第五章 数的存储与组织

第65课 二维数组上的递归

《信息学奥赛一本通·编程启蒙 C++版》

【例 65.1】 循环比赛

【题目描述】

设有 N 个选手进行循环比赛，其中 $N=2^M$，要求每名选手要与其他 N-1 名选手都赛一次，每名选手每天比赛一次，循环赛共进行 N-1 天，要求每天没有选手轮空。

【输入格式】

输入：M

【输出格式】

输出：表格形式的比赛安排表。一行各数据间用一个空格隔开。

【样例输入】

3

【样例输出】

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

【代码实现】

```cpp
#include<bits/stdc++.h>
using namespace std;
int n;
int a[1005][1005];
void deal(int x){
   if(x==2){
      a[1][1]=1;
      a[1][2]=2;
      a[2][1]=2;
      a[2][2]=1;
      return ;
   }
   deal(x/2);
   for(int i=1;i<=x/2;i++){
      for(int j=1;j<=x/2;j++){
         a[i+x/2][j]=a[i][j]+x/2;
         a[i][j+x/2]=a[i][j]+x/2;
         a[i+x/2][j+x/2]=a[i][j];
      }
   }
}
int main(){
   cin>>n;
   n=(1<<n);
   deal(n);
   for(int i=1;i<=n;i++){
      for(int j=1;j<=n;j++){
         cout<<a[i][j]<<" ";
      }
      cout<<endl;
   }
   return 0;
}
```

【例 65.2】 清除地雷

【题目描述】

在一张 n * m 的地图上，有一些地雷，用 1 表示，其余没有地雷的地方用 0 表示；如果某个地雷被引爆了，那么它的波及范围会炸掉 周围八个位置的土地和该地雷所在位置的土地，如果在它的八个位置上也有地雷，那么这些地雷将会连锁反应，继续引爆，爆炸后的土地用 2 表示。请你编写程序，输出引爆某个地雷后地图的模样。

【输入格式】

输入共计 n + 2 行：

第一行包含两个整数：n（1≤n≤100），m（1≤m≤100），表示地图的大小为 n* m。

第 2 到 n+1 行，每行包含 m 个字符，表示地图的样貌。

第 n+2 行包含两个整数：x（1≤x≤n），y（1≤y≤m），表示地图上指定开始(引爆)的坐标。如果指定的坐标是土地，也就不需要引爆任何地雷。

【输出格式】

输出 n 行，每行包含 m 个数，表示爆炸后的地图模样。

【样例输入】

```
5 5
10100
01000
00000
00000
00001
1 1
```

【样例输出】

```
22220
22220
22200
00000
00001
```

【代码实现】

```
#include<bits/stdc++.h>
using namespace std;
int n,m,x,y,dx[8]={-1,-1,0,1,1,1,0,-1},dy[8]={0,1,1,1,0,-1,-1,-1};
char boom[105][105];
void f(int x,int y){
   boom[x][y]='2';
   for(int i=0;i<8;i++){
     int nx=dx[i]+x,ny=dy[i]+y;
      if(nx>=1&&nx<=n&&ny>=1&&ny<=m){
         if(boom[nx][ny]=='1') f(nx,ny);
         boom[nx][ny]='2';}
      }
   }
}
int main(){
   cin>>n>>m;
   for(int i=1;i<=n;i++)
      for(int j=1;j<=m;j++) cin>>boom[i][j];
   cin>>x>>y;
   if(boom[x][y]=='1') f(x,y);
   for(int i=1;i<=n;i++){
      for(int j=1;j<=m;j++) cout<<boom[i][j];
      cout<<endl;
   }
   return 0;
}
```

【例 65.3】 细胞个数

【题目描述】

一矩形阵列由数字 0 到 9 组成，数字 1 到 9 代表细胞，细胞的定义为沿细胞数字上下左右还是细胞数字则为同一细胞，求给定矩形阵列的细胞个数。如：

阵列。

```
4 10
0234500067
1034560500
2045600671
0000000089
```

有 4 个细胞。

【输入格式】

第一行为矩阵的行 n（1≤n≤100），列 m（1≤m≤100）。

下面为一个 n x m 的矩阵。

【输出格式】

细胞个数。

【样例输入】

```
4 10
0234500067
1034560500
2045600671
0000000089
```

【样例输出】

```
4
```

【代码实现】

```
#include<bits/stdc++.h>
using namespace std;
int n,m,ans;
string s;
int c[100][100];
int dx[10]={-1,0,1,0},
    dy[10]={0,1,0,-1};
bool check(int x,int y){
  return c[x][y]>=1 && c[x][y]<=9 && x>=1 && x<=n && y>=1 && y<=m;
}
void dfs(int x,int y){
  for(int i=0;i<=3;i++){
    if(check(x+dx[i],y+dy[i])){
      c[x+dx[i]][y+dy[i]]=0;
      dfs(x+dx[i],y+dy[i]);
    }
  }
}
int main(){
  cin>>n>>m;
  for(int i=1;i<=n;i++){
    cin>>s;
    for(int j=1;j<=m;j++){
      c[i][j]=s[j-1]-48;
    }
  }
  for(int i=1;i<=n;i++){
    for(int j=1;j<=m;j++){
      if(c[i][j]>=1 && c[i][j]<=9){
        ans++;
        c[i][j]=0;
        dfs(i,j);
      }
    }
  }
  cout<<ans;
  return 0;
}
```

