2021《计算机程序设计》课程大作业

李佩哲
PB21051049
项目起止日期：自2021.11.21至2021.11.25
项目名称：

Project 5. 矩阵的基本运算

〇、题目

---

1 知识点:

二维数组，动态内存分配，相关数学知识。

主要功能模块：

---

2 功能要求:

用C程序实现矩阵的基本运算(根据需要设定运算条件):

初始化矩阵;
打印矩阵;
求转置矩阵;
两个矩阵相加;
求两个矩阵的乘积;
求满足条件的方阵$\mathbf{A}$的伴随矩阵$\mathbf{A}^*$ 、逆矩阵$\mathbf{A}^{-1}$;

一、设计和开发过程总结

过程

首先，根据题意，可以将此Project分为3个功能区，即初始化、运算、控制，其中运算又可分为6个部分，分别完成题设6种运算。

初始化功能区

用于获取用户输入。

包含三个函数choose()、getInput_mn()、getInput()，分别获取用户输入的功能选择、矩阵大小、矩阵内容。

此功能区由于采用键盘的stdinput，故易产生输入类型错误、输入超限等错误，因此应采用异常处理以防止程序异常结束。

鉴于C没有现成的try语句，这里采用判断scanf()是否成功+goto语句的方法来进行异常处理：if (!scanf(“%d”,&某某))goto LOOP;若scanf()未成功就会执行goto语句，报错并返回上一步，要求用户重新输入，直到正确。中间采用getchar()是为了在程序运行中间插入一个间断点，防止程序始终判定scanf()错误而陷入死循环。如：

```
LOOP3:
    printf("请输入m2×n2...\nm2 n2=");
    if (!scanf("%d%d",&m2,&n2)) {//判断是否输入整数
        printf("\n键入不是整数！请检查您的输入，并重新键入符合规范的数字！\n");
        getchar();//用于间断程序
        goto LOOP3;
    }
    break;
```

除了上述类型错误外，还会出现功能未定义、矩阵大小不存在、不符合运算法则等输入超限的错误，这些错误用if语句比较输入数值与可行数值区间比较就可以判断正误，比较简单。如：

```
LOOP2:
    getchar();//用于间断程序

    ..........

    switch (choice) {//判断行列数是否合规
        case 1://加减
            if(!(m1==m2&&n1==n2)||(!(m1>0&&n1>0&&m2>0&&n2>0))){
                printf("\n键入矩阵行列数不合规范！请检查您的输入，并重新键入符合规范的行数与列数！\n");
                goto LOOP2;
            }
            break;
```

运算功能区

用于进行题设运算。这里着重说一下求行列式，其余的要么简单得不必讲，要么可以由行列式的结果快速推出。

先看代码：

```
//Line244～Line272:以下为求方阵的行列式
double MIJ(double A[m1][n1],int i,int j, int n){//A是传入的方阵，i，j是去掉的元素所在的第i行第j列，n为A的阶数//Line244
    //这里设置阶数的目的是：因为根据函数定义，只能传入一个固定大小的数组，而剥洋葱的算法会让方阵一层层缩小，于是将方阵置于原方阵的左上角，使用阶数n来避免调用多余的量，这样也就有了Line179每次都要让M[][]归零的操作
//    求余子式
```

```
    double det(double A[m1][n1], int n);//下面那个求行列式的函数的函数声明
    double M[m1][n1];//定义余子式矩阵
    for(int r=0;r<m1;r++)//先让Mij为零矩阵，防止出错//Line249
        for(int s=0;s<n1;s++)
            M[s][r]=0;
    for(int a=0;a<n-1;a++)//获得余子式Mij
        for(int b=0;b<n-1;b++){
            if(a<i && b<j)M[a][b]=A[a][b];//获得i行j列元素的左上角区域
            else if (a<i && b>=j)M[a][b]=A[a][b+1];//获得i行j列元素的右上角区域
            else if (a>=i && b<j)M[a][b]=A[a+1][b];//获得i行j列元素的左下角区域
            else M[a][b]=A[a+1][b+1];//获得i行j列元素的右下角区域
        }
    return det(M,n-1);//获得余子式的行列式，然后无限套娃剥洋葱直至返回一个1x1行列式得到最终值，在逐层向上传递结果，得到最终答案
}

double det(double A[m1][n1], int n){//A是传入的方阵，n为A的阶数;
//这里设置阶数的目的是：因为根据函数定义，只能传入一个固定大小的数组，而剥洋葱的算法会让方阵一层层缩小，于是将方阵置于原方阵的左上角，使用阶数n来避免调用多余的量
//    求行列式
    double detA=0;//令行列式的初值为零，方便后面的加法运算
    //在这里令detA=0而不是在全局变量里令的原因是，每次解行列式，MIJ()都是在反复调用本函数，也就是说，会产生局部里的局部变量、局部里局部里的局部变量……以此类推，会出现detA套娃的现象，而最后的结果是要求detA一层一层向上返回值，所以如果它是全局变量，就会出现问题
    if(n==1)return A[0][0];//如果得到的是一个1x1的方阵，就返回其值，这也是上面求余子式函数的出口，最终返回值是这个，然后再一层一层向上返回，直到得出第n-1阶余子式行列式的值
    else if (n>1)//其他情况就继续往下剥洋葱算
        for (int i=0; i<n; i++)//按第一行展开
            detA+=pow(-1,i+2)*A[0][i]*MIJ(A, 0, i, n);//这里detA加上了一个代数余子式，而余子式是要调用上面一个函数来剥洋葱算的，最终的返回值在这里被使用
    return detA;//最后的最后，返回行列式的值
}//Line272
//以上为求方阵的行列式
```

