**Find the sum and product of Symmetric Matrices Experiment Report**

班级：计科201 实验日期：10.29

姓名1：郑永坤 学号：20410020101

姓名2：翟聪 学号：20403070103

姓名3：张子晗 学号：20405050102

1. **Experimental purpose**

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

1. **Experimental environment**

A computer with Visual C ++ 6.0 / CFree.

1. **Experimental content**

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

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

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

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

1. **Important data structures**

void init(istream &in, vector<vector<int> > &a){

for(int i = 0; i < n; i++) for(int j = 0; j < n; j++) in >> a[i][j];

}

vector<vector<int> > sum(vector<vector<int> > &a, vector<vector<int> > &b){

vector<vector<int> > ret(n, vector<int>(n));

for(int i = 0; i < n; i++){

for(int j = 0; j < n; j++){

ret[i][j] = a[i][j] + b[i][j];

}

}

return ret;

}

vector<vector<int> > multi(vector<vector<int> > &a, vector<vector<int> > &b){

vector<vector<int> > ret(n, vector<int>(n));

for(int i = 0; i < n; i++){

for(int j = 0; j < n; j++){

for(int z = 0; z < n; z++){

ret[i][j] += a[i][z] \* b[z][j];

}

}

}

return ret;

}

**Implementation analysis**

对称矩阵A的第i行和第j列的元素的数据存储在一维数组中的位置k的计算公式:

当i大于或等于j时，k = (i \* (i + 1)) / 2 + j (下三角)

当i小于j时，k = (j \* (j + 1)) / 2 + i (上三角)

1. **Debugging problem analysis**

**在矩阵乘法的函数里，我们刚开始认为只需要两层for循环就可以了，结果发现错误，**

**解决的方法，再加一层for循环**

1. **Summary**

加深了对矩阵运算的了解和对数组的认知，更让我们体会到数学对程序设计的重要性。

1. **Crew Division**

|  |  |  |
| --- | --- | --- |
| **Group division** | | |
| **Member name** | **Work done** | **Completion situation** |
| **郑永坤** | **第一问** | **圆满完成** |
| **翟聪** | **第二问** | **圆满完成** |
| **张子晗** | **第三问** | **圆满完成** |