第五章 数的存储与组织

第62课 认识二维数组

《信息学奥赛一本通·编程启蒙 C++版》

【例 62.1】 矩阵加法

【题目描述】

输入两个 n 行 m 列的矩阵 A 和 B，输出它们的和 A+B。

【输入格式】

第一行包含两个整数 n 和 m，表示矩阵的行数和列数。1≤n≤100，1≤m≤100。

接下来 n 行，每行 m 个整数，表示矩阵 A 的元素。

接下来 n 行，每行 m 个整数，表示矩阵 B 的元素。

相邻两个整数之间用单个空格隔开，每个元素均在 1 ~ 1000 之间。

【输出格式】

n 行，每行 m 个整数，表示矩阵加法的结果。相邻两个整数之间用单个空格隔开。

【样例输入】

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

【样例输出】

```
2 4 6
5 7 9
8 10 12
```

【代码实现】

```
#include<bits/stdc++.h>
using namespace std;
int n,m,x;
int a[105][105];
int main(){
    cin>>n>>m;
    for(int i=1;i<=n;i++)
       for(int j=1;j<=m;j++)
          cin>>a[i][j];
    for(int i=1;i<=n;i++)
       for(int j=1;j<=m;j++){
         cin>>x;
          a[i][j]+=x;
       }
    for(int i=1;i<=n;i++){
       for(int j=1;j<=m;j++){
          cout<<a[i][j]<<' ';
       }
       cout<<endl;
    }
    return 0;
}
```

【例 62.2】 相邻数之和

【题目描述】

请你编程求出二维数组中某点的相邻数之和。相邻数是指与该点邻接的 8 个元素，若该点在边角位置，则邻接元素相应减少。

下图以 4 行 5 列二维数组 a 为例：a[2][3] 元素的值为 7，其邻接元素为 8,9,10,5,8,6,8,0 和为 54 。再比如：a[1][0] 元素的值为 6，则其邻接元素为 1,2,7,3,4 和为 17。

1 2 3 4 5

6 7 8 9 10

3 4 5 7 8

2 5 6 8 0。

【输入格式】

第一行输入 4 个整数：h,l,c,r 分别代表二维数组的行列值和指定点的行列下标。

接下来输入 h 行 l 列的 int 型二维数组 a。其中 2≤h,l≤10；而 0≤c,r≤9；注意下标值从 0 开始。

【输出格式】

a[c][r] 的邻接元素之和。

【样例输入】

4 5 2 3

1 2 3 4 5

6 7 8 9 10

3 4 5 7 8

2 5 6 8 0

【样例输出】

54

【代码实现】

```
#include<bits/stdc++.h>
using namespace std;
long long a[15][15],h,l,c,r,sum,
dx[8]={-1,-1,-1,0,1,1,1,0},
dy[8]={-1,0,1,1,1,0,-1,-1};
bool check(int x,int y){
    return x>=0&&x<h&&y>=0&&y<l;
}
int main(){
   cin>>h>>l>>c>>r;
   for(int i=0;i<h;i++)
      for(int j=0;j<l;j++)
         cin>>a[i][j];
    for(int i=0;i<8;i++){
        int nx=c+dx[i],ny=r+dy[i];
        if(check(nx,ny)) sum+=a[nx][ny];
    }
    cout<<sum;
    return 0;
}
```

【例 62.3】 地雷数计算

【题目描述】

扫雷游戏是一款十分经典的单机小游戏。它的精髓在于，通过已翻开格子所提示的周围格地雷数，来判断未翻开格子里是否是地雷。

现在给出 n 行 m 列的雷区中的地雷分布，要求计算出每个非地雷格的周围格地雷数。

注:每个格子周围格有八个:上、下、左、右、左上、右上、左下、右下。

【输入格式】

第一行包含两个整数 n 和 m，分别表示雷区的行数和列数。1≤n≤100,1≤m≤100.

接下来 n 行，每行 m 个字符，‘*’表示相应格子中是地雷，‘?’表示相应格子中无地雷。字符之间无任何分隔符。

【输出格式】

n 行，每行 m 个字符，描述整个雷区。若相应格中是地雷，则用'*’表示，否则用相应的周围格地雷数表示。字符之间无任何分隔符。

【样例输入】

```
3 3
*??
???
?*?
```

【样例输出】

```
*10
221
1*1
```

【代码实现】

```
#include<bits/stdc++.h>
using namespace std;
int n,m,nx,ny,sum;
int dx[8]={-1,-1,0,1,1,1,0,-1};
int dy[8]={0,1,1,1,0,-1,-1,-1};
char a[105][105],c[105][105];
int main(){
   cin>>n>>m;
   for(int i=0;i<n;i++){
     for(int j=0;j<m;j++){
        cin>>c[i][j];
     }
 }
 for(int i=0;i<n;i++){
   for(int j=0;j<m;j++){
     if(c[i][j]=='*') cout<<"*";
     else{
       sum=0;
        for(int k=0;k<8;k++){
           nx=i+dx[k];
           ny=j+dy[k];
           if(nx>=0&&nx<n&&ny>=0&&ny<m&&c[nx][ny]=='*')
              sum++;
        }
        cout<<sum;
      }
   }
   cout<<endl;
 }
 return 0;
}
```

