Symmetric matrix operation

Class:计科201 student id:19404150205 name:刘志扬

Student id:20401010102 name:毛威任

Student id:19402010318 name:张俊杰

### Experiment purpose

Complete the operation of symmetric matrix through what you have learned

### 2.Experimental environment

1 1 2 4

1 2 3 5

2 3 4 6

4 5 6 7

1 1 1 1

1 1 1 1

1 1 1 1

1 1 1 1

A computer with Visual C ++ 6.0 / CFree.

This experiment has 4 class hours in all.

### 3.Experiment content

A and B are known to be two n × n-order symmetric matrix, write a program to realize:

(1)The lower triangular elements are stored in one-dimensional arrays a and B and output.

(2)Let C = a + B and output C in matrix mode.

(3) Let d = a × B. Output D in matrix mode.

Tip: the program can use int a [4] [4] and B [4] [4] two-dimensional arrays to represent the original matrices A and B.

### 4.Important data structures

Create classes to store matrix and lower triangular elements

typedef struct mat

{

int i;//行

int c;//列

int size;

int m[500][500];

}\*ptrtomat;

typedef ptrtomat matrix;

typedef struct angle1

{

int size;

int array[70];

}\*ptrtoangle;

typedef ptrtoangle angle;

### 5.implementation analysis

Adding and multiplying with arrays and create a new matrix to store the numbers added and multiplied

matrix add(matrix p1, matrix p2)

{

if (p1->i != p2->i)

{

cout << "两个矩阵格式不同不能运算" << endl;

throw - 2;

}

matrix p3 = create\_matrix();

p3->c = p1->c;

p3->i = p1->i;

p3->size = p1->size;

for (int i = 0; i <= p1->i - 1; i++)

{

for (int j = 0; j <= p1->c-1; j++)

{

p3->m[i][j] = p1->m[i][j] + p2->m[i][j];

}

}

return p3;

}

matrix mult(matrix p1, matrix p2)

{

if (p1->i != p2->i)

{

cout << "两个矩阵格式不同不能运算" << endl;

throw - 2;

}

matrix p3 = create\_matrix();

p3->c = p1->c;

p3->i = p1->i;

p3->size = p1->size;

int count = 0;

for (int i = 0; i <= p1->i - 1; i++)

{

for (int j = 0; j <= p1->c-1; j++)

{ count = 0;

for (int k = 0; k <= p1->i - 1; k++)

{

count += p1->m[i][k] \* p2->m[k][j];

}

p3->m[i][j] = count;

}

}

return p3;

}

read function:

Read array into matrix ,Determine whether it is a homogeneous matrix

### 6.debugging problem analysis

Problem:The operation process is too troublesome and the operation time is long.

Add rows and columns to the matrix class and bring them into the multiplication operation to reduce the operation process.

1. **Summary**

Through this experiment, we learned how to complete the operation of the matrix, and reviewed the creation, reading and writing of classes

### 8.crew division

|  |  |  |
| --- | --- | --- |
| Group division | | |
| Member name | Work done | Completion situation |
| 刘志扬 | Data struct using and each demand function | complete |
| 张俊杰 | Data struct using and each demand function | complete |
| 毛威任 | File read and write | Complete |