第一题

输入正整数 m 和方阵的阶次 n，且 m<=n\*n，参照输出样例的格式，按

照蛇形方式在方阵中填充从 0到 m-1的数字，数字用完后用“\*”代替，每行

数字或者'\*'之间用 Tab键分割，行末无 Tab键。

#include <stdio.h>

int main()

{

int i=0,x=0;

int j,k,m,n,a[100][100];

scanf("%d%d",&m,&n);

//把要输出的东西先存到二维数组

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

{

if(j%2==0) //0和偶数行是逆时针排序

{

for(k=n-1;k>=0;k--)

{

if(i<m)

{

a[j][k]=i;

i++;

}

else break; //超出m-1就跳出循环

}

}

else //顺时针排序

{

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

{

if(i<m)

{

a[j][k]=i;

i++;

}

else break;

}

}

}

//下面是输出二维数组过程

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

{

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

{

if(x<m) //未超出m-1，按照二维数组输出

{

printf("%d",a[j][k]);

x++;

printf("\t");

}

else

{

printf("\*\t");

}

}

printf("\n"); //一行结束，换行

}

return 0;

}

第二题

编写程序，定义一个 5×5的二维数组（数组元素为整数），按行读入数

组元素，计算数组最外层元素的平均数与次外层元素的平均数的差值（

结果

保留小数点后三位）。

/\*编写程序，定义一个 5×5的二维数组（数组元素为整数），按行读入数

组元素，计算数组最外层元素的平均数与次外层元素的平均数的差值（结果

保留小数点后三位）。\*/

#include <stdio.h>

int main()

{

int i,j,a[5][5],sum1,sum2;

float aver1,aver2;

for(i=0;i<5;i++)

for(j=0;j<5;j++)

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

for(i=0;i<5;i++) sum1=sum1+a[0][i]+a[4][i];

for(i=0;i<5;i++) sum1=sum1+a[i][0]+a[i][4];

sum1=sum1-a[0][0]-a[0][4]-a[4][0]-a[4][4];

aver1=sum1\*1.0/16;

for(i=1;i<4;i++) sum2=sum2+a[1][i]+a[3][i];

for(i=1;i<4;i++) sum2=sum2+a[i][1]+a[i][3];

sum2=sum2-a[1][1]-a[1][3]-a[3][1]-a[3][3];

aver2=sum2\*1.0/8;

printf("%.3f",aver1-aver2);

return 0;

}

第五题

/\*(1)定义考核人员信息的结构体 STAFF，包括编号（整数）、姓名（由不

含空格的 26个英文字母的大小写构成，不超过 10个字符）、年龄、5方面得

分和总分，其中得分和总分均为浮点数；\*/

#include <stdio.h>

struct STAFF

{

int num;

char name[10];

int age;

float score1,score2,score3,score4,score5,all;

}pt[5];

/\*（2）定义 voidinput(structSTAFF\*pt)函数，用来输入一位员工的信

息（编号、姓名和年龄）和各项得分，计算总分并保存；\*/

void input(struct STAFF \*pt)

{

scanf("%d%s%d%f%f%f%f%f",&pt->num,&pt->name,&pt->age,&pt->score1,&pt->score2,&pt->score3,&pt->score4,&pt->score5);

pt->all=pt->score1+pt->score2+pt->score3+pt->score4+pt->score5;

}

/\*（3）定义 voidoutput(structSTAFF\*pt)函数，用来输出一位员工的

编号、各项得分和总分（其中浮点数保留小数点后 2位）；\*/

void output(struct STAFF \*pt)

{

printf("%d %.2f %.2f %.2f %.2f %.2f %.2f",pt->num,pt->score1,pt->score2,pt->score3,pt->score4,pt->score5,pt->all);

}

/\*（4）定义 voidsort(structSTAFF\*pt)函数，对所有员工按总分由高

到低进行排序（得分相同保持原顺序）；\*/

void sort(struct STAFF \*pt)

{

int i,j,k;

struct STAFF temp;

for(i=0;i<4;i++)

{

k=i;

for(j=i+1;j<5;j++)

{

if(pt[j].all>pt[k].all) k=j;

}

if(k!=i)

{

temp=pt[i];

pt[i]=pt[k];

pt[k]=temp;

}

}

}

int main()

{

int i;

for(i=0;i<5;i++) input(&pt[i]);

sort(pt);

for(i=0;i<5;i++)

{

printf("Number%d:",i+1);

output(&pt[i]);

printf("\n");

}

return 0;

}

101 Louise 28 83 73 86 75 87

102 Tom 26 85 82 93 86 90

103 Cyan 25 73 81 68 86 76

104 Indira 25 86 82 84 96 93

105 Muriel 27 75 76 83 73 86

/\*在主函数中，输入整数 n(<20)，再输入 n个点的坐标（float型）。再利

用下面函数计算这些点中任意两点间最大、最小距离，通过 max,min返回

主函数。并在主函数输出这两个值。

//x--横坐标， y--纵坐标，n--点数

void dist2(float\*x,float\*y,intn,float\*max,float\*min);\*/

#include <stdio.h>

#include <math.h>

void dist2(float \*x,float \*y,int n,float \*max,float \*min)

{

int i,j;

float dis;

\*max=sqrt((x[0]-x[1])\*(x[0]-x[1])+(y[0]-y[1])\*(y[0]-y[1]));

\*min=sqrt((x[0]-x[1])\*(x[0]-x[1])+(y[0]-y[1])\*(y[0]-y[1]));

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

{

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

{

dis=sqrt((x[i]-x[j])\*(x[i]-x[j])+(y[i]-y[j])\*(y[i]-y[j]));

if(dis>\*max) \*max=dis;

if(dis<\*min) \*min=dis;

}

}

}

int main()

{

float x[20],y[20],\*a,\*b,max,min;

int i,n;

a=x;

b=y;

scanf("%d",&n);

for(i=0;i<n;i++) scanf("%f %f",&x[i],&y[i]);

dist2(a,b,n,&max,&min);

printf("%.6f\n%.6f",max,min);

return 0;

}