.HE4215 分数矩阵和

>#include<stdio.h>

int main()

{

long n,i,t,c;

double sum;

printf("Please input n: ");

scanf("%ld",&n);

printf("Output:\n");

while(n!=0)

{

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

sum+=1.0/(n-c)\*(c+1);

sum=sum\*2+n;

printf("the total is: %.2lf\n",sum);

printf("Please input n: ");

scanf("%ld",&n);

}

return 0;

}

2.WB4204 按要求对二维数组进行排序

<p>写一个C程序，将一个5×5的整形矩阵中最大的元素放在中心，4个角分别放4个最小的元素（顺序为从左到右，从上到下顺序依次从小到大存放）。

#include <stdio.h>

int main() {

void T(int mat[][5], int n);

int mat[5][5];

int i, j;

printf("Please input 5\*5 array:\n");

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

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

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

T(mat, 5);

printf("Output:\nThe new array is:\n");

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

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

printf("%4d", mat[i][j]);

printf("\n");

}

return 0;

}

/\* 矩阵处理函数 \*/

void T(int mat[][5], int n) {

int i, j, temp;

int \*a, \*b, \*c, \*d, \*max;

/\* 设置4个最小元素以及最大元素指针的初始值 \*/

a = &mat[0][0];

b = &mat[0][0];

c = &mat[0][0];

d = &mat[0][0];

max = &mat[0][0];

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

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

if (mat[i][j] < \*a) {

d = c; c = b; b = a;

a = &mat[i][j];

} else if (mat[i][j] < \*b) {

d = c; c = b;

b = &mat[i][j];

} else if (mat[i][j] < \*c) {

d = c;

c = &mat[i][j];

} else if (mat[i][j] < \*d) {

d = &mat[i][j];

}

if (mat[i][j] > \*max)

max = &mat[i][j];

}

}

/\* 依次替换4个最小元素以及最大的元素 \*/

temp = mat[0][0]; mat[0][0] = \*a; \*a = temp;

temp = mat[0][4]; mat[0][4] = \*b; \*b = temp;

temp = mat[4][0]; mat[4][0] = \*c; \*c = temp;

temp = mat[4][4]; mat[4][4] = \*d; \*d = temp;

temp = mat[2][2]; mat[2][2] = \*max; \*max = temp;

}

3.LY4301 [1226]输出杨辉三角形的前n行

<h3>题目描述</h3><p><p>输出杨辉三角形的前n行。>输入格式</h3><p><p>第一行有一个正整数n，保证n不超过20。输出</h3><p><p>共有n行，从第1行至第n行，每行输出与行数相等个数的整数，整数后输出一个空格。<br />请注意行尾输出换行。

#include <stdio.h>

int main() {

int dat[20][20];

int n, i, j;

printf("Input n: ");

scanf("%d", &n);

printf("Output:\n");

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

dat[i][0] = dat[i][i] = 1;

for (j = 0;j + 1 < i;j++)

dat[i][j + 1] = dat[i - 1][j] + dat[i - 1][j + 1];

}

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

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

printf("%d ", dat[i][j]);

puts("");

}

return 0;

}

4.T4205 ;[2110]矩阵对角求和

题目描述求一个n×n矩阵对角线元素之和，其中

#include <stdio.h> //2110 把小于10000那个删除

int main()

{

int i,j,n;

int a[100][100];

int Zsum=0,Fsum=0;

printf("Please input n(1<=n<100): ");

scanf("%d",&n);

if(n<1 || n>=100)

{

printf("Output:\nInput error!\n");

return 0;

}

printf("Please input the number of a[n][n]:\n");

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

{

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

{

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

}

}

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

{

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

{

if(i==j)

{

Zsum = Zsum + a[i][j];

}

if(j==(n-1-i))

{

Fsum = Fsum + a[i][j];

}

}

}

printf("Output:\n%d\n%d\n",Zsum,Fsum);

}

5.D4203 从键盘上输入矩阵的阶数n(n<=14), 矩阵中元素的值等于其位置的行数和列数之和的n倍

#include <stdio.h>

int main(void)

