# 工程数学第七次实验报告

**实验名称：矩阵高阶运算2**

**姓 名： 薄劲阳**

**专 业： 计算机科学与技术**

**学 号： 2020115025**

**一、实验目的**

掌握矩阵运算基本概念和乘法结合律的应用及矩阵转置

**二、实验内容及设备**

**1、实验内容**

**生成一个4X4的随机正整数矩阵A，取三次不同的A，检验是否满足下式**

**（*A* +*I*）（*A* -*I*）=*A* 2-*I***

**（让左端减去右端，检验其差是否为零）**

2、笔记本电脑，台式机

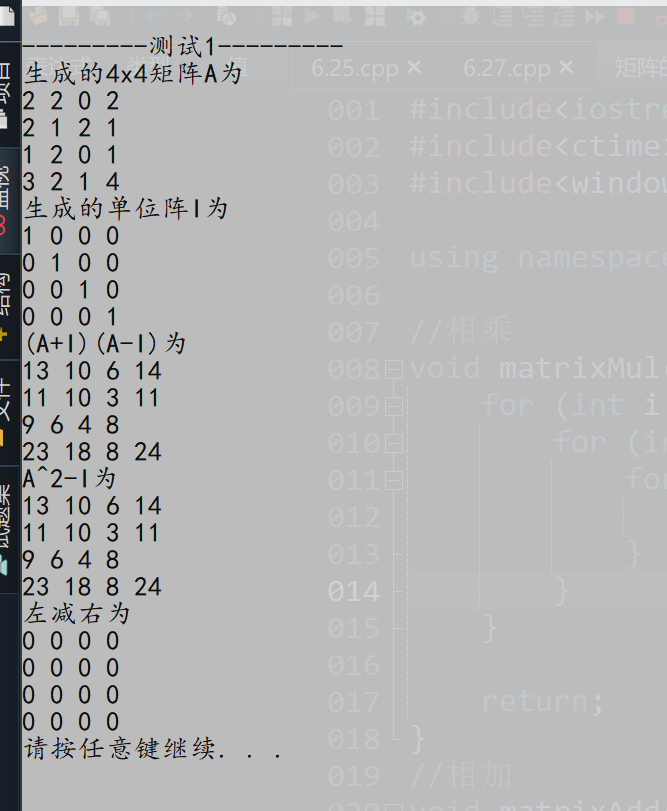
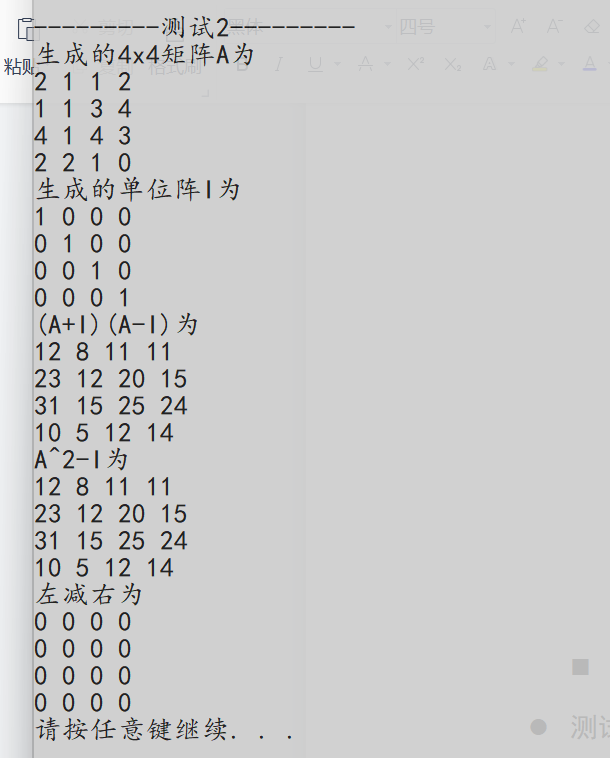
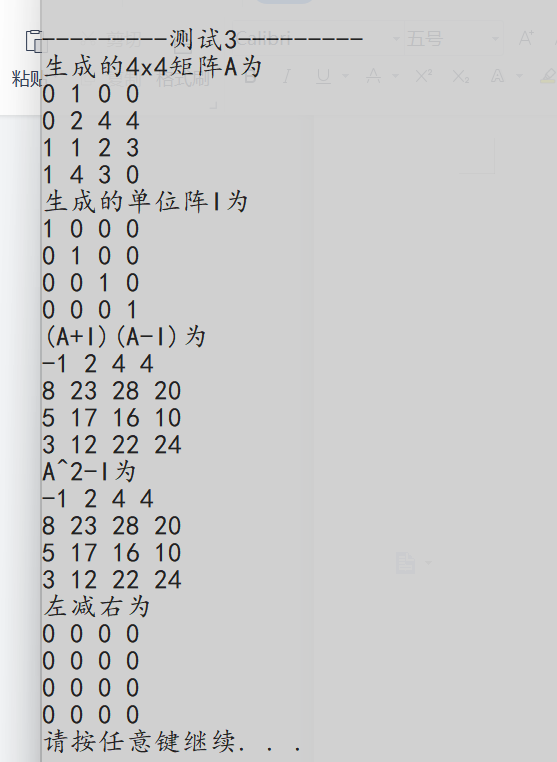
**三、程序代码**

