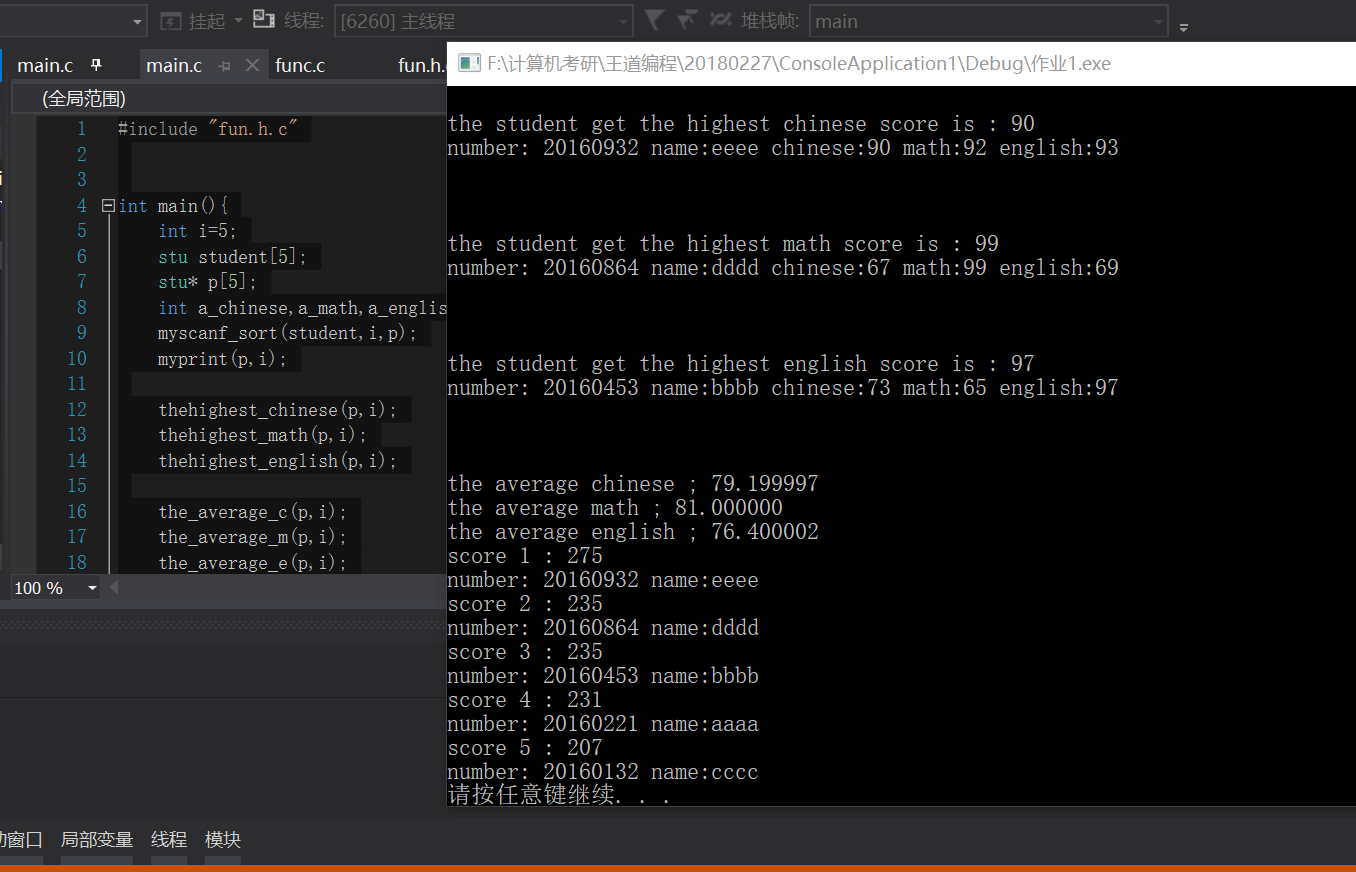
1.



