1.1 按照学号递增输出全部学生信息，每个学生的信息一行。

struct student

{

int num;

char name[20];

int scor1;

int scor2;

int scor3;

int sum;

};

#define N 5

int main(){

struct student arr[5];

int res;

memset(&arr, 0, sizeof(arr));

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

res = scanf("%d%s%d%d%d", &arr[i].num, arr[i].name, &arr[i].scor1, &arr[i].scor2, &arr[i].scor3);

arr[i].sum = arr[i].scor1 + arr[i].scor2 + arr[i].scor3;

}

int i, j;

struct student temp;

for (int i = 4; i > 0; i--) {

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

if (arr[j].num > arr[j +1].num) {

temp = arr[j];

arr[j] = arr[j +1];

arr[j + 1] = temp;

}

}

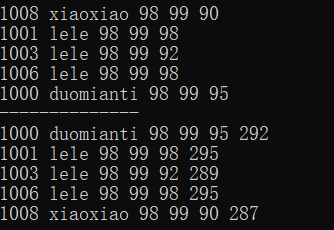
}

printf("--------------\n");

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

printf("%d %s %d %d %d %d\n", arr[i].num, arr[i].name, arr[i].scor1, arr[i].scor2, arr[i].scor3, arr[i].sum);

}

}

1.2

输出每门课程最高分的学生的信息

struct student

{

int num;

char name[20];

int scor1;

int scor2;

int scor3;

int sum;

};

#define N 5

int main(){

struct student arr[5];

int res;

memset(&arr, 0, sizeof(arr));

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

res = scanf("%d%s%d%d%d", &arr[i].num, arr[i].name, &arr[i].scor1, &arr[i].scor2, &arr[i].scor3);

arr[i].sum = arr[i].scor1 + arr[i].scor2 + arr[i].scor3;

}

int i, j;

struct student temp;

// scor1 最高分排

for (int i = 4; i > 0; i--) {

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

if (arr[j].scor1 > arr[j +1].scor1) {

temp = arr[j];

arr[j] = arr[j +1];

arr[j + 1] = temp;

}

}

}

printf("--------------\n");

i = 4;

printf("scor1 得分最高的同学的信息：\n");

printf("%d %s %d %d %d %d\n", arr[i].num, arr[i].name, arr[i].scor1, arr[i].scor2, arr[i].scor3, arr[i].sum);

// scor2 最高分排

for (int i = 4; i > 0; i--) {

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

if (arr[j].scor2 > arr[j + 1].scor2) {

temp = arr[j];

arr[j] = arr[j + 1];

arr[j + 1] = temp;

}

}

}

printf("--------------\n");

printf("scor2 得分最高的同学的信息：\n");

printf("%d %s %d %d %d %d\n", arr[i].num, arr[i].name, arr[i].scor1, arr[i].scor2, arr[i].scor3, arr[i].sum);

// scor2 最高分排

for (int i = 4; i > 0; i--) {

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

if (arr[j].scor3 > arr[j + 1].scor3) {

temp = arr[j];

arr[j] = arr[j + 1];

arr[j + 1] = temp;

}

}

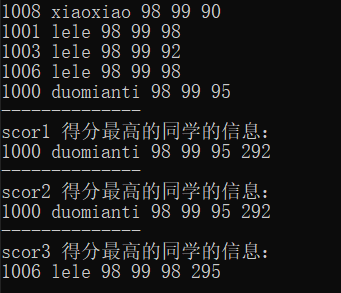
}

printf("--------------\n");

printf("scor3 得分最高的同学的信息：\n");

printf("%d %s %d %d %d %d\n", arr[i].num, arr[i].name, arr[i].scor1, arr[i].scor2, arr[i].scor3, arr[i].sum);

}



1.3 求每门课的平均值

//

//#define N 5

//int main(){ //1.3

// struct student arr[5];

// int res;

// memset(&arr, 0, sizeof(arr));

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

// res = scanf("%d%s%f%f%f", &arr[i].num, arr[i].name, &arr[i].scor1, &arr[i].scor2, &arr[i].scor3);

// //arr[i].sum = arr[i].scor1 + arr[i].scor2 + arr[i].scor3;

// }

// int i, j;

// // scor1 平均分

// float sum = 0;

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

// sum += arr[i].scor1;

// }

// printf("scor1这门课的平均分是： %.2f\n", sum / 5);

//

// // scor2 平均分

// sum = 0;

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

// sum += arr[i].scor2;

// }

// printf("scor2这门课的平均分是： %.2f\n", sum / 5);

