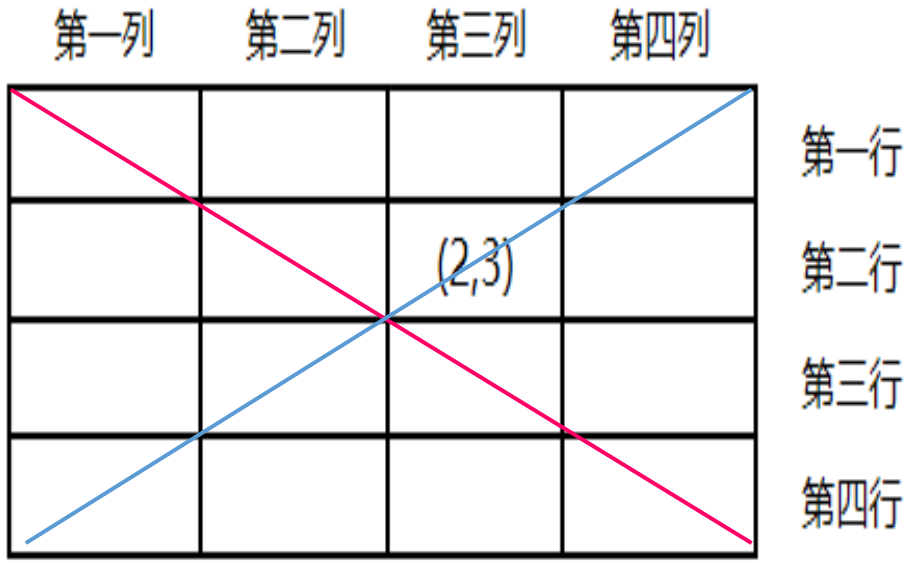
矩阵中的位置（下标从1，1开始）

复习：斜线与[i][j]关系 （下标从1，1开始）



结论：

1、同一行时：i相等　　同一列时：j相等

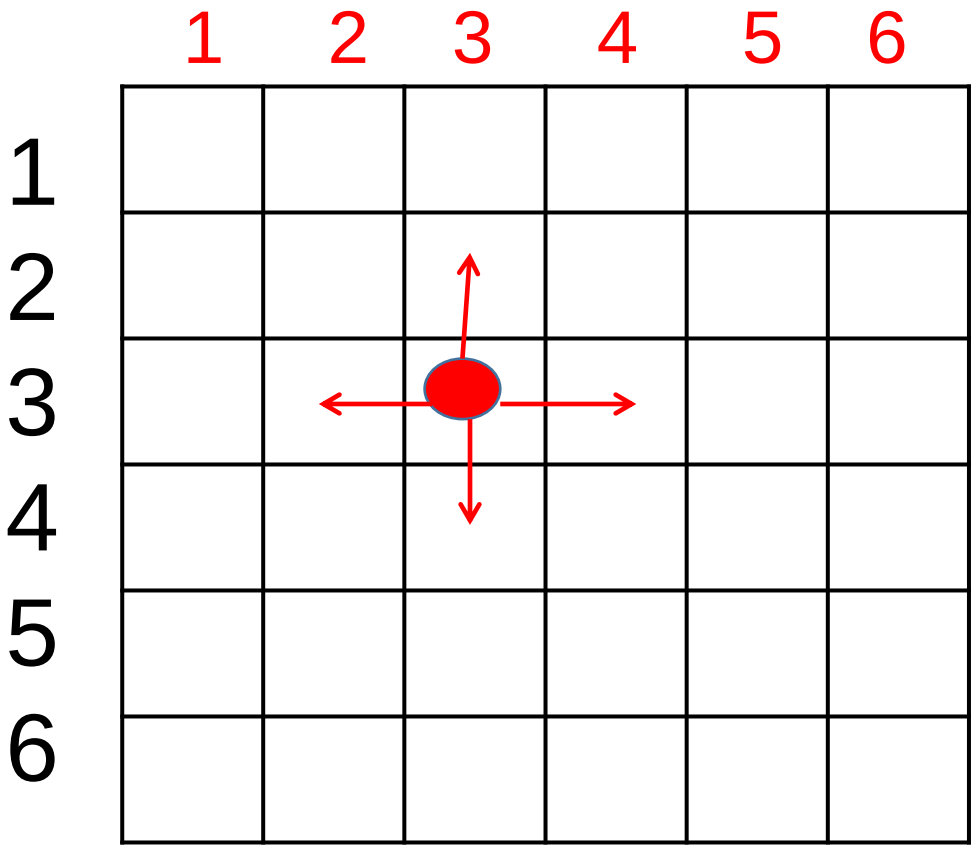
2、N*N矩阵时，同一对角线，i-j=0　　或 i+j=n+1　（**i=n+1-j**）