**Funh.c**

#include <stdio.h>

#include <stdlib.h>

typedef struct student{

int number;

char name[20];

int chinese;

int math;

int english;

}stu;

void myscanf\_sort(stu student[],int n,stu\* p[]);

void myprint(stu\* p[],int n);

void thehighest\_chinese(stu\* p[],int n);

void thehighest\_math(stu\* p[],int n);

void thehighest\_english(stu\* p[],int n);

void the\_average\_c(stu\* p[],int n);

void the\_average\_m(stu\* p[],int n);

void the\_average\_e(stu\* p[],int n);

void the\_ranking(stu\* p[],int n);

**func.c**

#include "fun.h.c"

void myscanf\_sort(stu student[],int n,stu\* p[]){

int i=0,j;

stu\* temp;

while(i<n){

scanf("%d%s%d%d%d",&student[i].number,&student[i].name,&student[i].chinese,&student[i].math,&student[i].english);

i++;

}

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

p[j]=&student[j];

for(j=5;j>0;j--){

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

if(p[i]->number>p[i-1]->number){

temp=p[i-1];

p[i-1]=p[i];

p[i]=temp;

}

}

}

}

void myprint(stu\* p[],int n){

int i=4;

while(i>=0){

printf("number: %d name:%s chinese:%d math:%d english:%d \n",p[i]->number,p[i]->name,p[i]->chinese,p[i]->math,p[i]->english);

i--;

}

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

}

void thehighest\_chinese(stu\* p[],int n){

stu\* max=p[0];

int i;

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

if(max->chinese<p[i]->chinese)

max=p[i];

}

printf("the student get the highest chinese score is : %d \n",max->chinese);

printf("number: %d name:%s chinese:%d math:%d english:%d \n",max->number,max->name,max->chinese,max->math,max->english);

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

}

void thehighest\_math(stu\* p[],int n){

stu\* max=p[0];

int i;

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

if(max->math<p[i]->math)

max=p[i];

}

printf("the student get the highest math score is : %d \n",max->math);

printf("number: %d name:%s chinese:%d math:%d english:%d \n",max->number,max->name,max->chinese,max->math,max->english);

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

}

void thehighest\_english(stu\* p[],int n){

stu\* max=p[0];

int i;

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

if(max->english<p[i]->english)

max=p[i];

}

printf("the student get the highest english score is : %d \n",max->english);

printf("number: %d name:%s chinese:%d math:%d english:%d \n",max->number,max->name,max->chinese,max->math,max->english);

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

}

void the\_average\_c(stu\* p[],int n){

int i;

float sum=0;

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

sum+=p[i]->chinese;

printf("the average chinese ; %f\n",sum/n);

}

void the\_average\_m(stu\* p[],int n){

int i;

float sum=0;

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

sum+=p[i]->math;

printf("the average math ; %f\n",sum/n);

}

void the\_average\_e(stu\* p[],int n){

int i;

float sum=0;

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

sum+=p[i]->english;

printf("the average english ; %f\n",sum/n);

}

void the\_ranking(stu\* p[],int n){

int i,j;

int sum\_score;

stu\* temp;

sum\_score=p[0]->chinese+p[0]->english+p[0]->math;

for(j=n;j>0;j--){

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

if(sum\_score<(p[i]->chinese+p[i]->english+p[i]->math)){

temp=p[i-1];

p[i-1]=p[i];

p[i]=temp;

sum\_score=p[i]->chinese+p[i]->english+p[i]->math;

}

}

}

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

printf("score %d : %d\n",i+1,p[i]->chinese+p[i]->english+p[i]->math);

printf("number: %d name:%s \n",p[i]->number,p[i]->name);

}

}

**Main.c**

#include "fun.h.c"

int main(){

int i=5;

stu student[5];

stu\* p[5];

int a\_chinese,a\_math,a\_english;

myscanf\_sort(student,i,p);

myprint(p,i);

thehighest\_chinese(p,i);

thehighest\_math(p,i);

thehighest\_english(p,i);

the\_average\_c(p,i);

the\_average\_m(p,i);

