### 作业2

第九章课后题：

1.

#include<stdio.h>

const int mon[] = {0, 31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31}; //每月日期常量

struct Cal{

int year, month, day;

}date;

int check(){ //闰年判断 是二月+1

if(date.month > 2 && (date.year % 400 == 0 || date.year % 100 != 0 && date.year % 4 == 0))

return 1;

return 0;

}

int dy(){

int sum = date.day; //当前月份的天数

for(int i = 1; i < date.month; ++ i) sum += mon[i]; //

sum += check(); //加闰年判断

return sum;

}

int main( )

{

puts("请输入年月日：");

scanf("%d-%d-%d", &date.year, &date.month, &date.day);

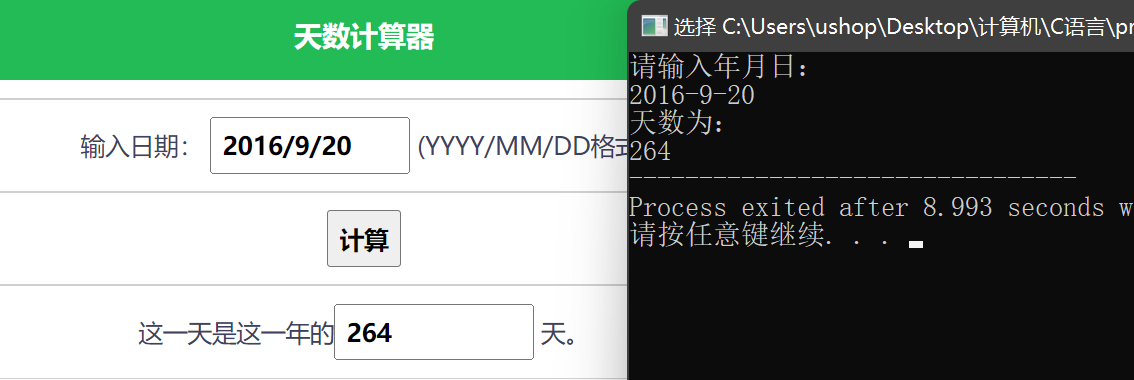
puts("天数为：");

printf("%d", dy());

return 0;

}





2.

在第一题的基础上简单修改即可

#include<stdio.h>

const int mon[] = {0, 31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31}; //每月日期常量

struct Cal{

int year, month, day;

}date;

int check(){ //闰年判断 是二月+1

if(date.month > 2 && (date.year % 400 == 0 || date.year % 100 != 0 && date.year % 4 == 0))

return 1;

return 0;

}

int days(int year, int month, int day){

int sum = day;

for(int i = 1; i < month; ++ i) sum += mon[i];

sum += check();

return sum;

}

int main( )

{

puts("请输入年月日：");

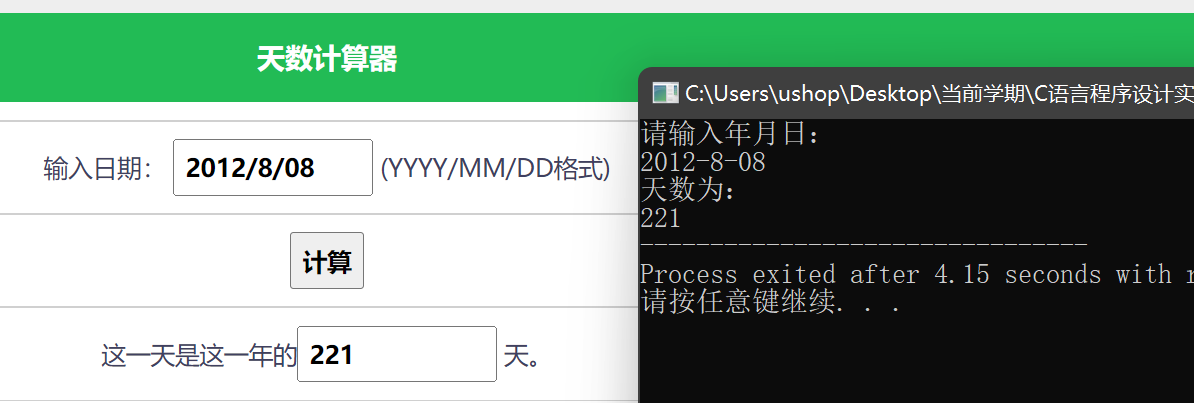
scanf("%d-%d-%d", &date.year, &date.month, &date.day);

puts("天数为：");

printf("%d", days(date.year, date.month, date.day));

return 0;

}



3.