3、N*M矩阵中，在同一斜线的位置：

i-j相等时在 （左上到右下） 每一斜线值等于多少？

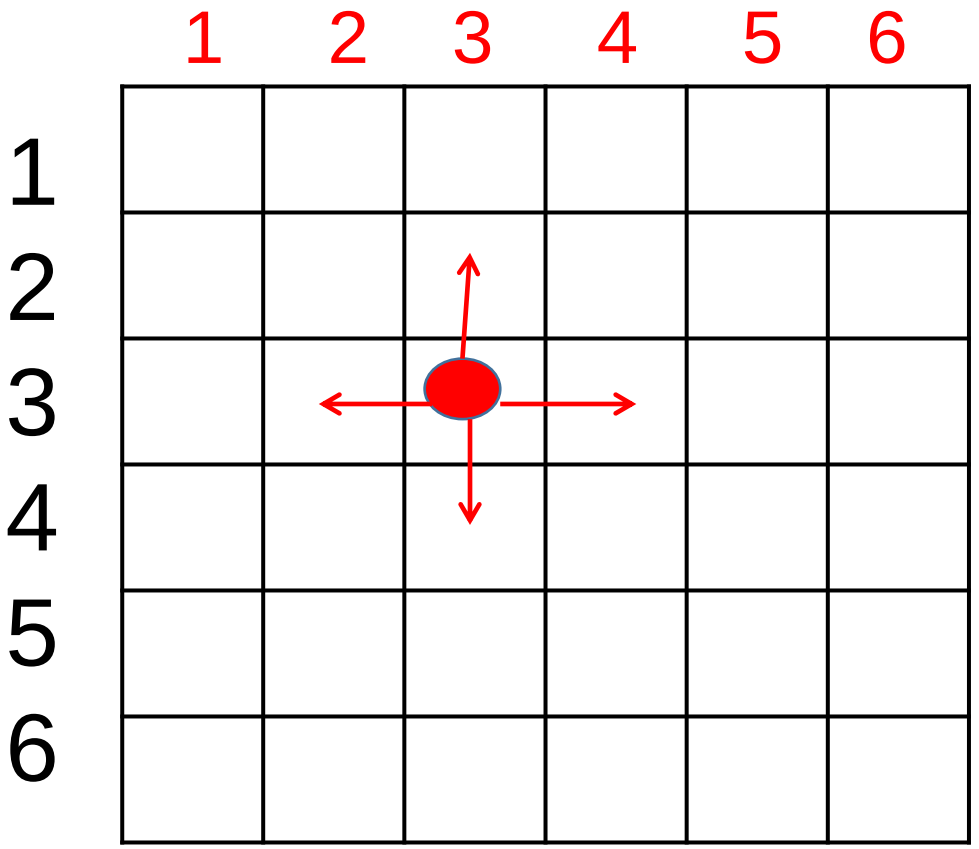
i+j 相等时在（左下到右上）

元素与 4 个位置关系：



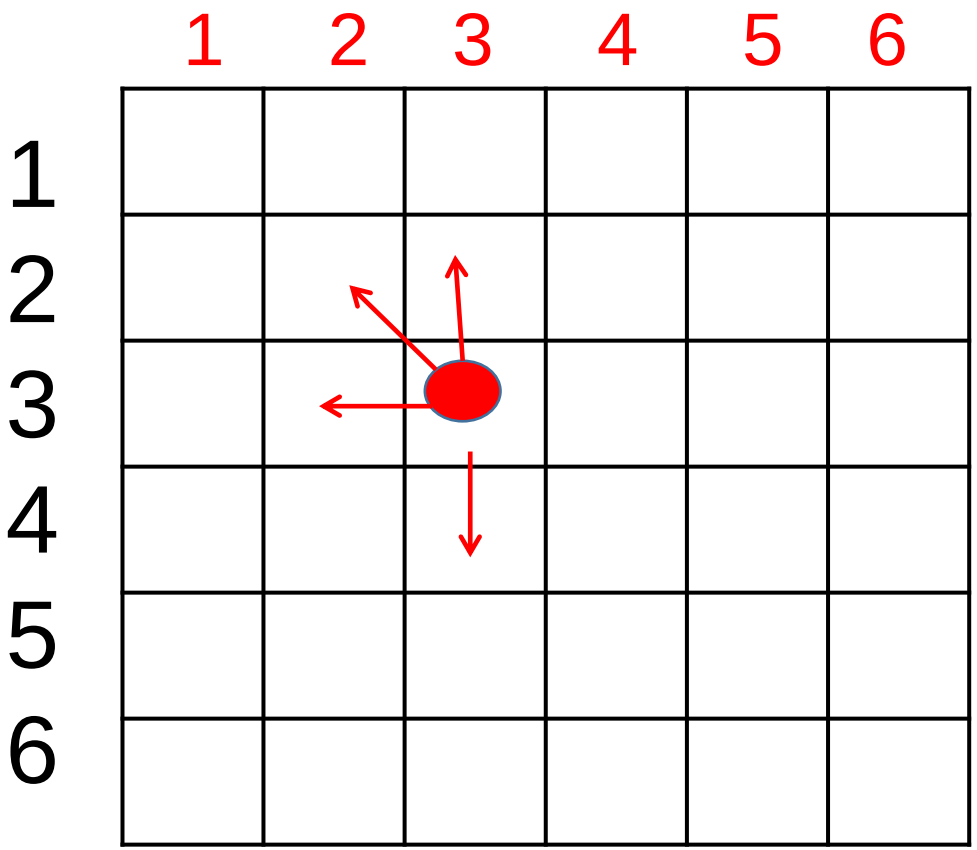
一维表示法：

```
Dx[4]={-1,1,0,0}
Dy[4]={0,0,-1,1}    对应：上下左右
                          0 1 2 3
For(int  i=0;i<4;i++){ //枚举4个方向
    x=x+dx[i];y=y+dy[i];

  }
```



二维表示法：

```
dir[ ][2]={ {0,-1},{0,1},{-1,0},{1,0} };

For(int i=0;i<4;i++){ //枚举4个方向
    x=x+ dir[i][0];y=y+ dir[i][1];
}
```



A[x][y]8方向二维表示法：

dir[][2]={ {1,0},{0,1},{0,-1},{-1,0},{1,1},{-1,-1},{-1,1},{1,-1} } ;

```
For(int i=0;i<8;i++){ //枚举8个方向
    x=x+dir[i][0];y=y +dir[i][1];
}
```

《扫雷地图》

实验舱蛟龙三班

二维数组（3）

zlj

2022.8

一：矩阵旋转

1、按i、j 不同顺序输出

2、借助i，j位置变化，构造新的矩阵

思考：如何将n*n矩阵顺时针旋转90

如：

3

1 2 3

4 5 6

7 8 9

7 4 1

8 5 2

9 6 3

1 2 3

4 5 6

7 8 9

直接输出：

矩阵旋转也可以在原数组的基础上，改变行列输出顺序达到目的。
如：顺时针旋转90度， 也可以用先输出列再行完成：

```
void T90(){  //  N*M矩阵
  for(int i=1;i<＝m;i++){
      for(int j=n;j>=1;j--){
          cout<<  a[ j ][ i] <<" ";
      }
      cout<<endl;
   }
}
```

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

输出：

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

n*n图像顺时针旋转90 （构造矩阵）

如：

3

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

试找一找有没有规律？
尝试写出规律：
1、列变成了？
2、行变成了？
由 矩阵1 生成 矩阵2

A[1][1]→b[1][3]

A[1][2]→b[2][3]