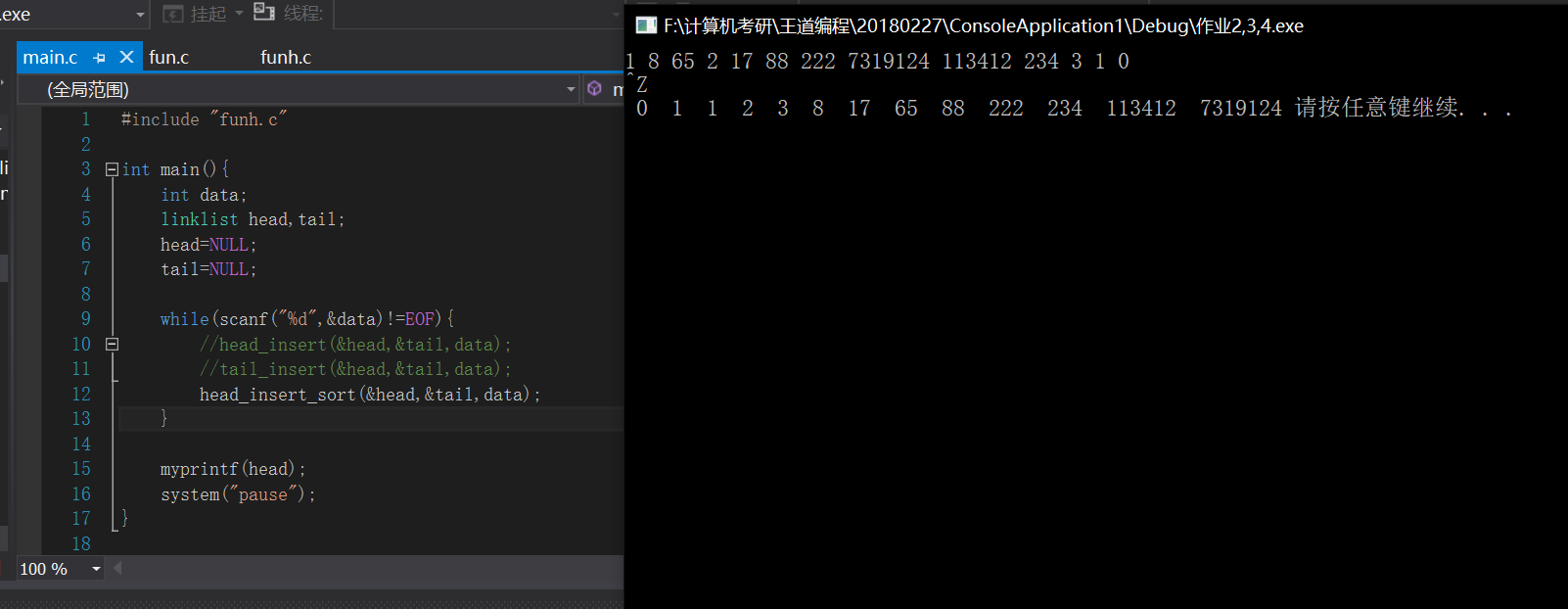
the\_average\_e(p,i);

the\_ranking(p,i);

system("pause");

}

**2.**



**Funh.c**

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

typedef struct line{

int num;

struct line\* next;

}lnode,\*linklist;

void head\_insert(linklist\* head,linklist\* tail,int i);

void tail\_insert(linklist\* head,linklist\* tail,int i);

void head\_insert\_sort(linklist\* head,linklist\* tail,int i);

void myprintf(linklist head);

**fun.c**

#include "funh.c"

void head\_insert(linklist\* head,linklist\* tail,int i){

linklist pnew;

pnew=(linklist)calloc(1,sizeof(lnode));

pnew->num=i;

if(\*head==NULL){

\*head=pnew;

\*tail=pnew;

}else{

pnew->next=\*head;

\*head=pnew;

}

}

void tail\_insert(linklist\* head,linklist\* tail,int i){

linklist pnew;

pnew=(linklist)calloc(1,sizeof(lnode));

pnew->num=i;

if(\*tail==NULL){

\*head=pnew;

\*tail=pnew;

}else{

(\*tail)->next=pnew;

\*tail=pnew;