根据行列式的定义，求解行列式的一般方法为按行按列展开。另一种化为三角阵的方法虽较为简便，但不会产生代数余子式，不利于伴随矩阵的计算，所以不采用。

于是采用按第一行展开。

整个代码块可以分为两部分：一部分是用于求余子式的MIJ()，另一部分是根据余子式求解行列式的值的detA()。

在调用函数时，二维数组A[m1][n1]首先传入detA()，然后detA依法求A的各个余子式的值，再求代数余子式，从而获得行列式的值。

这里遇到一个问题：余子式的值从何而来？这就需要调用MIJ()来获取余子式。通过代码我们可以看到，MIJ()运行结束后，只会得到A的一个余子式，而不能求其值，而且获得的余子式M[m1−1][n1−1]也不能作为返回值传回detA()。这就需要在MIJ()的返回值处调用detA()，以求得M[m1−1][n1−1]的值。如果我们假定detA()运行正常，那么它就一定会返回M的值，从而作为MIJ()的最终返回值传入到第一层调用的detA()中。

通过上面的推理，我们可以看出，上述两个函数是互相调用无限套娃的。这个时候就要我们采取一种有效措施使得这种循环能够在某种条件下终止，并逐层向上返回终值，直至最外层detA()的返回值。

接下来的主要矛盾就转化为如何去界定这个终止条件。

我们可以看出，求行列式的过程就宛如剥洋葱一般，是逐层深入、层层减少的。洋葱会剥到心处不可再剥，拆余子式时也会拆到一个不可再分的单位，于是我们自然而然地就联想到这个单位：一个1×1的矩阵，也就是一个元素。好了，终止条件找到了，就是当余子式的阶数降到1时，就终止循环，

并且返回一个数值。于是我们很容易就能发现我们要返回的这个数是多少：1×1的矩阵，即这个元素自身，它的行列式就是其本身。于是我们就返回这个值。
终止条件与返回值找到了，在循环中被调用的一层层函数就会一层层向上返回终值，直到M[m1−1][n1−1]的值。那么这个时候，MIJ()就会返回该值到最初的代数余子式计算式中。现在只要我们让detA()能够按第一行展开就可以使其正常运行，上述假设就能成立，就能得到最终行列式的值。那么设立一个for循环，第一行的每个元素都计算一遍，然后加和，就可以输出答案。从而就可以使其正常运行，上述假设就能成立，就能得到最终行列式的值。

其他的都是套公式，不必多言。

控制功能区
这个区域主要是main()函数，通过讲上述定义的所有函数有机串联在一起，通过适当的判断、赋值、传参，就可以实现代码的正常运行。

自评

经验心得：debug时可分块调试；在函数循环调用时一定给出终止条件；每次调用函数时都有新的数据存入的变量不可定义为全局变量（如函数`double det(double A[m1][n1], int n)`中的`double detA=0`就要每次重新定义）
任务量：正常（偏易）
工作难点：行列式的计算（见上）
优点：容错率高
缺点：程序运行在cmd或terminal的CLI中，美观性与实用性没有GUi强
创新点：使用if语句来行使try语句的功能，大大提高了容错率，在任务完成之前程序基本上不会崩溃

