

# 第五章 数的存储与组织

## 第64课 二维数组的存储方式

---

《信息学奥赛一本通·编程启蒙 C++版》

二维数组  $A[m][n]$  可以视为由  $m$  个行向量组成的向量，或者是由  $n$  个列向量组成的向量。

$$A_{mn} = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \dots & \dots & \dots & \dots \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{bmatrix}$$

由于计算机的内存是一维的，多维数组的元素应排成线性序列后存入存储器。数组一般不做插入和删除操作，即结构中元素个数和元素间的关系不变。所以采用顺序存储方法表示数组。

## (1) 行优先存储

将数组元素按行向量排列，第  $i+1$  个行向量紧接在第  $i$  个行向量后面，二维数组  $A[m][n]$  按行优先存储的线性序列为： $A[0][0]$ 、 $A[0][1]\cdots A[0][n]$ 、 $A[1][1]$   
 $A[1][1]\cdots A[1][n]\cdots A[m][n]$ 、 $A[m][1]\cdots A[m][n]$

在 PASCAL、C 语言和 C++ 语言中数组按行优先顺序存储。

## (2) 列优先存储

将数组元素按列向量排列，第  $i+1$  个列向量紧接在第  $i$  个列向量后面，二维数组  $A[m][n]$  按列优先存储的线性序列为： $A[0][0]$ 、 $A[1][0]\cdots A[m][0]$ 、 $A[0][1]$   
 $A[1][1]\cdots A[m][1]\cdots A[m][1]$ 、 $A[0][n]\cdots A[m][n]$

**FORTRAN 语言中**，数组按列优先顺序存储。

## 【例 63.1】 二维数组转置

### 【题目描述】

输入一个  $n$  行  $m$  列的数组，输出他的转置，具体来说输出的第  $i$  行第  $j$  个数字，应是输入的第  $j$  行第  $i$  个数字。

$1 \leq n \leq 20000$

$1 \leq m \leq 20000$

$1 \leq n * m \leq 20000$

$1 \leq a[i][j] \leq 1000$

特别注意，你并不能声明一个二维数组，两维的大小都是 20000。（数组不能太大，二维数组的大小是两维的乘积）

二维数组本质上和一维数组等价，所以你只需要一个大小为 20000 一维数组

### 【输入格式】

第一行两个整数  $n$ ， $m$  表示数组的行数和列数 接下来  $n$  行，每行  $m$  个整数表示数组内容。

### 【输出格式】

第一行先输出  $m$ ， $n$ 。 接下来输出转置的结果，共  $m$  行  $n$  列。 其中第  $i$  行第  $j$  个数字，应是输入的第  $j$  行第  $i$  个数字。

### 【样例输入】

3 2

1 2

3 4

5 6

### 【样例输出】

2 3

1 3 5

2 4 6

## 【代码实现 1】

```
1. #include<bits/stdc++.h>
2. using namespace std;
3. int a[20005],n,m;
4. int main(){
5.     cin>>n>>m;
6.     for(int i=0;i<n*m;i++) cin>>a[i];
7.
8.     cout<<m<<' '<<n<<endl;
9.     for(int j=0;j<m;j++){
10.         for(int i=0;i<n;i++) cout<<a[i*m+j]<<' ';
11.         cout<<endl;
12.     }
13.     return 0;
14. }
```

## 【代码实现 2】

```
1. #include<bits/stdc++.h>
2. using namespace std;
3. int x,n,m;
4. vector<int> a[20001];
5. int main(){
6.     cin>>n>>m;
7.     for(int i=0;i<n;i++)
8.         for(int j=0;j<m;j++){
9.             cin>>x;
10.            a[i].push_back(x);
11.        }
12.    cout<<m<<' ' <<n<<endl;;
13.    for(int j=0;j<m;j++){
14.        for(int i=0;i<n;i++) cout<<a[i][j]<<' ';
15.        cout<<endl;
16.    }
17.    return 0;
18.}
```

## 【例 64.2】 生日相同

### 【题目描述】

在一个有 180 人的大班级中,存在两个人生日相同的概率非常大,现给出每个学生的名字,出生月日。

试找出所有生日相同的学生。

### 【输入格式】

第一行为整数  $n$ ,表示有  $n$  个学生,  $n \leq 180$ 。

此后每行包含一个字符串和两个整数,分别表示学生的名字(名字第一个字母大写,其余小写,不含空格,且长度小于 20)和出生月( $1 \leq m \leq 12$ )日( $1 \leq d \leq 31$ )。