#include<stdio.h>

const int N = 5;

struct Student{

int num;

char name[20];

int score[3];

}stu[N];

void print(){

puts("\n NO. name score1 score2 score3");

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

printf("%5d%10s", stu[i].num, stu[i].name);

printf("%10d%10d%10d\n", stu[i].score[0], stu[i].score[1], stu[i].score[2]);

}

return;

}

int main( )

{

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

scanf("%d %s", &stu[i].num, stu[i].name);

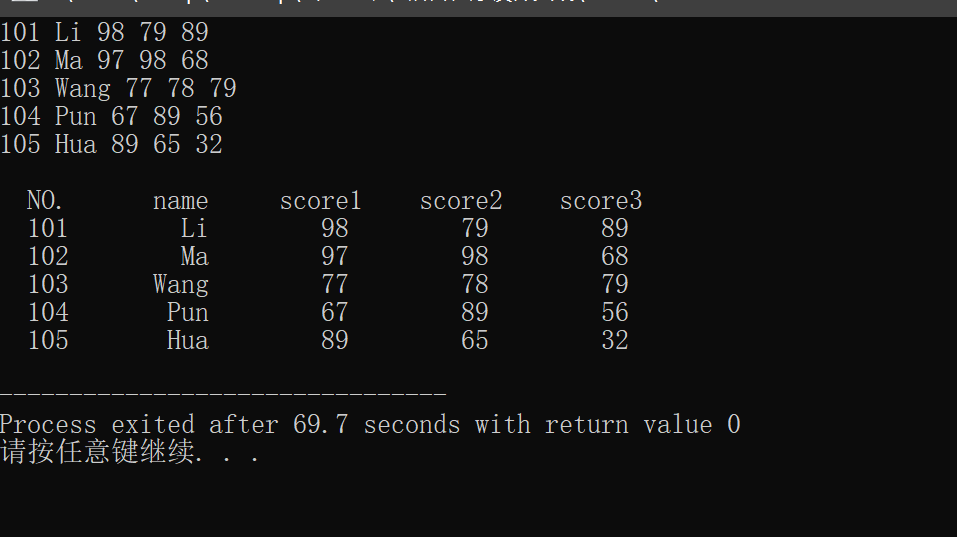
scanf("%d %d %d", &stu[i].score[0], &stu[i].score[1], &stu[i].score[2]);

}

print();

return 0;

}



4.

#include<stdio.h>

const int N = 5;

struct Student {

int num;

char name[20];

int score[3];

} stu[N];

void input() {

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

scanf("%d %s", &stu[i].num, stu[i].name);

scanf("%d %d %d", &stu[i].score[0], &stu[i].score[1], &stu[i].score[2]);

}

return;

}

void print() {

puts("\n NO. name score1 score2 score3");

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

printf("%5d%10s", stu[i].num, stu[i].name);

printf("%10d%10d%10d\n", stu[i].score[0], stu[i].score[1], stu[i].score[2]);

}

return;

}

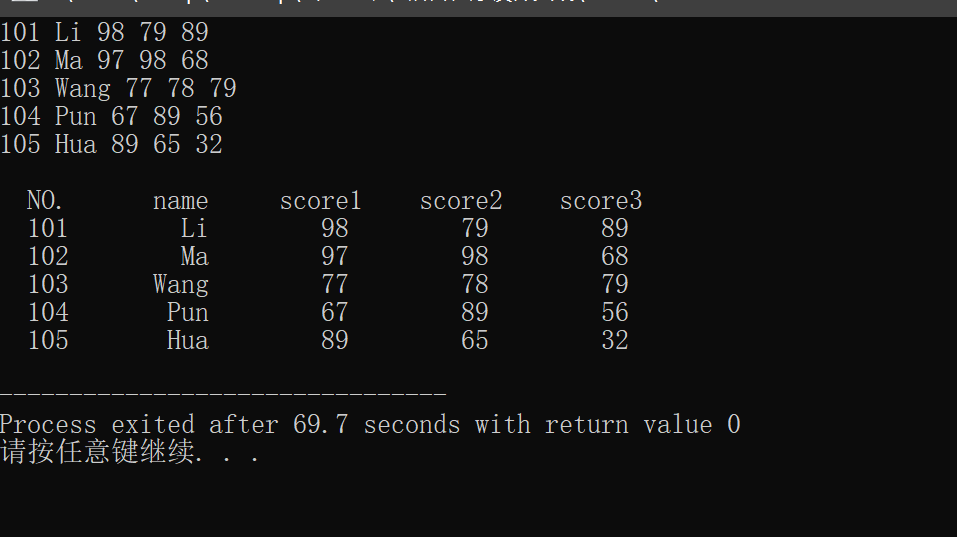
int main( ) {

input();

print();

return 0;

}



5.

#include<stdio.h>

const int N = 10;

double aver;

int sum, maxp;

struct Student {

int num;

char name[20];

int score[3];

} stu[N];

int max(int a, int b, int c){

if(a < b) a = b;

if(a < c) a = c;

return a;

}

void input() {

int maxss = 0;

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

scanf("%d %s", &stu[i].num, stu[i].name);

scanf("%d %d %d", &stu[i].score[0], &stu[i].score[1], &stu[i].score[2]);

sum += stu[i].score[0] + stu[i].score[1] + stu[i].score[2];

int maxs = max(stu[i].score[0], stu[i].score[1], stu[i].score[2]); //记录三个数中的最大值

if(maxs > maxss){

maxss = maxs;

maxp = i; //记录最高分的序号

}

}

return;

}

void print() {

puts("\n NO. name score1 score2 score3");

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

printf("%5d%10s", stu[i].num, stu[i].name);

printf("%10d%10d%10d\n", stu[i].score[0], stu[i].score[1], stu[i].score[2]);

