运算用另外一个函数实现。

$$\text{例如: } A = \begin{bmatrix} 1 & 2 & -1 \\ 3 & 6 & 2 \\ -8 & 9 & 4 \end{bmatrix} \quad B = \begin{bmatrix} 31 & 2 & 0 \\ -3 & 7 & 9 \\ 10 & 1 & -11 \end{bmatrix}$$

$$\text{则: } A+B = \begin{bmatrix} 32 & 4 & -1 \\ 0 & 13 & 11 \\ 2 & 10 & -7 \end{bmatrix} \quad A-B = \begin{bmatrix} -30 & 0 & -1 \\ 6 & -1 & -7 \\ -18 & 8 & 15 \end{bmatrix}$$

程序清单:

```
#include "stdio.h"
#include "stdlib.h"
#define N 10
int n;

void init(int x[N][N]) // 数组初始化
{
    int i, j;
    for(i=0; i<n; i++) // 随机生成 n 阶矩阵 a 的所有元素
        for(j=0; j<n; j++)
            x[i][j]=rand()%100;
}

void add(int a[][N], int b[][N], int c[][N])
{
    int i, j;
    for(i=0; i<n; i++)
        for(j=0; j<n; j++)
            c[i][j]=a[i][j]+b[i][j];
}

void dec(int a[][N], int b[][N], int c[][N])
{
    int i, j;
    for(i=0; i<n; i++)
```

```

        for(j=0;j<n;j++)
            c[i][j]=a[i][j]-b[i][j];
    }

void output(int c[][N])
{
    int i,j;
    for(i=0;i<n;i++)
    {
        for(j=0;j<n;j++)
            printf("%5d",c[i][j]);
        printf("\n");
    }
}

main()
{
    int i,j;
    int a[N][N],b[N][N],sum[N][N],diff[N][N];
    printf("Input n(n<=10):");
    scanf("%d",&n);
    init(a);
    printf("matrix A\n");
    output(a);
    init(b);
    printf("matrix B\n");
    output(b);
    add(a,b,sum);
    printf("A+B\n");
    output(sum);
    dec(a,b,diff);
    printf("A-B\n");
    output(diff);
    getch();
}

```

矩阵转置

数字知识:

把矩阵 \mathbf{A} 的行与列互换所得到的矩阵称为矩阵 \mathbf{A} 的转置矩阵, 记为 \mathbf{A}^T , 即

$$\mathbf{A} = \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \cdots & \cdots & \cdots & \cdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{pmatrix}, \quad \mathbf{A}^T = \begin{pmatrix} a_{11} & a_{21} & \cdots & a_{m1} \\ a_{12} & a_{22} & \cdots & a_{m2} \\ \cdots & \cdots & \cdots & \cdots \\ a_{1n} & a_{2n} & \cdots & a_{mn} \end{pmatrix}$$

矩阵的转置方法与行列式相类似, 但是矩阵转置后, 行、列数都变了, 各元素的位置也变了, 所以通

常 $A \neq A^T$ 。

问题描述:

对于n阶方阵A, 阶数n ($1 < n \leq 10$) 由键盘输入; 使用随机函数产生 n^2 个小于 100 的整数存储在方阵A中, 输出方阵A和其对应转置矩阵 A^T 。

例如:

n=3, 所生成的 3 阶方阵 A 和 A^T 如下图所示:

$$A = \begin{bmatrix} 1 & 4 & 9 \\ 6 & 2 & 5 \\ 8 & 7 & 3 \end{bmatrix} \quad A^T = \begin{bmatrix} 1 & 6 & 8 \\ 4 & 2 & 7 \\ 9 & 5 & 3 \end{bmatrix}$$