练 62.1 输出矩形的下三角部分

【题目描述】

输出矩形的下三角部分。

矩形的下三角是指第 1 行的前 1 个数，第 2 行的前两个数，第 n 行的前 n 个数。

【输入格式】

第 1 行是矩阵的大小 n。

第 2~ n+1 行是矩阵的内容 1≤n≤100，1 ≤ 所有数字≤1000。

【输出格式】

输出 n 行。

第 1 行是矩阵的第 1 个数。

第 2 行是矩阵的前 2 个数。

... ...

第 n 行是矩阵的前 n 个数

【样例输入】

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

【样例输出】

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

【代码实现】

```
#include<bits/stdc++.h>
using namespace std;
int n,a[101][101];
int main(){
    cin>>n;
    for(int i=1;i<=n;i++)
      for(int j=1;j<=n;j++)
         cin>>a[i][j];
    for(int i=1;i<=n;i++){
      for(int j=1;j<=i;j++)
          cout<<a[i][j]<<' ';
      cout<<endl;
    }
    return 0;
}
```

练 62.2 矩阵乘法

【题目描述】

计算两个矩阵的乘法。n×m 阶的矩阵 A 乘以 m×k 阶的矩阵 B 得到的矩阵 C 是 n×k 阶的，且 C[i][j]=A[i][0]×B[0][j]+A[i][1]×B[1][j]+……+A[i][m-1]×B[m-1][j](C[i][j] 表示 C 矩阵中第 i 行第 j 列元素)。

【输入格式】

第一行为 n，m，k，表示 A 矩阵是 n 行 m 列，B 矩阵是 m 行 k 列，n,m,k 均小于 100;

然后先后输入 A 和 B 两个矩阵，A 矩阵 n 行 m 列，B 矩阵 m 行 k 列，矩阵中每个元素的绝对值不会大于 1000。

【输出格式】

输出矩阵 C，一共 n 行，每行 k 个整数，整数之间以一个空格分开

【样例输入】

```
3 2 3
1 1
1 1
1 1
1 1 1
1 1 1
```

【样例输出】

```
2 2 2
2 2 2
2 2 2
```

【代码实现】

```
#include<bits/stdc++.h>
using namespace std;
int n,m,k,a[105][105],b[105][105],sum;
int main(){
    cin>>n>>m>>k;
    for(int i=1;i<=n;i++)
        for(int j=1;j<=m;j++) cin>>a[i][j];
    for(int i=1;i<=m;i++)
        for(int j=1;j<=k;j++) cin>>b[i][j];
    for(int i=1;i<=n;i++){
        for(int j=1;j<=k;j++){
            sum=0;
            for(int l=1;l<=m;l++) sum+=a[i][l]*b[l][j];
            cout<<sum<<' ';
        }
        cout<<endl;
    }
    return 0;
}
```

练 62.3 学习效率

【题目描述】

某班有 n2 名同学，他们的座位有 n 排，每排 n 个人，每个人的勤奋值为 ai，j 现在要统计每个同学的学习效率，学习效率定义为每个人的勤奋值加上周围最多 8 个同学的勤奋值，也就是以每个人为中心的 3×3 方阵内勤奋值之和。现在请你统计每个人的学习效率。

【输入格式】

第一行一个整数 n (1≤n≤10)，表示班级的规模。

接下来 n 行，每行 n 个整数 a i,j (1≤a i,j ≤100)，表示每个人的勤奋值。

【输出格式】

输出 n 行，每行 n 个整数，每个整数表示每个人的学习效率

【样例输入】

2

1 2

3 4

【样例输出】

10 10

10 10

【代码实现】

```
#include<bits/stdc++.h>
using namespace std;
int a[15][15],n;
int main(){
    cin>>n;
    for(int i=1;i<=n;i++)
        for(int j=1;j<=n;j++)
            cin>>a[i][j];
   for(int i=1;i<=n;i++){
      for(int j=1;j<=n;j++){
       cout<<a[i][j]+a[i-1][j-1]+a[i-1][j]+a[i-1][j+1]+a[i][j-1]
            +a[i][j+1]+a[i+1][j-1]+a[i+1][j]+a[i+1][j+1]<<' ';
        }
        cout<<endl;
    }
    return 0;
}
```

谢谢！