{

int i, j, n, sum1 = 0, sum2 = 0;

int array[15][15];

printf("Input:\nPlease Enter n:\n");

scanf("%d", &n);

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

{

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

{

array[i][j] = (i + j) \* n;

}

}

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

{

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

{

sum1 += array[i][j];

}

}

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

{

sum2 += array[i][i];

}

printf("Output:\n");

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

{

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

{

printf("%4d", array[i][j]);

}

printf("\n");

}

printf("sum1=%d\nsum2=%d\n", sum1, sum2);

return 0;

6.WB4105 二维数组转置

<p>输入的一个3×3的二维整型数组，将其转置，即行列互换。输出转置后的二维数组。

#include <stdio.h>

#include <math.h>

int main() {

void T(int array[][3]);

int i, j;

int dat[3][3];

printf("Please input 3\*3 array:\n");

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

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

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

T(dat);

printf("Output:\nThe transposed array is:\n");

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

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

printf("%d ", dat[i][j]);

puts("");

}

return 0;

}

void T(int array[][3]) {

int i, j, temp;

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

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

temp = array[i][j];

array[i][j] = array[j][i];

array[j][i] = temp;

}

}

}

7.HE4204 杨辉三角

：n行杨辉三角，每个数字按位宽为四&nbsp;输出，

#include<stdio.h>

#define N 10

int main()

{

int i,j,a[N][N],n;

printf("Please input n: ");

scanf("%d",&n);

printf("Output:\n");

if(n>10 || n<=0)

{

printf("Input error!\n");

return 0;

}

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

{

a[i][i]=1;

a[i][0]=1;

}

for(i=2;i<=n-1;i++)

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

a[i][j]=a[i-1][j]+a[i-1][j-1];

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

{

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

printf("%4d",a[i][j]);

printf("\n");

}

}

8.HH4203 最大的两个数

<p>输入一个四行五列的矩阵，找出每列最大的两个数。

#include<stdio.h>

#define P(z) for(i=0;i<5;++i)printf("%d ",z[i]);printf("\n");

int main(){

int n,i,j,t,a[5],b[5];

scanf("%d",&n);

while(n--){

scanf("%d%d%d%d%d%d%d%d%d%d",a,a+1,a+2,a+3,a+4,b,b+1,b+2,b+3,b+4);

for(i=0;j=i%5,i<10;++i){

scanf("%d",&t);

if(t>a[j]){

if(a[j]<b[j]){

a[j]=b[j];

b[j]=t;

} else

b[j]=t;

} else if(t>b[j])

b[j]=t;

}

printf("Output:\n");

P(a);P(b);

}

}

9.LY4201 [1223]计算矩阵对角线元素之和

#include <stdio.h>

int main() {

int dat[10][10];

int n, i, j, sum;

printf("Input n: ");

scanf("%d", &n);

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

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

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

}

}

sum = 0;

printf("Output:\n");

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

sum += dat[i][i];

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

return 0;

}

10.HH4206 对称矩阵

<p>输入一个N维矩阵，判断是否对称。

#include<stdio.h>

void run(int n)

{

int a[105][105],i,j;

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

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

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

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

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

if(a[i][j]!=a[j][i])

{

printf("Output:\nNo!\n");

return;

}

printf("Output:\nYes!\n");

}

int main()

{

int n;

scanf("%d",&n);

run(n);

}

11.WB4203 输出螺旋数

<p>螺旋数：</p><p>编写C程序，要求：输入一个正整数N（1-15），则生成一个N\*N二维数组，由数组左上角从1开始用自然数列顺时针螺旋填充该数组直至数组中心位置，将该N\*N数组中元素全部填充完即程序结束。

#include <stdio.h>

void main()

{

int i, j, n=1, a[16][16],N;

printf("Please input N [1-15]:\n");

scanf("%d",&N);

printf("Output:\n");

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

{

for (j=i; j<N-i; j++)

a[i][j]=n++;

for (j=i+1; j<N-i; j++)

a[j][N-i-1]=n++;

for (j=N-i-2; j>i; j--)

a[N-i-1][j]=n++;

for (j=N-i-1; j>i; j--)

a[j][i]=n++;