A[1][3]→b[3][3]

n*n图像顺时针旋转90

如：

3

1 2 3
4 5 6
7 8 9

7 4 1
8 5 2
9 6 3

b[j][n+1-i]=a[i][j]

行变成列， 列变行

第1行变最后一列

n*n顺时旋转90

```
3  int a[50][50],b[50][50],n;
4  int main() {
5      cin>>n;
6      for(int i=1; i<=n; i++)
7          for(int j=1; j<=n; j++) {
8              cin>>a[i][j];//读入a矩阵
9              b[j][n+1-i]=a[i][j]; //构造 b矩阵
10         }
11
12     for(int i=1; i<=n; i++) {
13         for(int j=1; j<=n; j++)
14             cout<<b[i][j]<<" ";//输出 b矩阵
15         cout<<endl;
16     }
17     return 0;
```

2：n*m的图像顺时针旋转90

把N*N改在N*M呢？如何修改程序？

3 4

1 2 3 4

5 6 7 8

9 10 11 12

样例输出

9 5 1

10 6 2

11 7 3

12 8 4

规律：

1、列变成了？

2、行变成了？

由 矩阵1 生成 矩阵2

3、两个矩阵的行列数有交换？

b[j][n+1-i]=a[i][j]

n*m的图像顺时针旋转90 （注意输出的矩阵行列变化）

```
3  int a[101][100],b[100][101],n,m;
4  int main() {
5      cin>>n>>m;
6      for(int i=1; i<=n; i++)
7          for(int j=1; j<=m; j++) {
8              cin>>a[i][j];//读入a矩阵
9              (1) =a[i][j]; //构造 b矩阵
10         }
11
12     for(int i=1; i<= (2) ; i++) {
13         for(int j=1; j<= (3) ; j++)
14             cout<<b[i][j]<<" ";//输出 b矩阵
15         cout<<endl;
16     }
17     return 0;
```

矩阵旋转规则比较（关注行列的变化）

1、（n*n）顺时针旋转90度

```
Void Ts90(){
  for(int i=1;i<=n;i++)
     for(int j=1;j<=n;j++){
        b[j][n+1-i]=a[i][j];
       }
}
```

2、（n*m）顺时针旋转90度

```
Void Ts90(){
  for(int i=1;i<=n;i++)
    for(int j=1;j<=m;j++){
      b[j][n+1-i]=a[i][j];}
}
Void out(){
   for(int i=1;i<=m;i++){
      for(int j=1;j<=n;j++)
         cout<<b[i][j] <<“ “;
     cout<<endl; }
}
```

3、矩阵逆时针旋转90度

如：

3

1 2 3　　3 6 9

4 5 6　　2 5 8

7 8 9　　1 4 7

1 2 3 6　　1 2 3 6

4 5 6 7　　4 5 6 7

7 8 9 8　　7 8 9 8

（n*n）逆时针旋转90度

```
Void Tn90(){
 for(int i=1;i<=n;i++)
    for(int j=1;j<=n;j++){
       b[n+1-j][i]=a[i][j];
      }
}
```

4、矩阵顺时针旋转180度

（n*n）顺时针旋转180度

```
Void Ts180(){
  for(int i=1;i<=n;i++)
    for(int j=1;j<=n;j++){
      b[n+1-i][n+1-j]=a[i][j];}
}
```

如：

3

1 2 3

4 5 6

7 8 9

9 8 7

6 5 4

3 2 1

1 2 3

4 5 6

7 8 9

例2：变幻的矩阵

给两个N*N的矩阵，请判断是用哪种方式旋转变化而得到（只能旋转一次），1、顺时90度，2、逆时90度，3、顺时180度，4、没旋转或旋转360，5、以上都不是。

样例输入：

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

样例输出：

```
3
```

分析：

1、按照几种旋转方式，生成矩阵，与原矩阵比较

样例输入：

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

样例输出：

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

参考代码：

```
38  int main() {
39      cin>>n;
40      for(int i=1; i<=n; i++)//输入a,b数组
41          for(int j=1; j<=n; j++)
42              cin>>a[i][j];
43      for(int i=1; i<=n; i++)
44          for(int j=1; j<=n; j++)
45              cin>>b[i][j];
46      if( TR90() )cout<<1;//比较函数
47      else if( TL90() )cout<<2;
48      else if( T180() )cout<<3;
49      else if( T360() )cout<<4;
50      else cout<<5;
51      cout<<endl;
52      return 0;
53  }
```

