## 第五次作业，包含的两个main函数如下：

#define \_CRT\_SECURE\_NO\_WARNINGS

#include<iostream>

#include<vector>

#include<Windows.h>

#include<assert.h>

using namespace std;

class Birthday

{

private:

int year, month, day;

public:

Birthday(int yy, int mm, int dd)

{

year = yy;

month = mm;

day = dd;

}

void BirthdayPrint()

{

cout << "生日：" << year << "/" << month << "/" << day << endl;

}

};

class People

{

public:

Birthday birthday;

friend ostream & operator<<(ostream&,People& );

friend istream & operator>>(istream&,People& );

People(string nn, string num, string ss, int yy, int mm, int dd, string id) : birthday(yy, mm, dd)

{

name = nn;

number = num;

sex = ss;

ID = id;

}

private:

string name;

string number;

string sex;

string ID;

};

ostream& operator<<(ostream& ,People& people)

{

cout <<"姓名："<< people.name << endl;

cout << "编号：" << people.number << endl;

cout << "性别：" << people.sex << endl;

people.birthday.BirthdayPrint();

cout << "ID：" << people.ID << endl;

return cout;

}

istream& operator>>(istream&,People& people)

{

//people.name[10] = {0};

cout << "请输入姓名：" << endl;

cin >> people.name;

cout << "请输入性别：" << endl;

cin >> people.sex;

cout << "请输入编号：" << endl;

cin >> people.number;

cout << "请输入生日：" << endl;

int yy, mm, dd;

cin >> yy >> mm >> dd;

people.birthday = { yy,mm,dd };

cout << "请输入ID：" << endl;

cin >> people.ID;

return cin;

}

//考虑到使用数组可能会占用不必要的空间，这里使用vector来代替数组，行使同样的功能；

int main()

{

People EMPTY = {"empty","empty","empty",0,0,0,"empty"};

vector<People> employee;

cout << "请输入员工数目：" << endl;

int N;

cin >> N;

system("cls");

for (int i = 0; i < N; i++)

{

cout << "对于第" << i + 1 << "个员工" << endl;

People Instance = EMPTY;

cin >> Instance;

employee.push\_back(Instance);

system("cls");

}

for (int i = 0; i < N; i++)

{

cout << employee[i];

cout << endl;

}

return 0;

}

class Matrix

{

private:

double\* data;

int x;

int y;

public:

Matrix() :data(NULL), x(0), y(0){}

Matrix(double\* pData, int xx, int yy):data(pData),x(xx),y(yy){}

~Matrix() { delete[]data; }

double& operator()(int xx, int yy);

friend ostream& operator<<(ostream&,Matrix& matrix);

void Trans()

{

cout << "该矩阵转置后为:" << endl;

vector<int> MatrixTemp;

for (int i = 0; i < x; i++)

{

for (int j = 0; j < y; j++)

MatrixTemp.push\_back(data[i \* y + j]);

}

int temp=x, num = 0;

x = y;

y = temp;

for (int i = 0; i < y; i++)

{

for (int j = 0; j < x; j++)

{

data[j \* y + i] = MatrixTemp[num];

num++;

}

}

}

};

double& Matrix :: operator()(int xx, int yy)

{

cout << "请依次输入矩阵的各项的值：" << endl;

data = new double[xx \* yy];

for (int i = 0; i < xx; i++)

{

for (int j = 0; j < yy; j++)

cin >> data[i \* yy + j];

}

return \*data;

}

ostream& operator<<(ostream&,Matrix& matrix)

{

for (int i = 0; i < matrix.x; i++)

{

for (int j = 0; j < matrix.y; j++)

{

cout << matrix.data[i \* matrix.y + j] << " ";

}

cout << endl;

}

return cout;

}

int main()

{

int x, y;

cout << "请输入所需要的矩阵的x,y值：" << endl;

cin >> x >> y;

Matrix m(NULL,x, y);

m(x, y);

cout << m;

m.Trans();

cout << m;

}

## 结果显示，第一题中间输入的过程需要操作可以看到步骤；