}

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

{

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

printf("%5d", a[i][j]);

printf("\n");

}

}

13.HE4104 构建矩阵

现请你构建一个N\*N的矩阵，第i行j列的元素为i与j的乘积。

#include<stdio.h>

int main()

{

int c,n,i,j,k,a[10][10];

printf("Please input test times: ");

scanf("%d",&c);

printf("Output:\n");

while(c--)

{

printf("input rows or cols: ");

scanf("%d",&n);

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

{

k=0;

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

{

a[i][j]=i\*j;

printf(k++?" %d":"%d",a[i][j]);

}

printf("\n");

}

}

return 0;

}

14.WB4211 蛇形N矩阵

<p>蛇形矩阵是由1开始的自然数依次排列成的一个矩阵上三角形。输入：本题有多组数据，每组数据由一个正整数N组成。

#include<stdio.h>

#define M 100

int main()

{

int a[M][M];

int n,i,j;

printf("Please input n:\n");

scanf("%d",&n);

if(n <= 0)

return 0;

printf("Output:\n");

a[0][0] = 1;//因为每次第一个数必是1

printf("%d ",a[0][0]);

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

{

a[i+1][0] = a[i][0] + i + 1;//提前算出下一行的第一个数据

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

{

a[i][j+1] = a[i][j] + j + i + 2;//算出每一行的每个数据

printf("%d ",a[i][j+1]);//打印每一行的数据

}

printf("\n%d ",a[i+1][0]);//打印下一行的第一个数据

}

printf("\n");

return 0;

}

15.LY4302 [1227]找出一个二维数组中的所有鞍点

<h3>题目描述找出一个二维数组中的所有鞍点，即该位置上的元素在该行中最大但是在该列中最小。需要注意有可能鞍点不存在，此时需要输出“NO”。

#include <stdio.h>

int main() {

int dat[50][50];

int n, m, i, j, k, saddle, saddleCnt = 0;

printf("Input n,m: ");

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

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

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

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

}

}

printf("Output:\n");

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

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

saddle = 1;

for (k = 0;k < m && saddle == 1;k++)

if (dat[i][k] > dat[i][j])

saddle = 0;

for (k = 0;k < n && saddle == 1;k++)

if (dat[k][j] < dat[i][j])

saddle = 0;

if (saddle == 1) {

saddleCnt++;

printf("%d %d %d\n", dat[i][j], i, j);

}

}

}

if (saddleCnt == 0)

puts("NO");

return 0;

}

16.WB4107 输出不及格学生4门课成绩

<p>有一个班级，有3名学生，各学4门课，查找有一门或一门以上课程不及格（低于&nbsp;60分）的学生，并输出他们的所有成绩。

#include <stdio.h>

int main() {

int a[3][4];

int i, j;

printf("Please input 3 students' scores:[0-100]\n");

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

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

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

printf("Output:\nNot pass students are:\n");

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

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

if (a[i][j] < 60)

break;

if (j < 4) {

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

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

printf("\n");

}

}

return 0;

}

<p>蛇形矩阵是由1开始的自然数依次排列成的一个矩阵上三角形。<br />Input&nbsp;<br />本题有多组数据，每组数据由一个正整数N组成。

#include<stdio.h>

#define M 100

int main()

{

int a[M][M];

int n,i,j;

printf("Please input n:\n");

scanf("%d",&n);

if(n <= 0)

return 0;

printf("Output:\n");

a[0][0] = 1;//因为每次第一个数必是1

printf("%d ",a[0][0]);

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

{

a[i+1][0] = a[i][0] + i + 1;//提前算出下一行的第一个数据

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

{

a[i][j+1] = a[i][j] + j + i + 2;//算出每一行的每个数据

printf("%d ",a[i][j+1]);//打印每一行的数据

}

printf("\n%d ",a[i+1][0]);//打印下一行的第一个数据

}

printf("\n");

return 0;

}