程序清单:

```
#include "stdio.h"
#include "stdlib.h"
#define N 11
int a[N][N], at[N][N], n;

void init()
{
    int i, j;
    printf("Input n:");
    scanf("%d", &n);
    for(i=1; i<=n; i++)
        for(j=1; j<=n; j++)
        {
            a[i][j]=rand()%100;
            at[j][i]=a[i][j];
        }
}

void output(int x[N][N])
{
    int i, j;
    for(i=1; i<=n; i++)
    {
        for(j=1; j<=n; j++) printf("%5d", x[i][j]);
        printf("\n");
    }
}

main()
{
    init();
    printf("matrix A:\n");
    output(a);
```

```
printf("matrix A(T):\n");
output(at);
getch();
}
```

矩阵乘法

数学知识:

如果矩阵 A 的列数与矩阵 B 的行数相同, 即 A 是 $m \times s$ 矩阵, B 是 $s \times n$ 矩阵, 那么 A 、 B 可以相乘, 记为 AB 或 $A \cdot B$, 称为矩阵 A 、 B 的乘积。 $AB = C$ 表示一个 $m \times n$ 矩阵, 矩阵 C 的构成规则如下:

B 的第 1 列元素依次与 A 的各行元素相组合, 形成 C 的第 1 列元素; B 的第 2 列元素依次与 A 的各行元素相组合, 形成 C 的第 2 列元素; ……以此类推, 最后 B 的第 n 列元素依次与 A 的各行元素相组合, 形成 C 的第 n 列元素。这里的“组合”表示两两相乘再相加。

若记 $A = (a_{ij})_{m \times s}$, $B = (b_{ij})_{s \times n}$, $C = (c_{ij})_{m \times n}$, 且 $C = AB$, 则乘积矩阵 C 的元素可用公式表示为

$$c_{ij} = \sum_{k=1}^s a_{ik} b_{kj} \quad (i=1, 2, \dots, m; j=1, 2, \dots, n) \quad (2.1)$$

例如

$$\begin{pmatrix} 3 & -1 \\ 0 & 3 \\ 1 & 4 \\ 2 & 1 \end{pmatrix} \begin{pmatrix} 1 & -2 & 3 \\ 2 & 1 & 0 \end{pmatrix} = \begin{pmatrix} 3 \times 1 + (-1) \times 2 & 3 \times (-2) + (-1) \times 1 & 3 \times 3 + (-1) \times 0 \\ 0 \times 1 + 3 \times 2 & 0 \times (-2) + 3 \times 1 & 0 \times 3 + 3 \times 0 \\ 1 \times 1 + 4 \times 2 & 1 \times (-2) + 4 \times 1 & 1 \times 3 + 4 \times 0 \\ 2 \times 1 + 1 \times 2 & 2 \times (-2) + 1 \times 1 & 2 \times 3 + 1 \times 0 \end{pmatrix} = \begin{pmatrix} 1 & -7 & 9 \\ 6 & 3 & 0 \\ 9 & 2 & 3 \\ 4 & -3 & 6 \end{pmatrix}$$

程序清单:

```
#include "stdio.h"
#define M 4
#define S 2
#define N 3

// 实现矩阵的乘法运算, 结果存放在数组 c 中
void matrix(int a[][S], int b[][N], int c[][N])
{
    int i, j, k;
    for(i=0; i<M; i++)
        for(j=0; j<N; j++)
        {
            for(k=0; k<S; k++) c[i][j] += a[i][k] * b[k][j];
        }
}

void output(int c[][N]) // 输出 c 矩阵
```

```

{
    int i, j;
    for(i=0; i<M; i++)
    {
        for(j=0; j<N; j++) printf("%3d", c[i][j]);
        printf("\n");
    }
}

main()
{
    int a[M][S]={3, -1, 0, 3, 1, 4, 2, 1}; // 矩阵 A 是一个 m*s 的矩阵
    int b[S][N]={1, -2, 3, 2, 1, 0};      // 矩阵 B 是一个 s*n 的矩阵
    int c[M][N]={0};                      // 矩阵 C 是一个 m*n 的矩阵
    matrix(a, b, c);
    output(c);
    getch();
}

```