1. #include<iostream>
2. #include<ctime>
3. #include<windows.h>
5. **using** **namespace** std;
7. //相乘
8. **void** matrixMul(**int** (\*A)[4], **int** (\*B)[4], **int** (\*AB)[4]) {
9. **for** (**int** i = 0; i < 4; i++) {
10. **for** (**int** j = 0; j < 4; j++) {
11. **for** (**int** k = 0; k < 4; k++) {
12. AB[i][j] += A[i][k] \* B[k][j]; //A(ik)\*B(kj)
13. }
14. }
15. }
17. **return**;
18. }
19. //相加
20. **void** matrixAdd(**int** (\*A)[4], **int** (\*B)[4], **int** (\*ApB)[4]) {
21. **for** (**int** i = 0; i < 4; i++) {
22. **for** (**int** j = 0; j < 4; j++) {
23. ApB[i][j] = A[i][j] + B[i][j];
24. }
25. }
26. **return**;
27. }
29. //相减
30. **void** matrixSub(**int** (\*A)[4], **int** (\*B)[4], **int** (\*AsB)[4]) {
31. **for** (**int** i = 0; i < 4; i++) {
32. **for** (**int** j = 0; j < 4; j++) {
33. AsB[i][j] = A[i][j] - B[i][j];
34. }
35. }
36. **return**;
37. }
39. //打印
40. **void** matrixPrint(**int** (\*A)[4]) {
41. **for** (**int** i = 0; i < 4; i++) {
42. **for** (**int** j = 0; j < 4; j++) {
43. cout << A[i][j] << " ";
44. }
45. cout << endl;
46. }
47. }
49. //矩阵生成
50. **void** matrixInit(**int** (\*A)[4]) {
51. srand((unsigned)time(NULL));
52. **for** (**int** i = 0; i < 4; i++) {
53. **for** (**int** j = 0; j < 4; j++) {
54. A[i][j] = rand() % 5;
55. }
56. }
57. }
59. **void** matrix\_EInit(**int** (\*A)[4]) {
60. **for** (**int** i = 0; i < 4; i++) {
61. A[i][i] = 1;
62. }
63. }
65. **void** test() {
66. **int** A[4][4] = {0};
67. **int** I[4][4] = {0};
68. //矩阵初始化
69. matrixInit(A);
70. matrix\_EInit(I);
71. cout << "生成的4x4矩阵A为" << endl;
72. matrixPrint(A);
73. cout << "生成的单位阵I为" << endl;
74. matrixPrint(I);
75. //计算(A+I)(A-I)
76. //计算A+I和A-I
77. **int** ApI[4][4] = {0}, AsI[4][4] = {0};
78. matrixAdd(A, I, ApI);
79. matrixSub(A, I, AsI);
80. **int** ApIAsI[4][4] = {0};
81. matrixMul(ApI, AsI, ApIAsI); //计算矩阵相乘
82. cout << "(A+I)(A-I)为" << endl;
83. matrixPrint(ApIAsI);
84. //计算A^2-I
85. **int** AA[4][4] = {0}, AAsI[4][4] = {0};
86. matrixMul(A, A, AA);
87. matrixSub(AA, I, AAsI);
88. cout << "A^2-I为" << endl;
89. matrixPrint(AAsI);
90. //计算左减右
91. **int** LsR[4][4] = {0};
92. matrixSub(ApIAsI, AAsI, LsR);
93. cout << "左减右为" << endl;
94. matrixPrint(LsR);
96. **return**;
97. }
99. **int** main() {
100. **for** (**int** i = 0; i < 3; i++) {
101. system("color 70");//设置背景色为白色
102. SetConsoleTitle("矩阵的高阶运算2");
103. cout << "\n---------测试" << i + 1 << "---------" << endl;
104. test();
105. //cmd翻页
106. system("pause");
107. system("cls");
108. }
110. **return** 0;
111. }

**四、代码分析**

* 验证该结论，只需按照对应步骤，进行相应的矩阵操作即可

**五、数据输入与运行结果**

* 测试一
  + 
* 测试二
  + 
* 测试三
  + 

**六、总结**

* 二维数组的本质
* 二维数组作为函数参数传递
* 矩阵的乘法结合律