源程序清单：

二、源码

```
//
//  main.c
//  120005
//
//  Created by 李佩哲 on 2021/11/25.
//

#include <stdio.h>
#include <stdlib.h>
#include <math.h>

int m1=1,n1=1,m2=1,n2=1,choice=0,typeOfGet;
//m1,n1,m2,n2为两个矩阵的大小，一直都要调用，故设为全局变量并赋初值，实际值在getInput_mn()中赋给；
//choice为选择的功能的序号，在初期判断是用哪个函数时有多个函数要调用，故设为全局变量
//typeOfGet为选择的功能所需要输入的矩阵的数量，在获取输入时有多个函数要调用，故设为全局变量

void choose(void){
//    选择要进行的功能,得到选择功能的序号
    printf("请选择：1.加减；2.数乘；3.转置；4.相乘；5.伴随；6.求逆；7.行列式；0.退出\n");
    goto LOOP1;
LOOP2:
    getchar();//用于间断程序
LOOP1:
    printf(">>>");
    if (!scanf("%d",&choice)) {//判断是否输入整数
        printf("\n键入不是整数！请检查您的输入，并重新键入符合规范的数字！\n");
        goto LOOP2;
    }
    switch (choice) {
        case 1://加减
            typeOfGet = 2;
            break;
        case 2://数乘
            typeOfGet = 1;
            break;
        case 3://转置
            typeOfGet = 1;
            break;
        case 4://相乘
            typeOfGet = 2;
            break;
        case 5://伴随
            typeOfGet = 1;
            break;
        case 6://求逆
            typeOfGet = 1;
            break;
        case 7://行列式
            typeOfGet = 1;
            break;
        case 0://退出
            exit(0);
            break;
        default:
            choice = 0;
            printf("\n键入不在范围内！请输入正确的数值！\n");
            goto LOOP2;
            break;
    }
}

void getInput_mn(void){
```

```
63. //     获取输入的矩阵规格
64.     m1=n1=m2=n2=1;
65.     choose();
66.     goto LOOP1;
67. LOOP2:
68.     getchar();//用于间断程序
69. LOOP1:
70.     switch (typeOfGet) {
71.         case 1://如果只涉及一个矩阵，就只输入一个，另外一个令为零矩阵
72.             printf("请输入m×n...\nm n=");
73.             if (!scanf("%d%d",&m1,&n1)) {//判断是否输入整数
74.                 printf("\n键入不是整数！请检查您的输入，并重新键入符合规范的数字！\n");
75.                 goto LOOP2;
76.             }
77.             break;
78.         case 2://如果涉及了两个矩阵，就输入两个
79.             printf("请输入m1×n1...\nm1 n1=");
80.             if (!scanf("%d%d",&m1,&n1)) {//判断是否输入整数
81.                 printf("\n键入不是整数！请检查您的输入，并重新键入符合规范的数字！\n");
82.                 goto LOOP2;
83.             }
84.         LOOP3:
85.             printf("请输入m2×n2...\nm2 n2=");
86.             if (!scanf("%d%d",&m2,&n2)) {//判断是否输入整数
87.                 printf("\n键入不是整数！请检查您的输入，并重新键入符合规范的数字！\n");
88.                 getchar();//用于间断程序
89.                 goto LOOP3;
90.             }
91.             break;
92.         default:
93.             printf("请输入正确的数值！\n>>>");
94.             goto  LOOP2;
95.             break;
96.     }
97.     switch (choice) {//判断行列数是否合规
98.         case 1://加减
99.             if(!(m1==m2&&n1==n2)||(!(m1>0&&n1>0&&m2>0&&n2>0))){
100.                 printf("\n键入矩阵行列数不合规范！请检查您的输入，并重新键入符合规范的行数与列数！\n");
101.                 goto LOOP2;
102.             }
103.             break;
104.         case 2 ... 3:
105.             if (!(m1>0&&n1>0&&m2>0&&n2>0)) {
106.                 printf("\n键入矩阵行列数不合规范！请检查您的输入，并重新键入符合规范的行数与列数！\n");
107.                 goto LOOP2;
108.             }
109.             break;
110.         case 4://相乘
111.             if ((n1!=m2)||(!(m1>0&&n1>0&&m2>0&&n2>0))) {
112.                 printf("\n键入矩阵行列数不合规范！请检查您的输入，并重新键入符合规范的行数与列数！\n");
113.                 goto LOOP2;
114.             }
115.             break;
116.         case 5 ... 7://伴随、求逆、行列式都要求为方阵
117.             if ((n1!=m1)||(!(m1>0&&n1>0&&m2>0&&n2>0))) {
118.                 printf("\n键入矩阵行列数不合规范！请检查您的输入，并重新键入符合规范的行数与列数！\n");
119.                 goto LOOP2;
120.             }
121.             break;
122.         case 0://退出
```

```
            exit(0);
            break;
        default:
            choice = 0;
            printf("\n请输入正确的数值！\n>>>");
            goto LOOP2;
            break;
    }
}

void getInput(double A[m1][n1], double B[m2][n2]){
//    获取矩阵输入
LOOP1:
    printf("请输入A：\n");
    for(int i=0;i<m1;i++){//输入A
        printf("第%d行：\n>>>",i+1);
        for(int j=0;j<n1;j++)
            if (!scanf("%lf",&A[i][j])) {//判断是否输入整数
                printf("\n键入不是整数！请检查您的输入，并重新键入符合规范的数字！\n");
                getchar();//用于间断程序
                goto LOOP1;
            }
    }
    printf("输入A=\n");
    for(int i=0;i<m1;i++){//输出A
        for(int j=0;j<n1;j++)
            printf("%lf ",A[i][j]);
        printf("\n");
    }//如果只涉及一个矩阵，就只输入一个，另外一个令为零矩阵
    if (typeOfGet == 2) {//如果涉及了两个矩阵，就输入两个
    LOOP2:
        printf("请输入B：\n");
        for(int i=0;i<m2;i++){//输入B
            printf("第%d行：\n>>>",i+1);
            for(int j=0;j<n2;j++)
                if (!scanf("%lf",&B[i][j])) {//判断是否输入整数
                    printf("\n键入不是整数！请检查您的输入，并重新键入符合规范的数字！
\n");
                    getchar();//用于间断程序
                    goto LOOP2;
                }
        }
        printf("输入B=\n");
        for(int i=0;i<m2;i++){//输出B
            for(int j=0;j<n2;j++)
                printf("%lf ",B[i][j]);
            printf("\n");
        }
    }
}

int plus(double A[m1][n1],double B[m2][n2]){
//    加法
    double C[m1][n1];
    if(!(m1==m2 && n1==n2)){
        printf("A，B行列数不一致！\n");
        return 0;//返回不能加
    }
    printf("A+B=\n");
