

# 暨南大学本科实验报告专用纸

课程名称 高级程序设计实验 成绩评定             
实验项目名称 矩阵运算和函数封装 指导教师 张鑫源  
实验项目编号 09 实验项目类型          实验地点           
学生姓名 易雪 学号 2019054617  
学院 智能科学与工程 系          专业 信息安全  
实验时间 2020 年 5 月 10 日

## （一）实验目的

复习矩阵的运算，熟练掌握函数的封装方法；

## （二）实验内容和要求

### 1.内容：

对于矩阵 A,B 常规的矩阵操作有：A 的转置，A+B，A-B，A\*B,A 的 n 次方，  
分别用函数封装上述矩阵操作。

### 2.要求

- 1) A 和 B 为程序自定义；
- 2) 设置合理的输出展示封装效果；

## （三）主要仪器设备

仪器：计算机

实验环境：Dev-c++

## （四）源程序

```
1.  #include <stdio.h>
2.
3.  int main()
4.  {
5.      //所有函数声明
6.      void print_array(int *a, int row, int line); //输出矩阵函数
7.      void transpose(int *a, int *aT, int row, int line) ; //转置矩阵函数
8.      void plus_array(int *a, int *b, int *c, int row_a, int row_b, int line_a,
9.                      int line_b) ; //相加矩阵函数
10.     void minus_array(int *a, int *b, int *c, int row_a, int row_b, int line_
11.                      a, int line_b) ; //相减矩阵函数
12.     void mult_array(int *a, int *b, int *c, int row_a, int row_b, int line_a,
13.                     int line_b) ; //相乘矩阵函数
```

```
11. void power_array(int *a, int *b, int n) ;//乘方矩阵函数
12.
13. int A[3][3] = {1,2,3,1,1,1,4,5,6} ;
14. int B[3][3] = {2,3,4,2,1,4,3,4,5} ;
15. int C[3][4] = {2,5,6,3,4,4,4,4,1,0,0,2} ;
16. int CT[4][3] ;
17. int AB_plus[3][3] ;
18. int AC_plus[4][4] ;
19. int AB_minus[3][3] ;
20. int AC_mult[3][4] ;
21. int CA_mult[3][4] ;
22. int A_3power[3][3] ;
23.
24. //C 矩阵转置并输出
25. transpose(&C[0][0], &CT[0][0], 3, 4) ;
26. printf("C' : \n\n") ;
27. print_array(&CT[0][0],4,3) ;
28.
29. //A,B 矩阵相加并输出
30. plus_array(&A[0][0], &B[0][0], &AB_plus[0][0], 3, 3, 3, 3) ;
31. printf("A + B : \n\n") ;
32. print_array(&AB_plus[0][0], 3, 3) ;
33.
34. //测试 A,C 矩阵相加
35. plus_array(&A[0][0], &C[0][0], &AC_plus[0][0], 3, 3, 3, 4) ;
36.
37. //A,B 矩阵相减并输出
38. minus_array(&A[0][0], &B[0][0], &AB_minus[0][0], 3, 3, 3, 3) ;
39. printf("A - B : \n\n") ;
40. print_array(&AB_minus[0][0], 3, 3) ;
41.
42. //A,C 矩阵相乘并输出
43. mult_array(&A[0][0], &C[0][0], &AC_mult[0][0], 3, 3, 3, 4) ;
44. printf("A + C : \n\n") ;
45. print_array(&AC_mult[0][0], 3, 4) ;
46.
47. //测试 C,A 相乘
48. mult_array(&C[0][0], &A[0][0], &CA_mult[0][0], 3, 4, 3, 3) ;
49.
50. //计算 A 的三次方并输出
51. power_array(&A[0][0], &A_3power[0][0], 3 ) ;
52. printf("A^3 : \n\n") ;
53. print_array(&A_3power[0][0], 3, 3) ;
54.
```

```

55.     return 0;
56. }
57.
58. //输出矩阵定义
59. void print_array(int *a, int row, int line) {
60.     int i, j ;
61.     for( i=0; i<row; i++) {
62.         for(j=0; j<line; j++) {
63.             printf("%4d", *(a+i*line+j)) ;
64.         }
65.         printf("\n") ;
66.     }
67.     printf("\n") ;
68. }
69.
70. //转置矩阵定义
71. void transpose(int *a, int *aT, int row, int line) {
72.     int i, j;
73.     for(i=0; i<row; i++) {
74.         for(j=0; j<line; j++) {
75.             *(aT+j*row+i) = *(a+i*line+j) ;
76.         }
77.     }
78. }
79.
80. //相加矩阵定义
81. void plus_array(int *a, int *b, int *c, int row_a, int row_b, int line_a, int
    line_b) {
82.     int i, j ;
83.     if(row_a == row_b && line_a == line_b) {
84.         for(i=0; i<row_a; i++) {
85.             for(j=0; j<line_a; j++) {
86.                 *(c+i*line_a+j) = *(a+i*line_a+j) + *(b+i*line_a+j) ;
87.             }
88.         }
89.     }
90.     else {
91.         printf("这两个矩阵不符合矩阵加法运算规则！\n\n") ;
92.     }
93. }
94.
95. //相减矩阵定义
96. void minus_array(int *a, int *b, int *c, int row_a, int row_b, int line_a, i
    nt line_b) {