参考函数：

```
4  bool TR90() {//顺时针旋转90
5      //a[i][j]  ---> b[j][n-i+1]
6      for(int i=1; i<=n; i++) {
7          for(int j=1; j<=n; j++) {
8              if(     ? ?          ) return false;
9          }
10     }
11     return true;
12 }
```

例3：熊孩子和向日葵

种植 N 株不同的向日葵，排序从最小到最大，并记录它们的高度连续N天。每天，她所有的花都比前一天长得更高。她在一张桌子上记录这些测量数据，每株一排，第一行记录最短的向日葵生长，最后一行记录最高的向日葵的生长。最左边的一栏是每个向日葵的第一次测量，最右边的一列是每一朵向日葵的最后一次测量。如果一朵向日葵在最初种植时比另一朵更小，那么它在每一次测量时都会更小。不幸的是，熊孩子们可能改变了她的尺寸，把她的桌子旋转了N个90度。你的工作是帮助芭芭拉确定她的原始数据。

【输入说明】

输入的第一行包含编号2≤*N*≤100)。接下来的N行每一行都包含N个正整数，每个正整数最多10^9。保证输入的数据代表芭芭拉数据的旋转版本。

【输出说明】

输出芭芭拉的原始数据，包括N行，每一行包含N个正整数。

输入样例

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

输出样例

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

分析：

1、每颗向日葵是有序的
2、每天的生长数据也是递增的
3、正确的数据一定也是递增的
4、根据什么来判断转了多少度？

参考代码：

```
29  int main() {
30      cin>>n;
31      //输入数据 并找出最小值（第一颗第一天数据位置）
32      for(int i=1; i<=n; i++)
33          for(int j=1; j<=n; j++) {
34              cin>>a[i][j];
35              b[i][j]=a[i][j];//360度或没旋转用
36              if(a[i][j]<minx) {
37                  minx=a[i][j];
38                  x=i,y=j; //记录位置
39              }
40          }
41      //判断位置并复原
42      if(x==n&&y==n) t180() ;
43      else if(x==1&&y==n)  lt90() ;
44      else if(x==n&&y==1)  rt90() ;
45      //输出旋转后的b数组
46      out();
47      return 0;
48  }
```