    for (int i=0; i<m1; i++) {//做加法并同时输出结果
        for (int j=0; j<n1; j++) {
            C[i][j]=A[i][j]+B[i][j];
            printf("%lf ",C[i][j]);
        }
        printf("\n");
```

```
    }
    return 1;//返回能加
}

void lambda(double A[m1][n1]){
//    数乘
    double l;
    printf("λ=");
    scanf("%lf",&l);
    double C[m1][n1];
    printf("λA=\n");
    for (int i=0; i<m1; i++) {//做数乘并同时输出结果
        for (int j=0; j<n1; j++) {
            C[i][j]=l*A[i][j];
            printf("%lf ",C[i][j]);
        }
        printf("\n");
    }
}

void T(double A[m1][n1]){
//    转置
    double C[n1][m1];
    printf("A^T=\n");
    for (int i=0; i<m1; i++)
        for (int j=0; j<n1; j++)//做转置
            C[j][i]=A[i][j];
    for (int i=0; i<n1; i++){
        for (int j=0; j<m1; j++)//做转置
            printf("%lf ",C[i][j]);
        printf("\n");
    }
}

int Times(double A[m1][n1],double B[m2][n2]){
//    乘法
    if(n1!=m2){
        printf("A列数，B行数不一致！\n");
        return 0;//返回不能乘
    }
    double C[m1][n2];
    for(int i=0;i<m1;i++)
        for(int j=0;j<n2;j++)
            C[i][j]=0;
    printf("A×B=\n");
    for (int i=0; i<m1; i++) {//做乘法并同时输出结果
        for (int j=0; j<n2; j++) {
            for (int r=0; r<n1; r++)
                C[i][j]+=A[i][r]*B[r][j];
            printf("%lf ",C[i][j]);
        }
        printf("\n");
    }
    return 1;//返回能乘
}

//Line244～Line272:以下为求方阵的行列式
double MIJ(double A[m1][n1],int i,int j, int n){//A是传入的方阵，i，j是去掉的元素所在的第i行第j列，n为A的阶数//Line244
    //这里设置阶数的目的是：因为根据函数定义，只能传入一个固定大小的数组，而剥洋葱的算法会让方阵一层层缩小，于是将方阵置于原方阵的左上角，使用阶数n来避免调用多余的量，这样也就有了Line179每次都要让M[][]归零的操作
//    求余子式
    double det(double A[m1][n1], int n);//下面那个求行列式的函数的函数声明
    double M[m1][n1];//定义余子式矩阵
    for(int r=0;r<m1;r++)//先让Mij为零矩阵，防止出错//Line249
```

```
250.         for(int s=0;s<n1;s++)
251.             M[s][r]=0;
252.     for(int a=0;a<n-1;a++)//获得余子式Mij
253.         for(int b=0;b<n-1;b++){
254.             if(a<i && b<j)M[a][b]=A[a][b];//获得i行j列元素的左上角区域
255.             else if (a<i && b>=j)M[a][b]=A[a][b+1];//获得i行j列元素的右上角区域
256.             else if (a>=i && b<j)M[a][b]=A[a+1][b];//获得i行j列元素的左下角区域
257.             else M[a][b]=A[a+1][b+1];//获得i行j列元素的右下角区域
258.         }
259.     return det(M,n-1);//获得余子式的行列式，然后无限套娃剥洋葱直至返回一个1x1行列式得到最终值，在逐层向上传递结果，得到最终答案
260. }
261.
262. double det(double A[m1][n1], int n){//A是传入的方阵，n为A的阶数；
263. //这里设置阶数的目的是：因为根据函数定义，只能传入一个固定大小的数组，而剥洋葱的算法会让方阵一层层缩小，于是将方阵置于原方阵的左上角，使用阶数n来避免调用多余的量
264. //     求行列式
265.     double detA=0;//令行列式的初值为零，方便后面的加法运算
266.     //在这里令detA=0而不是在全局变量里令的原因是，每次解行列式，MIJ()都是在反复调用本函数，也就是说，会产生局部里的局部变量、局部里局部里的局部变量……以此类推，会出现detA套娃的现象，而最后的结果是要求detA一层一层向上返回值，所以如果它是全局变量，就会出现问题
267.     if(n==1)return A[0][0];//如果得到的是一个1x1的方阵，就返回其值，这也是上面求余子式函数的出口，最终返回值是这个，然后再一层一层向上返回，直到得出第n-1阶余子式行列式的值
268.     else if (n>1)//其他情况就继续往下剥洋葱算
269.         for (int i=0; i<n; i++)//按第一行展开
270.             detA+=pow(-1,i+2)*A[0][i]*MIJ(A, 0, i, n);//这里detA加上了一个代数余子式，而余子式是要调用上面一个函数来剥洋葱算的，最终的返回值在这里被使用
271.     return detA;//最后的最后，返回行列式的值
272. }//Line272
273. //以上为求方阵的行列式
274.
275. double company(double A[m1][n1]){
276. //     伴随矩阵
277.     if(m1!=n1){
278.         printf("输入不是方阵！\n");
279.         return 0;
280.     }
281.     double C[m1][n1];
282.     printf("A*=\n");
283.     for (int i=0;i<m1;i++){
284.         for(int j=0;j<n1;j++){
285.             C[i][j]=pow(-1,i+j+2)*MIJ(A, j, i, m1);
286.             if(C[i][j]==-0)C[i][j]=0;
287.             printf("%lf ",C[i][j]);
288.         }
289.         printf("\n");
290.     }
291.     return 1;
292. }
293.
294. int contrast(double A[m1][n1]){
295. //     逆矩阵
296.     if(m1!=n1){
297.         printf("输入不是方阵！\n");
298.         return 0;
299.     }
300.     if(det(A, m1)==0){
301.         printf("不可逆！\n");
302.         return 0;
303.     }
304.     double C[m1][n1];
305.     printf("A^(-1)=\n");
306.     for (int i=0;i<m1;i++){
307.         for(int j=0;j<n1;j++){
```

```
            C[i][j]=(pow(-1,i+j+2)*MIJ(A, j, i, m1))/det(A, m1);
            if(C[i][j]==-0)C[i][j]=0;
            printf("%lf ",C[i][j]);
        }
        printf("\n");
    }
    return 1;
}

int main() {
LOOP:
    getInput_mn();//获得mxn
    double A[m1][n1],B[m2][n2];
    getInput(A,B);//获取A, B
    switch (choice) {
        case 1://加法
            if(!plus(A,B))goto LOOP;
            break;
        case 2://数乘
            lambda(A);
            break;
        case 3://转置
            T(A);
            break;
        case 4://乘法
            if(!Times(A,B))goto LOOP;
            break;
        case 5://伴随
            if(!company(A))goto LOOP;
            break;
        case 6://求逆
            if(!contrast(A))goto LOOP;
            break;
```

```
341.         case 7://行列式
342.             if(m1!=n1){
343.                 printf("输入不是方阵！\n");
344.                 goto LOOP;
345.             }
346.             printf("det(A)=%lf\n",det(A,m1));
347.             break;
348.         case 0://退出
349.             exit(0);
350.             break;
351.         default:
352.             break;
353.     }
354.     return 0;
355. }
```

