上机实验（三）:

1. 程序代码

#include <iostream>

using namespace std;

class Score

{

public:

double A;

double B;

double C;

double ave;

string name;

}stu[100], z;

double all\_ave[5] = { 0 };

int x;

void sort()

{

char X;

cout << "选择排序科目：A B C" << endl;

cin >> X;

int i;

switch (X)

{

case 'A':

for (i = 0; i < x - 1; i++)

{

if (stu[i].A < stu[i + 1].A)

{

z = stu[i];

stu[i] = stu[i + 1];

stu[i + 1] = z;

}

}

break;

case 'B':

for (i = 0; i < x - 1; i++)

{

if (stu[i].B < stu[i + 1].B)

{

z = stu[i];

stu[i] = stu[i + 1];

stu[i + 1] = z;

}

}

break;

case 'C':

for (i = 0; i < x - 1; i++)

{

if (stu[i].C < stu[i + 1].C)

{

z = stu[i];

stu[i] = stu[i + 1];

stu[i + 1] = z;

}

}

break;

}

}

void print()

{

int i;

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

{

cout << "\n姓名：" << stu[i].name << "\n科目： A：" << stu[i].A << " 科目： B：" << stu[i].B << " 科目： C：" << stu[i].C << " 平均成绩：" << stu[i].ave << endl << endl;

}

}

void print\_ave()

{

cout << "\n科目A平均成绩：" << all\_ave[0] << " 科目B平均成绩：" << all\_ave[1] << " 科目C平均成绩：" << all\_ave[2] << endl << endl;

}

void menu()

{

int i;

while (1)

{

cout << "1.显示每位同学成绩和平均分\n2.显示每门科目平均成绩\n3.成绩排序\n4.退出\n" << endl;

cin >> i;

switch (i)

{

case 1:

print();

break;

case 2:

print\_ave();

break;

case 3:

sort();

print();

break;

}

if (i == 4)

break;

}

}

void input()

{

int i = 0, y = 0;

cout << "请输入学生数？(2-100)" << endl;

while (1)

{

cin >> x;

if (x > 100 || x < 2)

{

cout << "输入错误！" << endl;

}

else

break;

}

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

{

cout << "输入学生" << i + 1 << "姓名" << endl;

cin >> stu[i].name;

cout << "输入成绩A：";

cin >> stu[i].A;

cout << "输入成绩B：";

cin >> stu[i].B;

cout << "输入成绩C：";

cin >> stu[i].C;

stu[i].ave = (stu[i].A + stu[i].B + stu[i].C) / 3;

all\_ave[0] += stu[i].A / x;

all\_ave[1] += stu[i].B / x;

all\_ave[2] += stu[i].C / x;

}

}

int main()