}

printf("Average = %.2f\n", sum \* 1.0 / 3 / N); //平均分计算

printf("The highest is : %d %s\n", stu[maxp].num, stu[maxp].name);

printf("His scores are: %3d %3d %3d", stu[maxp].score[0], stu[maxp].score[1], stu[maxp].score[2]);

return;

}

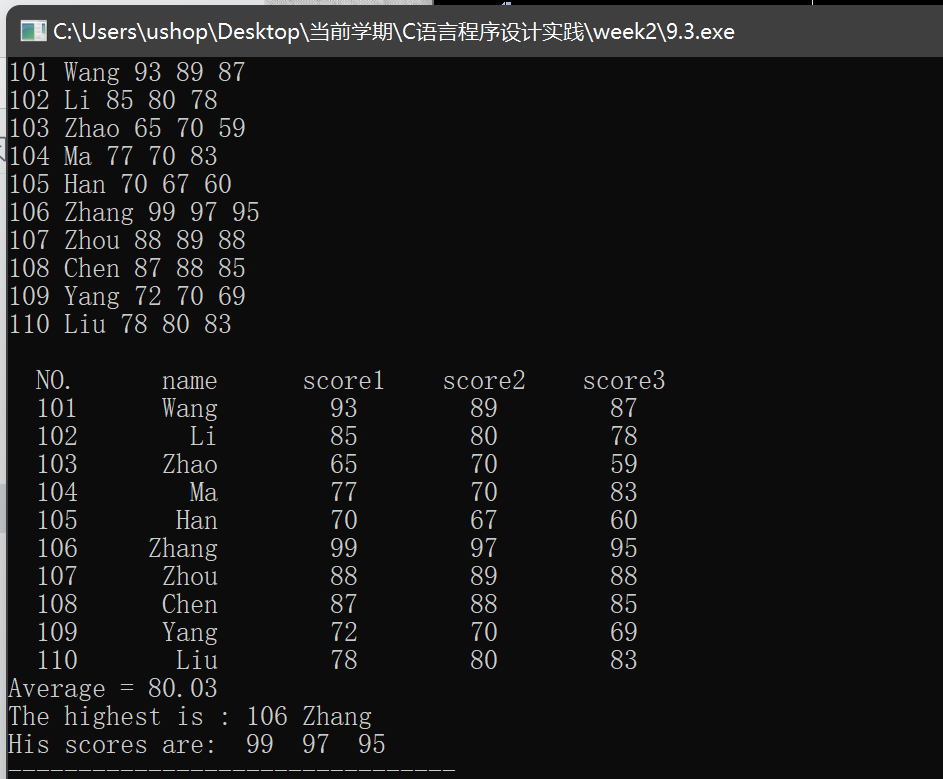
int main( ) {

input();

print();

return 0;

}



案例1：

//写文件

#include<stdio.h>

#include<process.h>

#include<stdlib.h>

#define SIZE 9

const int N = 3; //员工资料数

int Js, tax[SIZE], salary[N]; //级数、税率、工资

long right[SIZE], deduct[SIZE]; //上限 速算扣除率

char name[N][10];

void acceptdata(FILE \*fp1, FILE \*fp2) { //文件1存收税等级、文件2存员工数据

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

printf("Please enter data:");

printf("级数：");

scanf("%d", &Js);

scanf("%ld",&right[i]);

scanf("%d",&tax[i]);

scanf("%ld",&deduct[i]);

}

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

printf("Please enter data:");

scanf("%s%d", name[i], &salary[i]);

}

if (fwrite(right,sizeof(right[0]),SIZE,fp1)!=SIZE) //检测写入是否正常

printf ("file write error\n");

if (fwrite(tax,sizeof(tax[0]),SIZE,fp1)!=SIZE)

printf ("file write error\n");

if (fwrite(deduct,sizeof(deduct[0]),SIZE,fp1)!=SIZE)

printf ("file write error\n");

if (fwrite(name,sizeof(name[0]),N,fp2)!=N)

printf ("file write error\n");

if (fwrite(salary,sizeof(salary[0]),N,fp2)!=N)

printf ("file write error\n");

}

int main() {

FILE \*fp1, \*fp2;

if ((fp1=fopen("d:\\TAX.bin", "wb"))==NULL) { //打开文件并检测是否正常

printf("\ncannot open file\n");

exit(1);

}

if ((fp2=fopen("d:\\Salary.bin", "wb"))==NULL) {

printf("\ncannot open file\n");

exit(1);