练 65.1 水洼个数

【题目描述】

有一块 N×M 的土地，雨后积起了水，有水标记为‘W’，干燥为‘.’。八连通的积水被认为是连接在一起的。请求出院子里共有多少水洼？

【输入格式】

第一行为 N,M(1≤N,M≤100)。下面为 N*M 的土地示意图。

【输出格式】

一行，共有的水洼数

【样例输入】

```
10 12
W........WW.
.WWW.....WWW
....WW...WW.
.........WW.
.........W..
..W......W..
.W.W.....WW.
W.W.W.....W.
.W.W......W.
..W.......W.
```

【样例输出】

```
3
```

【代码实现】

```
#include<bits/stdc++.h>
using namespace std;
int n,m,cnt,color[120][120],
  dx[8]={-1,1,0,0,-1,-1,1,1},
  dy[8]={0,0,-1,1,-1,1,-1,1};
  char c[120][120];
  bool check(int x,int y){
     return  c[x][y]=='W'&&color[x][y]==0&&x>=1&&x<=n&&y>=1&&y<=m;
}
void dfs(int x,int y){
  color[x][y]=cnt;
  for(int i=0;i<8;i++){
     int nx=x+dx[i],ny=y+dy[i];
     if(check(nx,ny))dfs(nx,ny);
  }
}
int main(){
  cin>>n>>m;
  for(int i=1;i<=n;i++)
     for(int j=1;j<=m;j++)
        cin>>c[i][j];
  for(int i=1;i<=n;i++)
     for(int j=1;j<=m;j++){
        if(check(i,j)){
           cnt++;
           dfs(i,j);
        }
     }
  cout<<cnt;
  return 0;
}
```

练 65.2 跳房子

【题目描述】

奶牛们按不太传统的方式玩起了小孩子们玩的"跳房子"游戏。奶牛们创造了一个 5*5 的、由与 x,y 轴平行的数字组成的直线型网格，而不是用来在里面跳的、线性排列的、带数字的方格。 然后他们熟练地在网格中的数字中跳：向前跳、向后跳、向左跳、向右跳(从不斜过来跳)，跳到网格中的另一个数字上。他们再这样跳啊跳(按相同规则)，跳到另外一个数字上(可能是已经跳过的数字)。 一共在网格内跳过五次后，他们的跳跃构建了一个六位整数(可能以 0 开头，例如 000201)。 求出所有能被这样创造出来的不同整数的总数。

【输入格式】

第 1 到 5 行：这样的网格，一行 5 个整数。

【输出格式】

第 1 行：能构建的不同整数的总数

【样例输入】

```
1 1 1 1 1
1 1 1 1 1
1 1 1 1 1
1 1 1 2 1
1 1 1 1 1
```

【样例输出】

```
15
```

【代码实现】

```
#include<bits/stdc++.h>
using namespace std;
int Map[5][5];
int dx[]={1,-1,0,0};
int dy[]={0,0,1,-1};
set<int> res;
void dfs(int x,int y,int step,int number){
   if(step==5){
      res.insert( number) ;
      return;
   }

   for(int i=0;i<4;i++){
      int nx=x+dx[i];int ny=y+dy[i];
      if(0<=nx&&nx<5&&0<=ny&&ny<5){
         dfs(nx,ny,step+1,10*number+Map[nx][ny]);
      }
   }
}
int main(){
   for(int i=0;i<5;i++)
      for(int j=0;j<5;j++)
         cin>>Map[i][j];

   res.clear();
   for(int i=0;i<5;i++)
      for(int j=0;j<5;j++)
         dfs(i,j,0,Map[i][j]);
   cout<<res.size()<<endl;
   return 0;
}
```

练 65.3　螺旋矩阵

【题目描述】

一个 n 行 n 列的螺旋矩阵可由如下方法生成:

从矩阵的左上角(第 1 行第 1 列)出发，初始时向右移动;如果前方是未曾经过的格子，则继续前进，否则右转;重复上述操作直至经过矩阵中所有格子。根据经过顺序，在格子中依次填入 1,2，3,... ，　n，便构成了一个螺旋矩阵。

下图是一个 n=4 时的螺旋矩阵。

1	2	3	4
12	13	14	5
11	16	15	6
10	9	8	7

【输入格式】

输入共一行，包含三个整数 n,i,j，每两个整数之间用一个空格隔开，分别表示矩阵大小、待求的数所在的行号和列号。

1≤n≤30,000 ,1≤i≤n,1≤j≤n

【输出格式】

输出一个整数，表示相应矩阵中第 i 行第 j 列的数。

【样例输入】

4 2 3

【样例输出】

14

【代码实现】

```
#include<bits/stdc++.h>
using namespace std;
int n,x,y;
int dfs(int n,int x,int y,int cnt){
    if(x==1) return y+cnt;
    if(y==n) return n+x-1+cnt;
    if(x==n) return 2*n-1+n-y+cnt;
    if(y==1) return 3*n-2+n-x+cnt:
    return dfs(n-2,x-1,y-1,cnt+4*(n-1));
}
int main(){
    cin>>n>>x>>v:
    cout<<dfs(n,x,y,0);
    return 0;
}
```

谢谢！