```

```

97.     int i, j ;
98.     if(row_a == row_b && line_a == line_b) {
99.         for(i=0; i<row_a; i++) {
100.            for(j=0; j<line_a; j++) {
101.                *(c+i*line_a+j) = *(a+i*line_a+j) - *(b+i*line_a+j) ;
102.            }
103.        }
104.    }
105.    else {
106.        printf("这两个矩阵不符合矩阵减法运算规则！\n\n") ;
107.    }
108. }
109.
110. //相乘矩阵定义
111. void mult_array(int *a, int *b, int *c, int row_a, int row_b, int line_a, in
    t line_b) {
112.     int i, j, k ;
113.     if(line_a == row_b) {
114.         for(i=0; i<row_a; i++) {
115.             for(j=0; j<line_b; j++) {
116.                 *(c+i*line_b+j) = 0 ;
117.                 for(k=0; k<row_a; k++) {
118.                     *(c+i*line_b+j) += *(a+i*line_a+k) * *(b+k*line_b+j) ;
119.                 }
120.             }
121.         }
122.     }
123.     else {
124.         printf("这两个矩阵不符合矩阵乘法运算规则！\n\n") ;
125.     }
126. }
127.
128. //乘方矩阵定义
129. void power_array(int *a, int *b, int n) {
130.     int t, i, j, k ;
131.     if(n>1) {
132.         int c[3][3]={0} ;
133.         for(i=0; i<3; i++) {
134.             for(j=0; j<3; j++) {
135.                 *(b+i*3+j) = *(a+i*3+j) ;
136.             }
137.         }
138.         for(t=1; t<n; t++) {
139.             for(i=0; i<3; i++) {

```

```

140.         for(j=0; j<3; j++) {
141.             for(k=0; k<3; k++){
142.                 c[i][j] += *(b+i*3+k) * *(a+k*3+j) ;
143.             }
144.
145.         }
146.     }
147.     for(i=0; i<3; i++) {
148.         for(j=0; j<3; j++) {
149.             *(b+i*3+j) = c[i][j] ;
150.             c[i][j] = 0 ;
151.         }
152.     }
153. }
154. }
155. else if (n=1) {
156.     for(i=0; i<3; i++) {
157.         for(j=0; j<3; j++) {
158.             *(b+i*3+j) = *(a+i*3+j) ;
159.         }
160.     }
161. }
162. else if(n=0) {
163.     for(i=0; i<3; i++) {
164.         for(j=0; j<3; j++) {
165.             *(b+i*3+j) = 0 ;
166.         }
167.     }
168. }
169. else {
170.     printf("无法计算! \n\n" ) ;
171. }
172. }

```

## (五) 实验调试

1.问题：函数内 return 返回不了一个数组；

解决：采用指针直接计算出结果；

## (六) 实验结果与分析

C:\Users\18047\Desktop\数组运算.exe

C' :

2	4	1
5	4	0
6	4	0
3	4	2

A + B :

3	5	7
3	2	5
7	9	11

这两个矩阵不符合矩阵加法运算规则！

A - B :

-1	-1	-1
-1	0	-3
1	1	1

A + C :

13	13	14	17
7	9	10	9
34	40	44	44

这两个矩阵不符合矩阵乘法运算规则！

A^3 :

126	164	202
54	70	86
288	374	460

-----  
Process exited after 0.04018 seconds with return value 0

请按任意键继续. . .

暨南大学本科实验报告专用纸(附页)

---