}

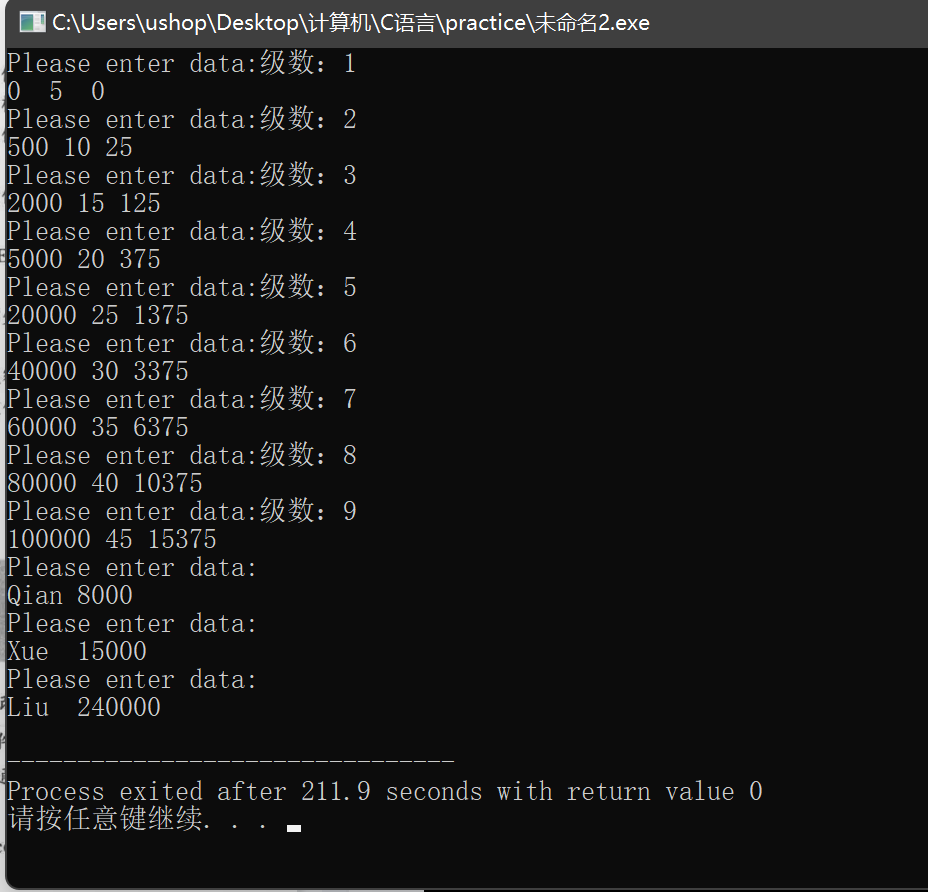
acceptdata(fp1, fp2);

fclose(fp1);

fclose(fp2);

return 0;

}



//读文件

#include<stdio.h>

#include<process.h>

#include<stdlib.h>

#define SIZE 9

const int N = 3;

int ff, tax[SIZE], salary[N];

long right[SIZE], deduct[SIZE];

char name[N][10];

void disp() {

double sal,s,taxx;

int i; //可考虑不用浮点数

printf("请输入税前扣除额：");

scanf("%d",&ff);

printf("请输入月收入：");

scanf("%lf",&sal);

//<0会如何？

s=sal-ff;

if(s<=0) taxx=0;

else {

for(i = 0; i < SIZE - 1; ++ i) { //用SIZE而不是具体值

if(s < right[i + 1] ) {

taxx=s\*tax[i]/100-deduct[i];

break;

} //应该加break

}

if(i >= SIZE - 1) taxx=s\*tax[8]/100-deduct[8];

}

printf("应纳个人所得税额是：%.2lf\n",taxx);

}

void disp1(FILE \*fp1, FILE \*fp2) {

puts("员工纳税：");

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

double sal = salary[j],s,taxx;

int i; //可考虑不用浮点数

s=sal-ff;

if(s<=0) taxx=0;

else {

for(i = 0; i < SIZE - 1; ++ i) { //用SIZE而不是具体值

if(s < right[i + 1]) {

taxx=s\*tax[i]/100-deduct[i];

break;

} //应该加break

//printf("%d %d %d\n", right[i], tax[i], deduct[i]);

}

if(i >= SIZE - 1) taxx=s\*tax[8]/100-deduct[8];

}

fprintf(fp1, "%2lf", taxx);

printf("%s应纳个人所得税额是：%.2lf\n",name[j], taxx);

}

}

int main() {

//读文件

FILE \*fp1, \*fp2;

if ((fp1=fopen("d:\\TAX.bin","rb"))==NULL) {

printf("\ncannot open file\n");

exit(1);

}

if ((fp2=fopen("d:\\Salary.bin","rb"))==NULL) {

printf("\ncannot open file\n");

exit(1);

}

if (fread(right,sizeof(right[0]),SIZE,fp1)!=SIZE) {

printf ("file read error\n");

exit(1);

}

if (fread(tax,sizeof(tax[0]),SIZE,fp1)!=SIZE) {

printf ("file read error\n");

exit(1);

}

if (fread(deduct,sizeof(deduct[0]),SIZE,fp1)!=SIZE) {

printf ("file read error\n");

exit(1);

}

if (fread(name,sizeof(name[0]),N,fp2)!=N) {

printf ("file read error\n");

exit(1);

}

if (fread(salary,sizeof(salary[0]),N,fp2)!=N) {

printf ("file read error\n");

exit(1);

