## 【对比实验】ch07-02指针作参数

## 【思考】

假定已有定义int a[4][3];

1. 二维数组的名字代表的含义是什么？
2. 如果要定义一个指针，使p1=&a[0][0],则指针p1的类型是什么？
3. 如果要定义一个指针，使p2=a; 则指针p2的类型应该是什么？（不要求）

## 【提示】

可以将列指针理解为行指针的具体元素，行指针理解为列指针的地址：

\*行指针 得到 列指针

&列指针 得到 行指针

### 【掌握】参考程序1：二维数组作形参

#include<stdio.h>

void OutputData(int a[][3],int m,int n) /\*二维数组名a作形参，a是指针，是一个行指针，因此对应的实参要传递有一个行指针过来，二维数组的名字本身就是一个行指针\*/

{

int i,j;

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

{

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

printf("%8d",a[i][j]);

printf("\n");

}

}

int main()

{

int i, data[4][3]={

{11,134,87},

{457,31,90},

{39,98,90},

{91,99,89}};

OutputData(data,4,3); // OutputData(&data[0],4,3);

}

### 【掌握】参考程序2：列指针做形参（列指针指向a[0][0]）

#include<stdio.h>

void OutputData(int \*p,int m,int n) //p指向元素a[0][0],因此需传入a[0][0]的地址

{

int i,j;

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

{

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

printf("%8d",\*p++);

printf("\n");

}

}

int main()

{

int i, data[4][3]={

{11,134,87},

{457,31,90},

{39,98,90},

{91,99,89}};

OutputData(&data[0][0],4,3);//OutputData(data[0],4,3); 或者 OutputData ( \*data,4,3);

}

### 【掌握】参考程序3：列指针做形参（列指针指向a[0][0]）

#include <stdio.h>

#define N 10

#define M 6

int arrmul(int \*p,int m , int n )

{ int u, v,mul=1;

for( u=0; u<m; u++)

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

mul = mul \* (\*(p+u\*N+v));

return mul ;

}

int main(){

int m,n,a[M][N],i,k,j;

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

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

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

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

k=arrmul(\*a,m,n); //&a[0][0],a[0],\*a都表示a[0][0]的地址

printf("%d",k);

return 0;

}

### 【拓展-不要求】参考程序3：行指针做参数

//pLine指向二维数组的第0个元素，即第0行，即第0个一维数组

#include<stdio.h>

void OutputData(int (\*pLine)[3],int m,int n)//形参是一个行指针，指向由3个元素构成的int类型一维数组

{

int i,j;

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

{

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

printf("%8d",(\*pLine)[j]);//输出pLine所指行的第j列元素

printf("\n");

pLine++;//指向下一行，因为pLine是一个行指针

}

}

int main()

{

int data[4][3]={

{11,134,87},

{457,31,90},

{39,98,90},

{91,99,89}};

OutputData(data,4,3);//OutputData(&data[0],4,3);

}