输入样例

3

4 3 1

6 5 2

9 7 3

输出样例

1 2 3

3 5 7

4 6 9

例5：疯狂的旋转

给你一个N*M的矩阵，请你顺时针旋转90度K次后，输出旋转后的矩阵。

如：

3 3 1	输出：
1 2 3	7 4 1
4 5 6	8 5 2
7 8 9	9 6 3

分析：

1、旋转1次、2次、3次、4次 如何旋转？

2、超过4次的，有什么特点？

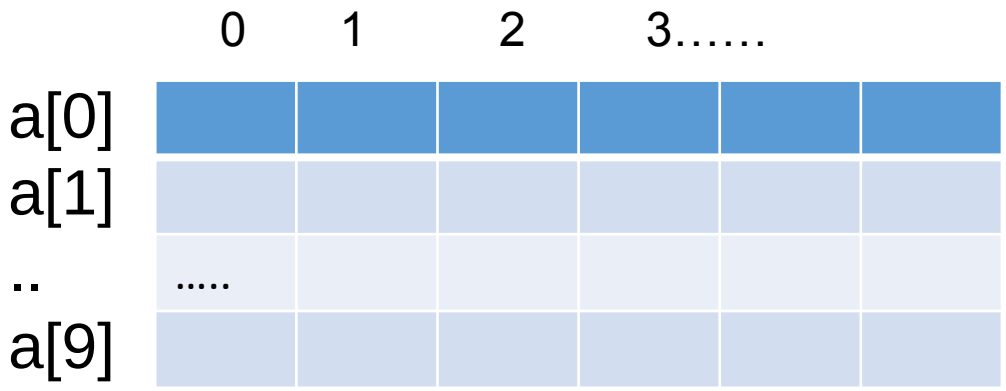
填空：

```
using namespace std;//矩阵疯狂旋转
int a[205][205],n,m,k,b[205][205];
void T90() {// 顺时90
    for(int i=1; i<=n; i++)
        for(int j=1; j<=m; j++) {
            b (1) =a[i][j];//旋转行列变化
        }
    for(int i=1; i<= (2) ; i++) { //输出b数组
        for(int j=1; j<= (3) ; j++)
            cout<<b[i][j]<<" ";
        cout<<endl;
    }
}
void T180() {// 顺时180
    for(int i=1; i<=n; i++)
        for(int j=1; j<=m; j++)
            (4) =a[i][j];  //旋转行列变化
    for(int i=1; i<=n; i++) {
        for(int j=1; j<=m; j++)
            cout<<b[i][j]<<" ";
        cout<<endl;
    }
}
void T270() {// 顺时270
    for(int i=1; i<=n; i++)
        for(int j=1; j<=m; j++)
            (5) ;//旋转行列变化
    for(int i=1; i<=m; i++) {
        for(int j=1; j<=n; j++)
            cout<<b[i][j]<<" ";
```

```
int main() {
    cin>>n>>m>>k;
    for(int i=1; i<=n; i++)
        for(int j=1; j<=m; j++)
            cin>>a[i][j];
    int k1= (6) ; //计算旋转圈数
    if(k1==1)
        T90();
    else if(k1==2)
        T180();
    else if(k1==3)
        T270();
    else
        for(int i=1; i<=n; i++) {
            for(int j=1; j<=m; j++)
                cout<<a[i][j]<<" ";
            cout<<endl;
        }
```

三：字符数组（二维）

String a[10];　　char　b[10][10]



6、最好的草

描述：奶牛Bessie计划好好享受柔软的春季新草。新草分布在R行C列的牧场里。
它想计算一下牧场中的草丛数量。
在牧场地图中，每个草丛要么是单个"#"，要么是有公共边的相邻两个“#”。
给定牧场地图，计算有多少个草丛。
输入格式：
第一行包含两个整数R和C，中间用单个空格隔开。
接下来R行，每行C个字符，描述牧场地图。字符只有“#”或“.”两种。(1<=*R*,*C*<=100)
输出格式：
输出一个整数，表示草丛数。
样例输入

```
5 6
.#....
..#...
..#..#
...##.
.#....
```

样例输出：

```
5
```

分析：

1、如何输入？
2、草丛有什么特点？
3、如何来计数草丛？

填空：

```
3   int r,c,ans;
4   string b[101];//b 矩阵
5   void sum() {
6       int x[4]= {0,-1,0,1};//左上右下
7       int y[4]= { (2) };
8       for(int i=0; i<r; i++)
9           for(int j=0; j<c; j++)
10              if(b[i][j]=='#') { //枚举每一个位置
11                  ans++;  //是# 则加1同时看他的上下左右4个位置有无#，并处理
12                  for(int k=0; k<4; k++)
13                      if(( (3) &&(j+y[k])>=0&&(j+y[k])<c&&b[i+x[k]][j+y[k]]=='#') {
14                          b[i+x[k]][j+y[k]]='.';//break;
15                      }
16              }
17      cout<<ans<<endl; // 输出答案
18  }
19  int main() {
20      cin>>r>>c;
21      for(int i=0; i< (1) ; i++)
22          cin>>b[i];//读入字符
23      sum();
24      return 0;
```

7、反反复复

Mo 和 $Larry$ 发明了一种信息加密方法。他们首先决定好列数，然后将信息（只包含字母）从上往下依次填入各列，并在末尾补充一些随机字母使其成为一个完整的字母矩阵。例如，若信息是 `There's no place like home on a snowy night` 并且有 5 列，Mo 会写成：

```
t o i o y
h p k n n
e l e a i
r a h s g
e c o n h
s e m o t
n l e w x
```