}

disp(); //调用计算并显示所得税程序

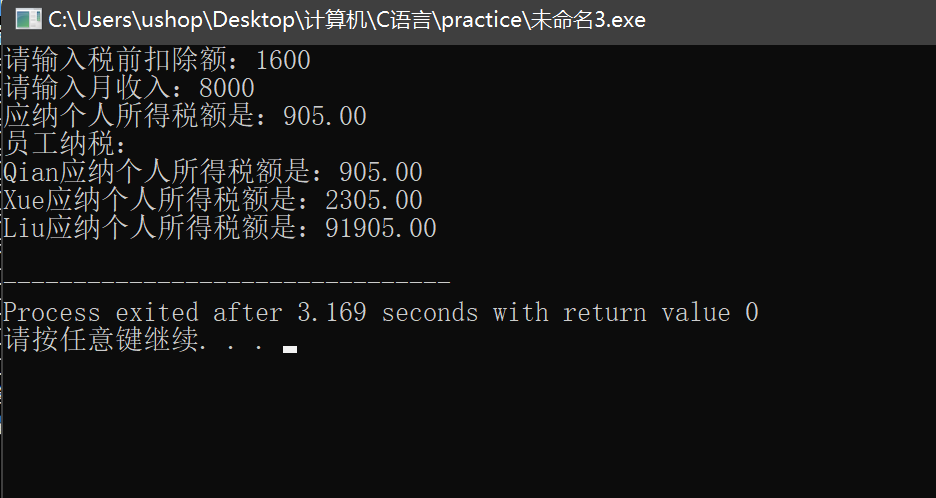
disp1(fp1, fp2);

fclose(fp1);

fclose(fp2);

return 0;

}



另外的优化：

循环文件时，终止条件应该用SIZE替换具体数值。因为国家政策改变，级数也会相对应地改变，用SIZE的话更改更方便。

确定级数时循环应当加上break，否则会一直满足判断条件，导致所有的数据最终都是最高级数，造成答案出错。

新建员工文件，方便了统一写入数据和计算。

如果工资减去起征点是负数也不用另外分类，因为已经包含在了if的判断条件里，所得税直接置零就可以了。

用二维数组，行存员工号，纵存该员工各种数据会更简单。

案例2：

1&2&3：输入、调试、改进、完善：

变量sum表示的是有效输入量，增加了sun = i的赋值语句使之能正常运行。

#include<stdio.h>

#include<string.h>

#include<conio.h>

#include<math.h>

#define SIZE 5

typedef struct student {

int number; //都是整型数

int score[3]; //平时、期末、总评

} STUDENT;

typedef enum boolen {

False, True //判断输入成绩是否错误用

} FLAG;

int accept\_data(STUDENT stu[], int grade[]);

void show\_data(STUDENT stu[], int sum, int grade[]);

void draw(int grade[]);

void count(int \*max, int \*min, double \*pass, double ave[], double f[], STUDENT stu[], int sum);

void show\_data2(int max, int min, double pass, double ave[], double f[]);

int main() {

int sum, max, min;

double pass = 0;

int grade[11] = {0}; //10分数段统计

STUDENT stu[SIZE];

double ave[SIZE], f[SIZE]; //求平均成绩、标准差用

sum = accept\_data(stu, grade); //输入

show\_data(stu, sum, grade); //显示

draw(grade); //画直方图

count(&max, &min, &pass, ave, f, stu, sum); //统算

show\_data2(max, min, pass, ave, f); //显示

return 0;

}

int accept\_data(STUDENT stu[], int grade[]) {

int i = 0, sum = 0, temp, a1, a2;

FLAG flag;

printf("请输入计算总评成绩时使用平时成绩的比例，用整数表示");

scanf("%d%d", &a1, &a2);

while (i < SIZE) {

printf("\n请输入学号：");

scanf("%d", &stu[i].number);

if (stu[i].number == -1) {

sum = i;

break;

}

printf("\n请输入学生的平时成绩和期末成绩：");

flag = True;

while (flag == True) {

scanf("%d%d", &stu[i].score[0], &stu[i].score[1]);

if (stu[i].score[0] <= 100 && stu[i].score[0] >= 0 && stu[i].score[1] <= 100 && stu[i].score[1] >= 0)

flag = False;

else

printf("\n\007错误数据！请再次输入学生的平时成绩和期末成绩：");

}

temp = (int)(1.0 \* a1 / 100 \* stu[i].score[0] + 1.0 \* a2 / 100 \* stu[i].score[1]);

stu[i].score[2] = temp;

temp = (stu[i].score[1]) / 10;//printf("%d\n", temp);

if (temp == 10)

grade[10] ++;

else

grade[temp + 1] ++;

++ i;

sum = i; //sum代表有效输入，应该加上这一语句

}

return sum;

}

void show\_data(STUDENT stu[], int sum, int grade[]) {

int i, j;

for (i = 0; i < sum; i++) {

printf("%4d", stu[i].number);

for (j = 0; j < 3; j++) {

printf("%4d", stu[i].score[j]);

}

printf("\n");

}

printf("\n分数段分布情况：\n");

for (i = 1; i <= 10; i++) {

printf(" %d", grade[i]);

}

}

void count(int \*max, int \*min, double \*pass, double ave[], double f[], STUDENT stu[], int sum) {

int i, j, p\_sum = 0;

int total[3];

\*max = \*min = stu[0].score[1];

if (stu[0].score[1] >= 60)

p\_sum++;

for (i = 1; i < sum; i++) {

if ((stu[i].score[1]) > \*max)