三、输入输出测试

1. 标准测试

（1）加法

```
Last login: Thu Nov 25 20:58:17 on ttys002
/Users/page/Documents/4-app/Homework/All_Homework_c/120005/120005/main ; exit;
(base) page@PagedeMacBook-Pro ~ % /Users/page/Documents/4-app/Homework/
All_Homework_c/120005/120005/main ; exit;
请选择：1.加减；2.数乘；3.转置；4.相乘；5.伴随；6.求逆；7.行列式；0.退出
>>>1
请输入m1×n1...
m1 n1=3 3
请输入m2×n2...
m2 n2=3 3
请输入A:
第1行:
>>>1 2 3
第2行:
>>>4 5 6
第3行:
>>>7 8 9
输入A=
1.000000 2.000000 3.000000
4.000000 5.000000 6.000000
7.000000 8.000000 9.000000
请输入B:
第1行:
>>>9 8 7
第2行:
>>>6 5 4
第3行:
>>>3 2 1
输入B=
9.000000 8.000000 7.000000
```

云算网 人算不如云算

首页 | 矩阵运算 | 信号处理 | 数学规划 | 数据挖掘 | 关于我们

矩阵运算工具一览
矩阵乘法
在线计算
线性方程组
在线求解
矩阵的逆矩阵
在线求逆
行列式的值
在线求解
特征值和特征向量
在线计算
Cholesky分解
在线求解
上三角下三角分解
在线求解
奇异值分解(SVD)
在线求解
QR分解
在线求解
矩阵的秩
在线计算
矩阵范数
在线计算
最小二乘解
在线求解
矩阵条件数
在线求解

矩阵求逆

您输入的矩阵如下：

第1列	第2列	第3列	第4列	第5列
1.0000	2.0000	3.0000	4.0000	5.0000
6.0000	7.0000	8.0000	9.0000	10.0000
11.0000	12.0000	1.0000	14.0000	15.0000
16.0000	17.0000	18.0000	1.0000	20.0000
21.0000	22.0000	23.0000	24.0000	2.0000

您所输入问题的解如下：

第1列	第2列	第3列	第4列	第5列
-2.0969	1.4217	-0.0833	-0.1111	-0.1304
2.2217	-1.7290	0.1667	0.1667	0.1739
-0.0833	0.1667	-0.0833	-0.0000	-0.0000
-0.1111	0.1667	0.0000	-0.0556	-0.0000
-0.1304	0.1739	0.0000	-0.0000	-0.0435

扫码关注云算网公众号，及时获取云算网相关信息：

喜欢我们的网站吗？
将本站加入收藏夹。

对我们的网站有建议吗？
给我们来信吧。

```
6.000000 5.000000 4.000000
3.000000 2.000000 1.000000
A+B=
10.000000 10.000000 10.000000
10.000000 10.000000 10.000000
10.000000 10.000000 10.000000
Saving session...
...copying shared history...
...saving history...truncating history files...
...completed.

[进程已完成]
```

(2) 数乘

```
Last login: Thu Nov 25 21:03:00 on ttys002
/Users/page/Documents/4-app/Homework/All_Homework_c/120005/120005/main ; exit;
(base) page@PagedeMacBook-Pro ~ % /Users/page/Documents/4-app/Homework/
All_Homework_c/120005/120005/main ; exit;
请选择: 1.加减; 2.数乘; 3.转置; 4.相乘; 5.伴随; 6.求逆; 7.行列式; 0.退出
>>>2
请输入m×n...
m n=3 3
请输入A:
第1行:
>>>1 2 3
第2行:
>>>4 5 6
第3行:
>>>7 8 9
输入A=
1.000000 2.000000 3.000000
4.000000 5.000000 6.000000
7.000000 8.000000 9.000000
λ=9
λA=
9.000000 18.000000 27.000000
36.000000 45.000000 54.000000
63.000000 72.000000 81.000000
Saving session...
...copying shared history...
...saving history...truncating history files...
...completed.

[进程已完成]
```

(3) 转置

```
Last login: Thu Nov 25 21:03:29 on ttys002
/Users/page/Documents/4-app/Homework/All_Homework_c/120005/120005/main ; exit;
(base) page@PagedeMacBook-Pro ~ % /Users/page/Documents/4-app/Homework/
All_Homework_c/120005/120005/main ; exit;
请选择: 1.加减; 2.数乘; 3.转置; 4.相乘; 5.伴随; 6.求逆; 7.行列式; 0.退出
>>>3
请输入m×n...
m n=3 4
请输入A:
第1行:
>>>1 2 3 4
第2行:
>>>5 6 7 8
第3行:
>>>9 10 11 12
输入A=
1.000000 2.000000 3.000000 4.000000
5.000000 6.000000 7.000000 8.000000
9.000000 10.000000 11.000000 12.000000
A^T=
1.000000 5.000000 9.000000
```

```
2.000000 6.000000 10.000000
3.000000 7.000000 11.000000
4.000000 8.000000 12.000000
Saving session...
...copying shared history...
...saving history...truncating history files...
...completed.

[进程已完成]
```