}

}

void head\_insert\_sort(linklist\* head,linklist\* tail,int i){

linklist pnew;

linklist pcur,ppre;

pcur=\*head;

ppre=\*head;

pnew=(linklist)calloc(1,sizeof(lnode));

pnew->num=i;

if(pcur==NULL){

\*head=pnew;

\*tail=pnew;

}else if(pcur->num>i){

pnew->next=\*head;

\*head=pnew;

}else {

while(pcur!=NULL){

if(pcur->num>i){

ppre->next=pnew;

pnew->next=pcur;

break;

}

ppre=pcur;

pcur=pcur->next;

}

if(pcur==NULL){

(\*tail)->next=pnew;

\*tail=pnew;

}

}

}

void myprintf(linklist head){

while(head!=NULL){

printf(" %d ",head->num);

head=head->next;

}

}

**Main.c**

#include "funh.c"

int main(){

int data;

linklist head,tail;

head=NULL;

tail=NULL;

while(scanf("%d",&data)!=EOF){

//head\_insert(&head,&tail,data);

//tail\_insert(&head,&tail,data);

head\_insert\_sort(&head,&tail,data);

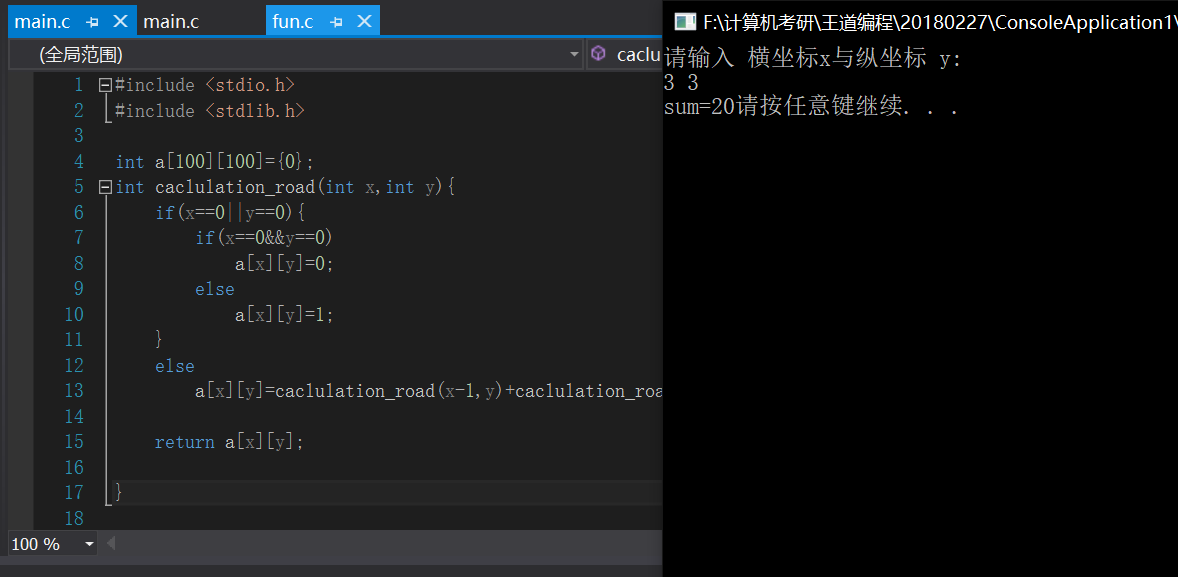
}

myprintf(head);

system("pause");

}

3.



#include <stdio.h>

#include <stdlib.h>

int a[100][100]={0};

int caclulation\_road(int x,int y){

if(x==0||y==0){

if(x==0&&y==0)

a[x][y]=0;

else

a[x][y]=1;

}

else

a[x][y]=caclulation\_road(x-1,y)+caclulation\_road(x,y-1);

return a[x][y];

}

int main(){

int x,y,sum=0;

printf("请输入 横坐标x与纵坐标 y:\n" );

scanf("%d %d",&x,&y);

sum=caclulation\_road(x,y);

printf("sum=%d",sum);

system("pause");

}