\*max = stu[i].score[1];

if ((stu[i].score[1]) < \*min)

\*min = stu[i].score[1];

if (stu[i].score[1] >= 60)

p\_sum++;

}

\*pass = (1.0 \* p\_sum / sum) \* 100;

for (i = 0; i <= 2; i++)

total[i] = 0;

for (j = 0; j < 3; j++) {

for (i = 0; i < sum; i++) {

total[j] = total[j] + stu[i].score[j];

}

}

for (j = 0; j < 3; j++)

ave[j] = total[j] / sum;

for (j = 0; j < 3; j++) {

f[j] = 0;

for (i = 0; i < sum; i++) {

double temp = stu[i].score[j] - ave[j];

f[j] = f[j] + temp \* temp;

}

f[j] = sqrt(fabs(f[j]) / sum);

}

}

void show\_data2(int max, int min, double pass, double ave[], double f[]) {

int j;

char str1[3][20] = {"平时成绩平均分", "期末成绩平均分", "总评成绩平均分"};

char str2[3][20] = {"平时成绩标准差", "期末成绩标准差", "总评成绩标准差"};

printf("\n及格率=%6.2f%% 最高分=%d 最低分=%d\n", pass,max,min);

for (j = 0; j < 3; j++)

printf("\n%s=%6.2f %s=%6.2f\n", str1[j], ave[j], str2[j], f[j]);

}

void draw(int grade[]) {

int i, j, max, k, temp, x;

char screen[22][44];

printf("\n\n模拟直方图\n");

max = 0;

for (i = 0; i <= 10; i++)

if (grade[i] > max)

max = grade[i];

for (i = 1; i <= 10; i++)

grade[i] = (int)(20.0 \* grade[i] / max + 0.5);

for (i = 0; i <= 21; i++) {

for (j = 0; j <= 41; j++)

screen[i][j] = 0;

}

for(i=0; i<=41; i++)

screen[21][i]='——';

screen[21][41]='X';

screen[0][0]='Y';

for(i=1; i<=21; i++)

screen[i][0]='|';

k=1;

for(x=1; x<=10; x++,k=k+4) {

temp=grade[x];

if(temp!=0) {

for(i=1; i<=temp; i++) {

for(j=1; j<=4; j++) {

screen[20-i+1][j+k]='\*'; //printf("%d %d\n", 20 - i + 1, j + k);

}

}

}

}

for(i=0; i<=21; i++) {

for(j=0; j<=41; j++) {

if(screen[i][j]!=0)

printf("%c",screen[i][j]);

else

printf(" ");

}

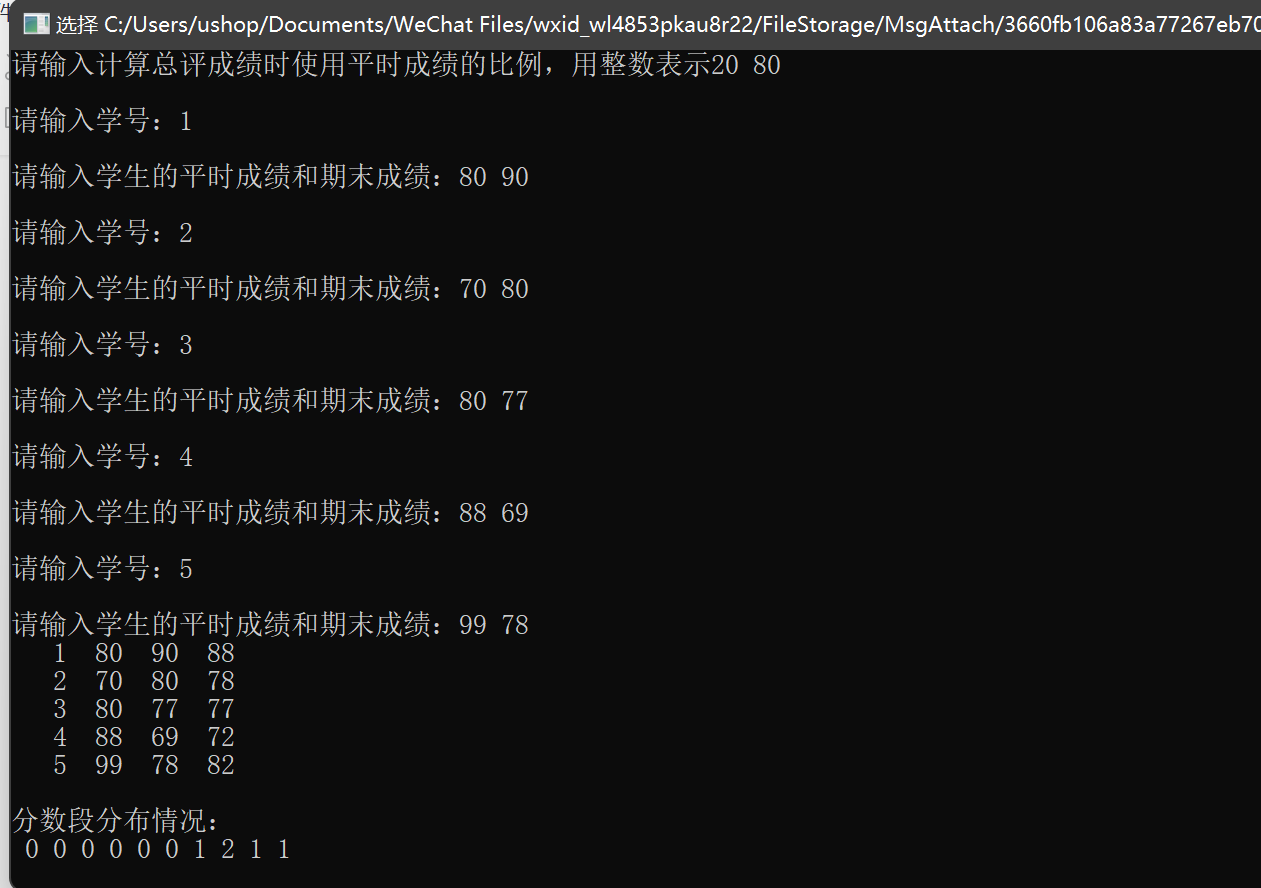
printf("\n");