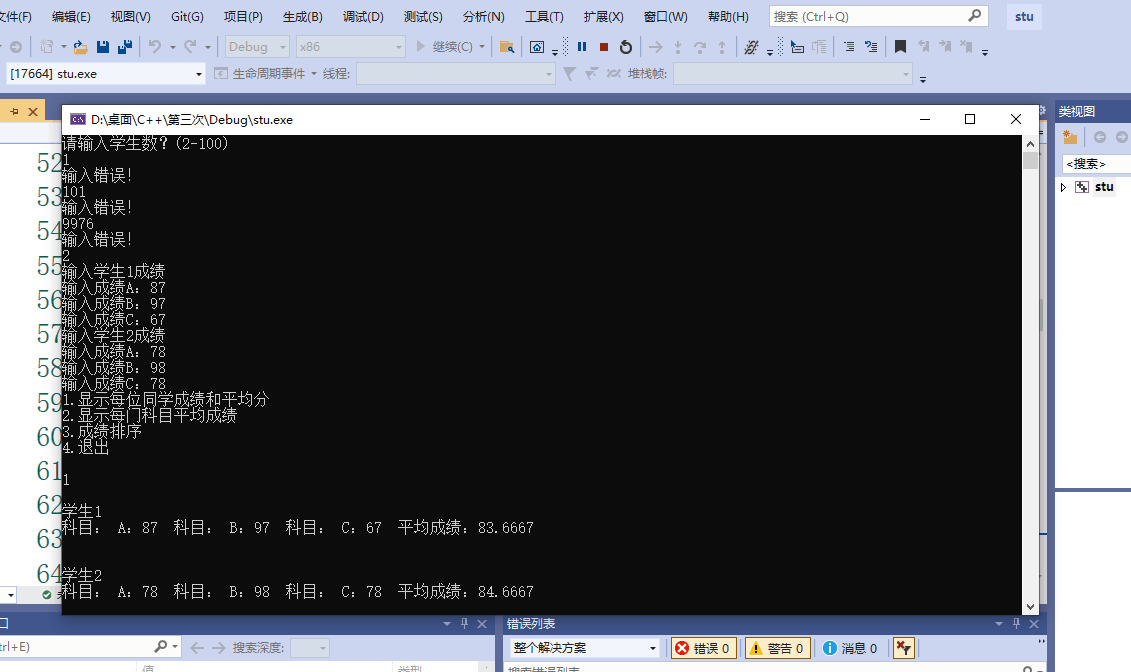
{

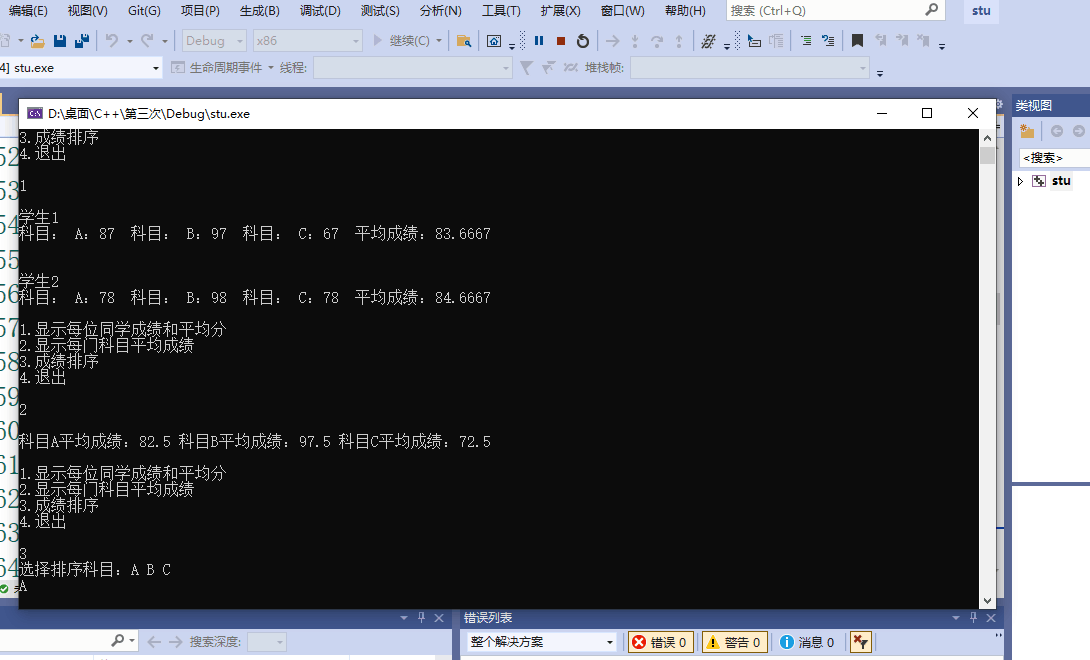
input();

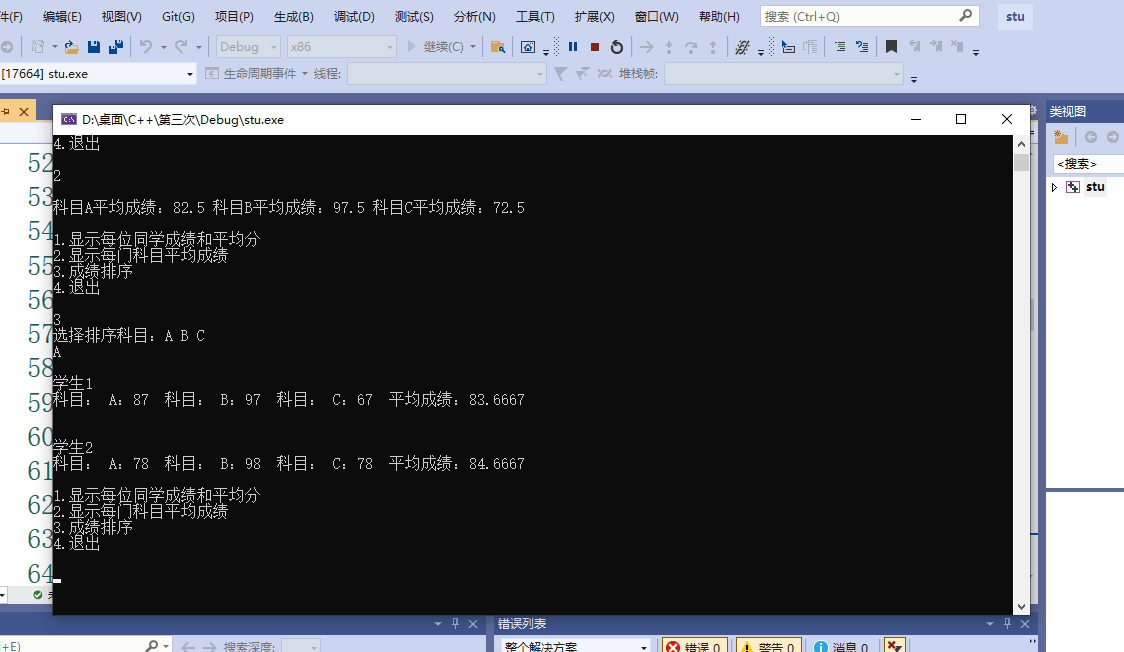
menu();

}

2.运行结果







3.实验总结：

本次实验较于前两次难度提升许多，让我充分的认识到了自己的不足，也体会到了c++于c语言的巨大区别。

Copyright ©2021-2099chenhe.

All rights reserved