（4）相乘

```
Last login: Thu Nov 25 21:05:55 on ttys002
/Users/page/Documents/4-app/Homework/All_Homework_c/120005/120005/main ; exit;
(base) page@PagedeMacBook-Pro ~ % /Users/page/Documents/4-app/Homework/
All_Homework_c/120005/120005/main ; exit;
请选择：1.加减；2.数乘；3.转置；4.相乘；5.伴随；6.求逆；7.行列式；0.退出
>>>4
请输入m1×n1...
m1 n1=3 4
请输入m2×n2...
m2 n2=4 5
请输入A:
第1行:
>>>1 2 3 4
第2行:
>>>5 6 7 8
第3行:
>>>9 10 11 12
输入A=
1.000000 2.000000 3.000000 4.000000
5.000000 6.000000 7.000000 8.000000
9.000000 10.000000 11.000000 12.000000
请输入B:
第1行:
>>>20 19 18 17 16
第2行:
>>>15 14 13 12 11
第3行:
>>>10 9 8 7 6
第4行:
>>>5 4 3 2 1
输入B=
20.000000 19.000000 18.000000 17.000000 16.000000
15.000000 14.000000 13.000000 12.000000 11.000000
10.000000 9.000000 8.000000 7.000000 6.000000
5.000000 4.000000 3.000000 2.000000 1.000000
A×B=
100.000000 90.000000 80.000000 70.000000 60.000000
300.000000 274.000000 248.000000 222.000000 196.000000
500.000000 458.000000 416.000000 374.000000 332.000000
Saving session...
...copying shared history...
...saving history...truncating history files...
...completed.

[进程已完成]
```

对比网络结果：

（5）伴随

```
Last login: Thu Nov 25 21:18:01 on ttys002
/Users/page/Documents/4-app/Homework/All_Homework_c/120005/120005/main ; exit;
(base) page@PagedeMacBook-Pro ~ % /Users/page/Documents/4-app/Homework/
All_Homework_c/120005/120005/main ; exit;
请选择：1.加减；2.数乘；3.转置；4.相乘；5.伴随；6.求逆；7.行列式；0.退出
>>>5
请输入m×n...
```

```
m n=4 4
请输入A:
第1行:
>>>1 2 3 4
第2行:
>>>5 6 7 8
第3行:
>>>9 10 9 12
第4行:
>>>13 14 15 1
输入A=
1.000000 2.000000 3.000000 4.000000
5.000000 6.000000 7.000000 8.000000
9.000000 10.000000 9.000000 12.000000
13.000000 14.000000 15.000000 1.000000
A*=
272.000000 -228.000000 60.000000 16.000000
-318.000000 342.000000 -120.000000 -24.000000
60.000000 -120.000000 60.000000 0.000000
16.000000 -24.000000 0.000000 8.000000
Saving session...
...copying shared history...
...saving history...truncating history files...
...completed.

[进程已完成]
```

对比网络结果：

（6）求逆

```
Last login: Thu Nov 25 21:21:06 on ttys002
/Users/page/Documents/4-app/Homework/All_Homework_c/120005/120005/main ; exit;
(base) page@PagedeMacBook-Pro ~ % /Users/page/Documents/4-app/Homework/
All_Homework_c/120005/120005/main ; exit;
请选择：1.加减；2.数乘；3.转置；4.相乘；5.伴随；6.求逆；7.行列式；0.退出
>>>6
请输入m×n...
m n=5 5
请输入A:
第1行:
>>>1 2 3 4 5
第2行:
>>>6 7 8 9 10
第3行:
>>>11 12 1 14 15
第4行:
>>>16 17 18 1 20
第5行:
>>>21 22 23 24 2
输入A=
1.000000 2.000000 3.000000 4.000000 5.000000
6.000000 7.000000 8.000000 9.000000 10.000000
11.000000 12.000000 1.000000 14.000000 15.000000
16.000000 17.000000 18.000000 1.000000 20.000000
21.000000 22.000000 23.000000 24.000000 2.000000
A^(-1)=
-2.096860 1.421739 -0.083333 -0.111111 -0.130435
2.221739 -1.728986 0.166667 0.166667 0.173913
-0.083333 0.166667 -0.083333 0.000000 0.000000
-0.111111 0.166667 0.000000 -0.055556 0.000000
-0.130435 0.173913 0.000000 0.000000 -0.043478
Saving session...
...copying shared history...
...saving history...truncating history files...
...completed.

[进程已完成]
```

对比网络结果：

（7）（补充）求行列式

```
Last login: Thu Nov 25 21:28:48 on ttys002
/Users/page/Documents/4-app/Homework/All_Homework_c/120005/120005/main ; exit;
(base) page@PagedeMacBook-Pro ~ % /Users/page/Documents/4-app/Homework/
All_Homework_c/120005/120005/main ; exit;
请选择：1.加减；2.数乘；3.转置；4.相乘；5.伴随；6.求逆；7.行列式；0.退出
>>>7
请输入m×n...
m n=3 3
请输入A:
第1行:
>>>1 2 3
第2行:
>>>4 5 6
第3行:
>>>7 8 1
输入A=
1.000000 2.000000 3.000000
4.000000 5.000000 6.000000
7.000000 8.000000 1.000000
det(A)=24.000000
Saving session...
...copying shared history...
...saving history...truncating history files...
...completed.

[进程已完成]
```