}

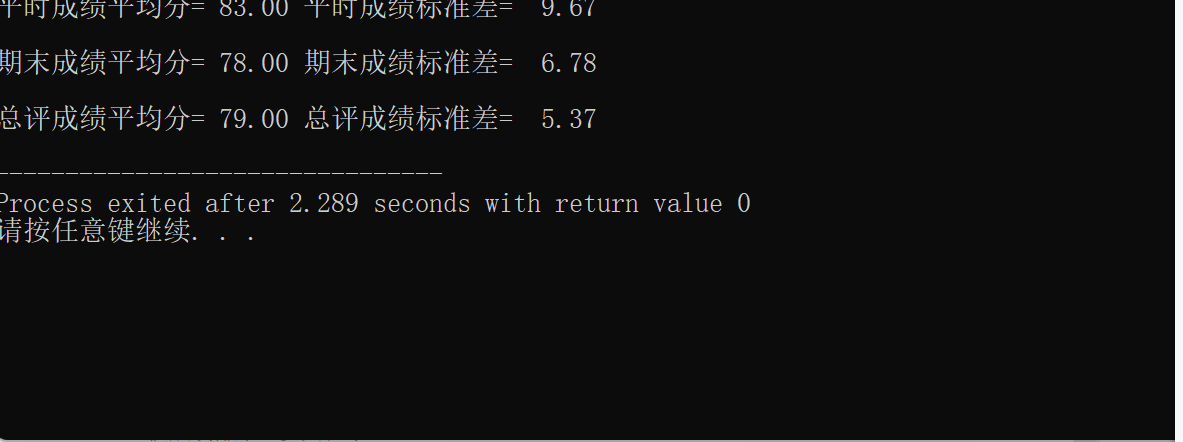
printf("0 10 20 30 40 50 60 70 80 90 100\n");

getch();

}







主函数输入数据和统计数据存入磁盘：

int main() {

FILE \*fp1, \*fp2;

int sum, max, min;

double pass = 0;

int grade[11] = {0}; //10分数段统计

STUDENT stu[SIZE];

double ave[SIZE], f[SIZE]; //求平均成绩、标准差用

sum = accept\_data(stu, grade); //输入

show\_data(stu, sum, grade); //显示

draw(grade); //画直方图

count(&max, &min, &pass, ave, f, stu, sum); //统算

show\_data2(max, min, pass, ave, f); //显示

if((fp1 = fopen("d://STU.bin", "wb")) == NULL) printf("can not open file!\n");

if(fwrite(stu, sizeof(STUDENT), SIZE, fp1) != SIZE) printf ("file write error\n");

if((fp2 = fopen("d://STUdata.bin", "w")) == NULL) printf("can not open file!\n");

for(int i = 0; i < 11; ++ i) fprintf(fp2, "%d", grade[i]);

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

fprintf(fp2, "%.1f", ave[i]);

fprintf(fp2, "%.1f", f[i]);

}

fclose(fp1);

fclose(fp2);

return 0;

}

增加期中成绩：

int accept\_data(STUDENT stu[], int grade[]) {

int i = 0, sum = 0, temp, a1, a2, a3;

FLAG flag;

printf("请输入计算总评成绩时使用平时成绩的比例，用整数表示");

scanf("%d%d%d", &a1, &a2, &a3);

while (i < SIZE) {

printf("\n请输入学号：");

scanf("%d", &stu[i].number);

if (stu[i].number == -1) {

sum = i;

break;

}

printf("\n请输入学生的平时成绩、期中成绩和期末成绩：");

flag = True;

while (flag == True) {

scanf("%d%d%d", &stu[i].score[0], &stu[i].score[1], &stu[i].score[2]);

if (stu[i].score[0] <= 100 && stu[i].score[0] >= 0 && stu[i].score[1] <= 100 && stu[i].score[1] && stu[i].score[2] <= 100 && stu[i].score[2]>= 0)

flag = False;

else

printf("\n\007错误数据！请再次输入学生的平时成绩、期中成绩和期末成绩：");

