**银行家算法问题**

回顾：银行家算法是一种预防死锁的算法。在避免死锁方法中允许进程动态地申请资源,但银行家算法在系统在进行资源分配之前,应先计算此次分配资源的安全性,若分配不会导致系统进入不安全状态,则分配,否则等待。

代码如下：

#include <iostream>  
#define maxP 10  
#define maxS 10  
using namespace std;  
int Available[maxS];  
int Max[maxP][maxS];  
int Finish[maxP];  
int Allocation[maxP][maxS];  
int Need[maxP][maxS];  
int Request[maxS];  
int path[maxP] = { 0 };  
int PNum, RNum;  
void outPath() {  
    cout << "系统安全序列是:\n";  
    cout << "P" << path[0] - 1;  
    for (int i = 1; path[i] != 0; i++) {  
        cout << "->P" << path[i] - 1;  
    }  
    for (int i = 0; i < PNum; i++) path[i] = 0;  
    cout << endl;  
}  
int BankSafe() {  
    int i, j, l = 0;  
    int Work[maxS];  
    for (i = 0; i < RNum; i++) Work[i] = Available[i];  
    for (i = 0; i < PNum; i++) Finish[i] = 0;  
    for (i = 0; i < PNum; i++) {  
        if (Finish[i] == 1)  
            continue;  
        else {  
            for (j = 0; j < RNum; j++) {  
                if (Need[i][j] > Work[j])  
                    break;  
            }  
            if (j == RNum) {  
                Finish[i] = 1;  
                for (int k = 0; k < RNum; k++)  
                    Work[k] += Allocation[i][k];  
                path[l++] = i + 1;  
                i = -1;  
            }  
            else continue;  
        }  
        if (l == PNum) {  
            return 1;  
        }  
    }  
    return 0;  
}  
void input(int PNum, int RNum) {  
    cout << "Enter the maximum number of resources required for each process:\n";  
    for (int i = 0; i < PNum; i++) {  
        cout << "P" << i << " : ";  
        for (int j = 0; j < RNum; j++)  
            cin >> Max[i][j];  
    }  
    cout << "Enter the number of resources that each process has allocated:\n";  
    for (int i = 0; i < PNum; i++) {  
        cout << "P" << i << " : ";  
        for (int j = 0; j < RNum; j++) {  
            cin >> Allocation[i][j];  
            Need[i][j] = Max[i][j] - Allocation[i][j];  
            if (Need[i][j] < 0) {  
                cout << "输入的进程P" << i << "所拥有的第" << j + 1 << "个资源错误，请重新输入:\n";  
                j--;  
                continue;  
            }  
        }  
    }  
    cout << "Please enter the existing number of each resource:\n";  
    for (int i = 0; i < RNum; i++)  
        cin >> Available[i];  
}  
int requestP() {  
    int Pi;  
    cout << "Enter the process number (0-4) of the resource you want to request:";  
    cin >> Pi;  
    Pi;  
    cout << "Enter the number of resources requested by the process:";  
    for (int i = 0; i < RNum; i++)  
        cin >> Request[i];  
    for (int i = 0; i < RNum; i++) {  
        if (Request[i] > Need[Pi][i]) {  
            cout << "The number of requested resources exceeds the demand of the process!\n";  
            return 1;  
        }  
        if (Request[i] > Available[i]) {  
            cout << "The number of requested resources exceeds all resources of the system!\n";  
            return 1;  
        }  
    }  
    for (int i = 0; i < RNum; i++) {  
        Available[i] -= Request[i];  
        Allocation[Pi][i] += Request[i];  
        Need[Pi][i] -= Request[i];  
    }  
    if (BankSafe()) {  
        cout << "System security!\n";  
        outPath();  
        cout << "Agree to the assignment request!\n";  
    }  
    else {  
        for (int i = 0; i < RNum; i++) {  
            Available[i] += Request[i];  
            Allocation[Pi][i] -= Request[i];  
            Need[Pi][i] += Request[i];  
        }  
        cout << "After the request, the system is not safe, your request is refused!\n";  
    }  
    return 0;  
}  
void outDATA() {  
    int i, j;  
    cout << "\nThe number of resources available for the system is:";  
    for (j = 0; j < RNum; j++)  
        cout << " " << Available[j];  
    cout << endl << "Additional resources required by processes:" << endl;  
    for (i = 0; i < PNum; i++) {  
        cout << "process P" << i << " :";  
        for (j = 0; j < RNum; j++)  
            cout << " " << Need[i][j];  
        cout << endl;  
    }  
    cout << endl << "Resources already available to processes:" << endl;  
    for (i = 0; i < PNum; i++) {  
        cout << "process P" << i << " :";  
        for (j = 0; j < RNum; j++)  
            cout << " " << Allocation[i][j];  
        cout << endl;  
    }  
    cout << endl;  
}  
int main() {  
    cout << "Number of input processes:";  
    cin >> PNum;  
    cout << "Type of input resource:";  
    cin >> RNum;  
    input(PNum, RNum);  
    if (BankSafe()) {  
        cout << "Current system security!\n";  
        outPath();  
    }  
    else {  
        cout << "The current system is not safe!\n";  
        return 0;  
    }  
    while (1) {  
        requestP();  
        outDATA();  
        char chose;  
        cout << "Request assignment again?If yes :Y/y，If no N/n:\n";  
        while (1) {  
            cin >> chose;  
            if (chose == 'Y' || chose == 'y' || chose == 'N' || chose == 'n')  
                break;  
            else {  
                cout << "Please re-enter as required:\n";  
                continue;  
            }  
        }  
        if (chose == 'Y' || chose == 'y')  
            continue;  
        else break;  
    }  
    return 0;  
}