对比网络结果：

(8) 退出

```
Last login: Thu Nov 25 21:32:07 on ttys002
/Users/page/Documents/4-app/Homework/All_Homework_c/120005/120005/main ; exit;
(base) page@PagedeMacBook-Pro ~ % /Users/page/Documents/4-app/Homework/
All_Homework_c/120005/120005/main ; exit;
请选择：1.加减；2.数乘；3.转置；4.相乘；5.伴随；6.求逆；7.行列式；0.退出
>>>0
Saving session...
...copying shared history...
...saving history...truncating history files...
...completed.

[进程已完成]
```

---

2. 异常测试

（1）判断所需功能时输入超限

所选功能不在范围内

```
Last login: Thu Nov 25 21:29:45 on ttys002
/Users/page/Documents/4-app/Homework/All_Homework_c/120005/120005/main ; exit;
(base) page@PagedeMacBook-Pro ~ % /Users/page/Documents/4-app/Homework/
All_Homework_c/120005/120005/main ; exit;
请选择：1.加减；2.数乘；3.转置；4.相乘；5.伴随；6.求逆；7.行列式；0.退出
>>>8

键入不在范围内！请输入正确的数值！
>>>
```

键入了非数字字符

```
Last login: Thu Nov 25 21:36:08 on ttys003
/Users/page/Documents/4-app/Homework/All_Homework_c/120005/120005/main ; exit;
(base) page@PagedeMacBook-Pro ~ % /Users/page/Documents/4-app/Homework/
All_Homework_c/120005/120005/main ; exit;
请选择：1.加减；2.数乘；3.转置；4.相乘；5.伴随；6.求逆；7.行列式；0.退出
>>>+

键入不是整数！请检查您的输入，并重新键入符合规范的数字！
>>>
```

（2）输入矩阵大小时输入超限

键入了非数字字符

```
Last login: Thu Nov 25 21:38:28 on ttys002
/Users/page/Documents/4-app/Homework/All_Homework_c/120005/120005/main ; exit;
(base) page@PagedeMacBook-Pro ~ % /Users/page/Documents/4-app/Homework/
All_Homework_c/120005/120005/main ; exit;
请选择：1.加减；2.数乘；3.转置；4.相乘；5.伴随；6.求逆；7.行列式；0.退出
>>>1
请输入m1×n1...
m1 n1=+

键入不是整数！请检查您的输入，并重新键入符合规范的数字！
请输入m1×n1...
m1 n1=
```

键入了非数字字符

```
Last login: Thu Nov 25 21:39:12 on ttys002
/Users/page/Documents/4-app/Homework/All_Homework_c/120005/120005/main ; exit;
(base) page@PagedeMacBook-Pro ~ % /Users/page/Documents/4-app/Homework/
All_Homework_c/120005/120005/main ; exit;
请选择：1.加减；2.数乘；3.转置；4.相乘；5.伴随；6.求逆；7.行列式；0.退出
>>>1
请输入m1×n1...
m1 n1=1 1
请输入m2×n2...
m2 n2=+

键入不是整数！请检查您的输入，并重新键入符合规范的数字！
请输入m2×n2...
m2 n2=
```

键入了非数字字符

```
Last login: Thu Nov 25 21:41:38 on ttys003
/Users/page/Documents/4-app/Homework/All_Homework_c/120005/120005/main ; exit;
(base) page@PagedeMacBook-Pro ~ % /Users/page/Documents/4-app/Homework/
All_Homework_c/120005/120005/main ; exit;
请选择：1.加减；2.数乘；3.转置；4.相乘；5.伴随；6.求逆；7.行列式；0.退出
>>>2
请输入m×n...
m n=+

```

```
键入不是整数！请检查您的输入，并重新键入符合规范的数字！
请输入m×n...
m n=
```

（3）输入矩阵大小时不合运算法则

矩阵大小不存在

```
Last login: Thu Nov 25 21:42:19 on ttys002
/Users/page/Documents/4-app/Homework/All_Homework_c/120005/120005/main ; exit;
(base) page@PagedeMacBook-Pro ~ % /Users/page/Documents/4-app/Homework/
All_Homework_c/120005/120005/main ; exit;
请选择：1.加减；2.数乘；3.转置；4.相乘；5.伴随；6.求逆；7.行列式；0.退出
>>>1
请输入m1×n1...
m1 n1=0 0
请输入m2×n2...
m2 n2=0 0

键入矩阵行列数不合规范！请检查您的输入，并重新键入符合规范的行数与列数！
请输入m1×n1...
m1 n1=
```

矩阵大小不存在

```
Last login: Thu Nov 25 21:42:47 on ttys002
/Users/page/Documents/4-app/Homework/All_Homework_c/120005/120005/main ; exit;
(base) page@PagedeMacBook-Pro ~ % /Users/page/Documents/4-app/Homework/
All_Homework_c/120005/120005/main ; exit;
请选择：1.加减；2.数乘；3.转置；4.相乘；5.伴随；6.求逆；7.行列式；0.退出
>>>2
请输入m×n...
m n=0 0

键入矩阵行列数不合规范！请检查您的输入，并重新键入符合规范的行数与列数！
请输入m×n...
m n=
```