}

stu[i].total = (1.0 \* a1 / 100 \* stu[i].score[0] + 1.0 \* a2 / 100 \* stu[i].score[1] + 1.0 \* a3 / 100 \* stu[i].score[2]);

temp = (stu[i].score[2]) / 10;

if (temp == 10)

grade[10] ++;

else

grade[temp + 1] ++;

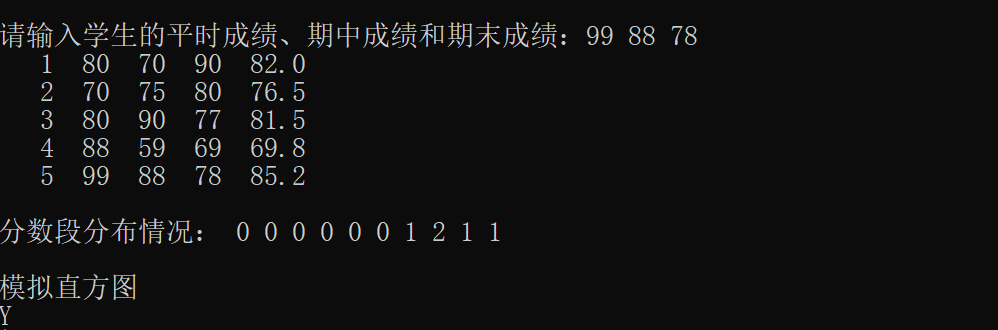
++ i;

sum = i;

}

return sum;

}



各部分标准差：

void count(int \*max, int \*min, double \*pass, double ave[], double f[], STUDENT stu[], int sum) {

int i, j, p\_sum = 0;

int total[3];

\*max = \*min = stu[0].score[1];

if (stu[0].score[1] >= 60)

p\_sum++;

for (i = 1; i < sum; i++) {

if ((stu[i].score[1]) > \*max)

\*max = stu[i].score[1];

if ((stu[i].score[1]) < \*min)

\*min = stu[i].score[1];

if (stu[i].score[1] >= 60)

p\_sum++;

}

\*pass = (1.0 \* p\_sum / sum) \* 100;

for (i = 0; i <= 2; i++)

total[i] = 0;

for (j = 0; j < 3; j++) {

for (i = 0; i < sum; i++) {

total[j] = total[j] + stu[i].score[j];

}

}

for (j = 0; j < 3; j++)

ave[j] = total[j] / sum;

for (j = 0; j < 3; j++) {

f[j] = 0;

for (i = 0; i < sum; i++) {

double temp = stu[i].score[j] - ave[j];

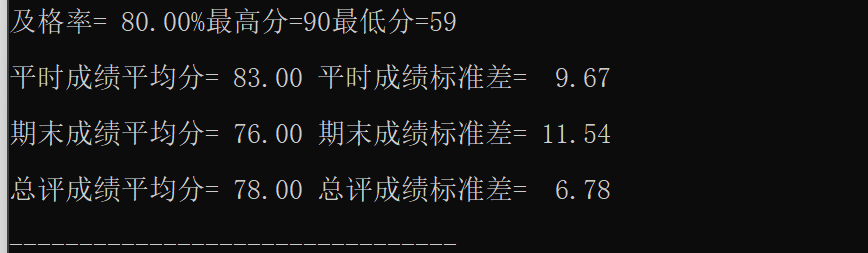
f[j] = f[j] + temp \* temp;

}

f[j] = sqrt(fabs(f[j]) / sum);

}

}



多关键字排序：

按照总评成绩由小到大排序，当总评成绩相同时，按照学号从小到大排序。

主要代码：

int cmp(const void \*a,const void \*b)

{

STUDENT \*aa = (STUDENT \*)a;

STUDENT \*bb = (STUDENT \*)b;

if(fabs(aa->total - bb->total) > 1e-6) return (((aa->total) > (bb->total)) > 1e-6 ? 1 : -1);

return aa->number > bb->number;

}

void show\_data(STUDENT stu[], int sum, int grade[]) {

int i, j;

qsort(stu, SIZE, sizeof(STUDENT), cmp);

for (i = 0; i < sum; i++) {

printf("%4d", stu[i].number);

for (j = 0; j < 3; j++) {

printf("%4d", stu[i].score[j]);

}

printf(" %.1f", stu[i].total);

printf("\n");

}

printf("\n分数段分布情况：");

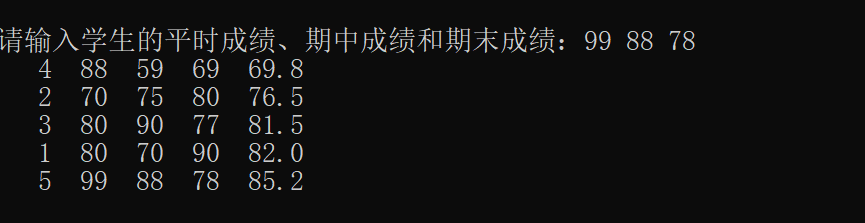
for (i = 1; i <= 10; i++) {

printf(" %d", grade[i]);

}

}

效果截图：



另：

将总评改为浮点数，表示更合理。

增加了“分数段分布情况”的提示字样。

修改了直方图的输出格式，还有各个模块间的间隔，使显示更加直观。