//

// // scor3 平均分

// sum = 0;

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

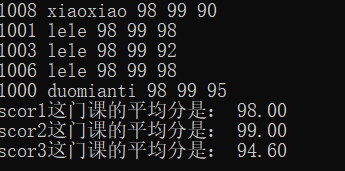
// sum += arr[i].scor3;

// }

// printf("scor3这门课的平均分是： %.2f\n", sum / 5);

//

//}



1.4 按总分排名

struct student //1.3

{

int num;

char name[20];

float scor1;

float scor2;

float scor3;

float sum;

};

#define N 5

int main(){

struct student arr[5];

int res;

memset(&arr, 0, sizeof(arr));

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

res = scanf("%d%s%f%f%f", &arr[i].num, arr[i].name, &arr[i].scor1, &arr[i].scor2, &arr[i].scor3);

arr[i].sum = arr[i].scor1 + arr[i].scor2 + arr[i].scor3;

}

int i, j;

struct student temp;

// scor1 最高分排

for (int i = 4; i > 0; i--) {

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

if (arr[j].sum > arr[j +1].sum) {

temp = arr[j];

arr[j] = arr[j +1];

arr[j + 1] = temp;

}

}

}

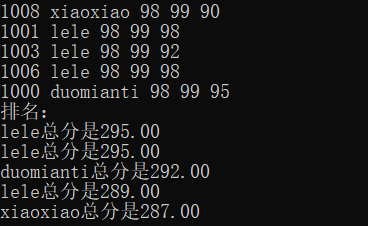
printf("排名：\n");

for (int i = N - 1; i >= 0; i--) {

printf("%s总分是%5.2f\n", arr[i].name, arr[i].sum);

}

}



2 尾插

typedef struct student {

int num;

float score;

struct student\* pNext;

}stu,\*pstu;

typedef int INTEGER;

void list\_tail\_insert(pstu\* pphead, stu\*\* pptail, int i) {

pstu pnew;

pnew = (pstu)malloc(sizeof(stu));

memset(pnew, 0, sizeof(stu));

pnew->num = i;

if (NULL == \*pptail) {

\*pphead = pnew;

\*pptail = pnew;

}

else {

(\*pptail)->pNext = pnew;

\*pptail = pnew;

}

}

void list\_print(pstu phead) {

while (phead != NULL)

{

printf("%3d", phead->num);

phead = phead->pNext;

}

printf("\n");

}

//main

int main() {

pstu p;

pstu phead = NULL, ptail = NULL; // 代表链表；

int i;

float f;

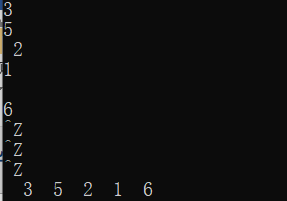
while (scanf("%d", &i) != EOF) { // 链表的头插，尾插，有序插入

list\_tail\_insert(&phead, &ptail, i);

}

list\_print(phead);

}



3 头插法

#define \_CRT\_SECURE\_NO\_WARNINGS

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

//定义结构体把小字节定义在一起，放到结构体最后

typedef struct student {

int num;

float score;

struct student\* pNext;

}stu,\*pstu;

typedef int INTEGER;

void list\_tail\_insert(pstu\* pphead, stu\*\* pptail, int i) {

pstu pnew;

pnew = (pstu)malloc(sizeof(stu));

memset(pnew, 0, sizeof(stu));

pnew->num = i;

if (NULL == \*pptail) {

\*pphead = pnew;

\*pptail = pnew;

}

else {

(\*pptail)->pNext = pnew;

\*pptail = pnew;

}

}

void list\_head\_insert(pstu\* pphead, stu\*\* pptail, int i) {

pstu pnew;

pnew = (pstu)malloc(sizeof(stu));

memset(pnew, 0, sizeof(stu));

pnew->num = i;

if (NULL == \*pptail) {

\*pphead = pnew;

\*pptail = pnew;

}

else {

pnew->pNext = \*pphead;

\*pphead = pnew;

}

}

void list\_print(pstu phead) {

while (phead != NULL)

{

printf("%3d", phead->num);

phead = phead->pNext;

}

printf("\n");

}

//main

int main() {

pstu p;

pstu phead = NULL, ptail = NULL; // 代表链表；

int i;

float f;

while (scanf("%d", &i) != EOF) { // 链表的头插，尾插，有序插入

list\_head\_insert(&phead, &ptail, i);

}

list\_print(phead);

}