两矩阵相加行列数不等

```
Last login: Thu Nov 25 21:43:18 on ttys002
/Users/page/Documents/4-app/Homework/All_Homework_c/120005/120005/main ; exit;
(base) page@PagedeMacBook-Pro ~ % /Users/page/Documents/4-app/Homework/
All_Homework_c/120005/120005/main ; exit;
请选择：1.加减；2.数乘；3.转置；4.相乘；5.伴随；6.求逆；7.行列式；0.退出
>>>1
请输入m1×n1...
m1 n1=1 1
请输入m2×n2...
m2 n2=2 2

键入矩阵行列数不合规范！请检查您的输入，并重新键入符合规范的行数与列数！
请输入m1×n1...
m1 n1=
```

两矩阵相乘$m_2 \neq n_1$

```
Last login: Thu Nov 25 21:45:41 on ttys002
/Users/page/Documents/4-app/Homework/All_Homework_c/120005/120005/main ; exit;
(base) page@PagedeMacBook-Pro ~ % /Users/page/Documents/4-app/Homework/
All_Homework_c/120005/120005/main ; exit;
请选择：1.加减；2.数乘；3.转置；4.相乘；5.伴随；6.求逆；7.行列式；0.退出
>>>4
请输入m1×n1...
m1 n1=2 2
请输入m2×n2...
m2 n2=3 3

键入矩阵行列数不合规范！请检查您的输入，并重新键入符合规范的行数与列数！
请输入m1×n1...
m1 n1=
```

非方阵求伴随矩阵

```
Last login: Thu Nov 25 21:46:24 on ttys002
/Users/page/Documents/4-app/Homework/All_Homework_c/120005/120005/main ; exit;
(base) page@PagedeMacBook-Pro ~ % /Users/page/Documents/4-app/Homework/
All_Homework_c/120005/120005/main ; exit;
请选择：1.加减；2.数乘；3.转置；4.相乘；5.伴随；6.求逆；7.行列式；0.退出
>>>5
请输入m×n...
m n=2 3

键入矩阵行列数不合规范！请检查您的输入，并重新键入符合规范的行数与列数！
请输入m×n...
m n=
```

非方阵求逆

```
Last login: Thu Nov 25 21:47:34 on ttys002
/Users/page/Documents/4-app/Homework/All_Homework_c/120005/120005/main ; exit;
(base) page@PagedeMacBook-Pro ~ % /Users/page/Documents/4-app/Homework/
All_Homework_c/120005/120005/main ; exit;
请选择：1.加减；2.数乘；3.转置；4.相乘；5.伴随；6.求逆；7.行列式；0.退出
>>>6
请输入m×n...
m n=2 3

键入矩阵行列数不合规范！请检查您的输入，并重新键入符合规范的行数与列数！
请输入m×n...
m n=
```

非方阵求行列式

```
Last login: Thu Nov 25 21:48:13 on ttys002
/Users/page/Documents/4-app/Homework/All_Homework_c/120005/120005/main ; exit;
(base) page@PagedeMacBook-Pro ~ % /Users/page/Documents/4-app/Homework/
All_Homework_c/120005/120005/main ; exit;
请选择：1.加减；2.数乘；3.转置；4.相乘；5.伴随；6.求逆；7.行列式；0.退出
>>>7
请输入m×n...
m n=2 3

键入矩阵行列数不合规范！请检查您的输入，并重新键入符合规范的行数与列数！
请输入m×n...
m n=
```

（4）输入矩阵时输入超限

键入了非数字字符

```
Last login: Thu Nov 25 21:48:37 on ttys002
/Users/page/Documents/4-app/Homework/All_Homework_c/120005/120005/main ; exit;
(base) page@PagedeMacBook-Pro ~ % /Users/page/Documents/4-app/Homework/
All_Homework_c/120005/120005/main ; exit;
请选择：1.加减；2.数乘；3.转置；4.相乘；5.伴随；6.求逆；7.行列式；0.退出
>>>1
请输入m1×n1...
m1 n1=2 2
请输入m2×n2...
m2 n2=2 2
请输入A:
第1行:
>>>+

键入不是整数！请检查您的输入，并重新键入符合规范的数字！
请输入A:
第1行:
>>>
```

键入了非数字字符

```
Last login: Thu Nov 25 21:49:50 on ttys002
/Users/page/Documents/4-app/Homework/All_Homework_c/120005/120005/main ; exit;
(base) page@PagedeMacBook-Pro ~ % /Users/page/Documents/4-app/Homework/
All_Homework_c/120005/120005/main ; exit;
```

```
请选择：1.加减；2.数乘；3.转置；4.相乘；5.伴随；6.求逆；7.行列式；0.退出
>>>1
请输入m1×n1...
m1 n1=2 2
请输入m2×n2...
m2 n2=2 2
请输入A:
第1行:
>>>2 2
第2行:
>>>2 2
输入A=
2.000000 2.000000
2.000000 2.000000
请输入B:
第1行:
>>>+

键入不是整数！请检查您的输入，并重新键入符合规范的数字！
请输入B:
第1行:
>>>
```

键入了非数字字符

```
Last login: Thu Nov 25 21:50:57 on ttys002
/Users/page/Documents/4-app/Homework/All_Homework_c/120005/120005/main ; exit;
(base) page@PagedeMacBook-Pro ~ % /Users/page/Documents/4-app/Homework/
All_Homework_c/120005/120005/main ; exit;
请选择：1.加减；2.数乘；3.转置；4.相乘；5.伴随；6.求逆；7.行列式；0.退出
>>>2
请输入m×n...
m n=2 2
请输入A:
第1行:
>>>+

键入不是整数！请检查您的输入，并重新键入符合规范的数字！
请输入A:
第1行:
>>>
```