注意 Mo 只会填入字母，且全部是小写形式。在这个例子中，Mo 用字母 "x" 填充了信息使之成为一个完整的矩阵，当然他使用任何字母都是可以的。

Mo 根据这个矩阵重写信息：首先从左到右写下第一行，然后从右到左写下第二行，再从左到右写下第三行……以此左右交替地从上到下写下各行字母，形成新的字符串。这样，例子中的信息就被加密为：`toioynnkpheleaigshareconhtomesnlewx`。

你的工作是帮助 $Larry$ 从加密后的信息中还原出原始信息（包括填充的字母）。

第一行包含一个 $n(1 \leq n \leq 20)$，表示使用的列数。

第二行是一个长度不超过 200 的字符串。

输出

一行，即原始信息。

样例输入

```
5
toioynnkpheleaigshareconhtomesnlewx
```

样例输出

```
theresnoplacelikehomeonasnowynightx
```

分析：

1、确立数据结构（如何输入？存储？）

2、如何根据规则构建矩阵？

3、按规则输出矩阵

填空：

```
3  string s;
4  char a[20][30];
5  int main() {
6      int i,j,m,n,len,k=0;
7      cin>>n>>s; //读入
8      len= (1) ;//字符数
9      m=len/n; //求行数
10     for(i=0; i<m; i++) //构造密码矩阵
11         if( (2) )
12             for(j=0; j<n; j++)
13                 a[i][j]=s[k++];
14         else for(j= (3) ; j>=0; j--)
15                 a[i][j]=s[k++];
16     for( i=0; i<n; i++)  //按列优先输出
17         for(j=0; j<m; j++)
18             cout<< (4) ;
19     return 0;
20 }
21
```

8、矩阵剪刀石头布2

$Bart$ 的妹妹 $Lisa$ 在一个二维矩阵上创造了新的文明。

矩阵上每个位置被三种生命形式之一占据：石头，剪刀，布。

每天，上下左右相邻的不同生命形式将会发生战斗。在战斗中，石头永远胜剪刀，剪刀永远胜布，布永远胜石头。

每一天结束之后，败者的领地将被胜者占领。

你的工作是计算出 n 天之后矩阵的占据情况。

输入

第一行包含三个正整数 r，c，n，分别表示矩阵的行数、列数以及天数。每个整数均不超过 100。

接下来 r 行，每行 c 个字符，描述矩阵初始时被占据的情况。

每个位置上的字符只能是 R，S，P 三者之一，分别代表石头，剪刀，布。相邻字符之间无空格。

输出

输出 n 天之后的矩阵占据情况。

每个位置上的字符只能是 R，S，P 三者之一，相邻字符之间无空格。

```
3 3 1
RRR
RSR
RRR
```

样例输出

```
RRR
RRR
RRR
```

主要代码：

```
3   char a[101][101], tmp[101][101];
4   int d[4][2]= {{1,0}, {-1,0}, {0,1}, {0,-1}},r,c,n;//四个方向
5   bool judge(char a, char b) {//判断胜负
6       return (a=='R' && b=='S')||(a=='S'&& b=='P')||(a=='P'&&b=='R');
7   }
8   void process() {//比赛
9       memcpy(tmp, a, sizeof(a));// 借助另一数组存胜负关系
10      for (int i = 0; i < r; i++)
11          for (int j = 0; j < c; j++)
12              for (int k = 0; k < 4; k++) {
13                  int tx = i + d[k][0], ty = j + d[k][1];
14                  if(tx>=0 && tx<r && ty>=0 && ty<c && judge(a[tx][ty], a[i][j]))
15                      tmp[i][j]=a[tx][ty];
16              }
17      memcpy(a,tmp,sizeof(tmp));//一天比完后，为下天准备，
18  }
19  int main() {
20      cin>>r>>c>>n;
21      for(int i = 0; i < r; i++)
22          cin >> a[i];
23      for (int i = 0; i < n; i++)//比赛n天
24          process();
25      for (int i = 0; i < r; i++)//输出
26          cout << a[i] << endl;
```

9、扫雷游戏地雷数计算

样例输入

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

样例输出

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

10、古风排版

输入样例：

```
4
This is a test case
```

输出样例：

```
asa T
st ih
e tsi
 ce s
```