名字、月、日之间用一个空格分隔。

### 【输出格式】

每组生日相同的学生,输出一行,其中前两个数字表示月和日,后面跟着所有在当天出生的学生的名字,数字、名字之间都用一个空格分隔。

对所有的输出,要求按日期从前到后的顺序输出。

对生日相同的名字,按名字从短到长按序输出,长度相同的按字典序输出。

如没有生日相同的学生,输出 "None"。

### 【样例输入】

```
6
Avril 3 2
Candy 4 5
Tim 3 2
Sufia 4 5
Lagrange 4 5
Bill 3 2
```

### 【样例输出】

```
3 2 Tim Bill Avril
4 5 Candy Sufia Lagr
```

## 【代码实现】

```
1. #include<bits/stdc++.h>
2. using namespace std;
3. int n,m,d;
4. string name;
5. bool flag;
6. vector<string> a[13][32];
7. bool cmp(string x,string y){
8.     return x.size()<y.size()||x.size()==y.size()&&x<y;
9. }
10.int main(){
11.     flag=false;
12.     cin>>n;
13.     for(int i=1;i<=n;i++){
14.         cin>>name>>m>>d;
15.         a[m][d].push_back(name);
16.     }
17.     for(int i=1;i<=12;i++){
18.         for(int j=1;j<=31;j++){
19.             if(a[i][j].size()>=2){
20.                 flag=true;
21.                 cout<<i<<" "<<j<<" ";
22.                 sort(a[i][j].begin(),a[i][j].end(),cmp);
23.                 for(int k=0,len=a[i][j].size();k<len;k++){
24.                     cout<<a[i][j][k]<<" ";
25.                 }
26.                 cout<<endl;
27.             }
28.         }
29.     }
30.     if(!flag) cout<<"None"<<endl;
31.     return 0;
32.}
```

## 练 63.1 图像旋转

### 【题目描述】

输入一个  $n$  行  $m$  列的黑白图像，将它顺时针旋转  $90$  度后输出。

### 【输入格式】

第一行包含两个整数  $n$  和  $m$ ，表示图像包含像素点的行数和列数。 $1 \leq n \leq 100$ ， $1 \leq m \leq 100$ 。

接下来  $n$  行，每行  $m$  个整数，表示图像的每个像素点灰度。相邻两个整数之间用单个空格隔开，每个元素均在  $0 \sim 255$  之间。

### 【输出格式】

$m$  行，每行  $n$  个整数，为顺时针旋转  $90$  度后的图像。相邻两个整数之间用单个空格隔开

### 【样例输入】

```
3 3
1 2 3
4 5 6
7 8 9
```

### 【样例输出】

```
7 4 1
8 5 2
9 6 3
```

## 【代码实现】

```
1. using namespace std;
2. int a[100][100];
3. int main(){
4.     int n,m;
5.     cin>>n>>m;
6.     for(int i=0;i<n;i++){
7.         for(int j=0;j<m;j++) cin>>a[i][j];
8.     }
9.
10.    for(int j=0;j<m;j++){
11.        for(int i=n-1;i>=0;i--) cout<<a[i][j]<<" ";
12.        cout<<endl;
13.    }
14.    return 0;
15.}
```

## 练 64.3 打印沙漏

### 【题目描述】

题要求你写个程序把给定的符号打印成沙漏的形状。例如给定 17 个“\*”，要求按下列格式打印：

```
*****
 ***
  *
 ***
*****
```

所谓“沙漏形状”，是指每行输出奇数个符号；各行符号中心对齐；相邻两行符号数差 2；符号数先从大到小顺序递减到 1，再从小到大顺序递增；首尾符号数相等。

给定任意 N 个符号，不一定能正好组成一个沙漏。要求打印出的沙漏能用掉尽可能多的符号。

### 【输入格式】

输入在一行给出 1 个正整数  $N(N \leq 1000)$  和一个符号，中间以空格分隔。

### 【输出格式】

首先打印出由给定符号组成的最大的沙漏形状，最后在一行中输出剩下没用掉的符号数。

### 【样例输入】

```
19 *
```

### 【样例输出】

```
*****
 ***
  *
 ***
*****
2
```

## 【代码实现】

```
1. #include<bits/stdc++.h>
2. using namespace std;
3. long long n,a[1000],i;
4. char c;
5. void print(int n){
6.     for(int i=n*2-1,sp=0;i>=1;i-=2,sp++){
7.         for(int j=1;j<=sp;j++)cout<<' ';
8.         for(int j=1;j<=i;j++)cout<<c;
9.         cout<<endl;
10.    }
11.    for(int i=2,sp=n-2;i<=n;i++,sp--){
12.        for(int j=1;j<=sp;j++)cout<<' ';
13.        for(int j=1;j<=i*2-1;j++)cout<<c;
14.        cout<<endl;
15.    }
16.}
17.int main(){
18.    cin>>n>>c;
19.    a[1]=1;
20.    for(i=2;;i++){
21.        a[i]=a[i-1]+2*(2*i-1);
22.        if(a[i]>n)break;
23.    }
24.    i--;
25.    print(i);
26.    cout<<n-a[i];
27.    return 0;
28.}
```

谢谢！

—