

C 胡乱走的和尚——by Arthur

题目描述:

从前有座山，山里有座庙，庙里有个不知姓名的老和尚和小和尚~~~o(∩_∩)o~~~

前两天，什么事情也没有发生呢~! OwO! ~

这天，老和尚发现了一片错落有致的正方形区域上，如下图：

```
1  11 21 31 41 51 61 71 81 91
2  12 22 32 42 52 62 72 82 92
3  13 23 33 43 53 63 73 83 93
4  14 24 34 44 54 64 74 84 94
5  15 25 35 45 55 65 75 85 95
6  16 26 36 46 56 66 76 86 96
7  17 27 37 47 57 67 77 87 97
8  18 28 38 48 58 68 78 88 98
9  19 29 39 49 59 69 79 89 99
10 20 30 40 50 60 70 80 90 100
```

这居然还编号了!!! ●▽●

方阵是从左上角的 1 按照图中顺序到右下角的 n^2

于是老和尚让小和尚蒙着眼站在 1 处，然后给他一系列指令

分别是向上(U)（或者左(L)，下(D)，右(R)）走整数 k 个格子

执行完这一系列的走位之后呢，看看小和尚知不知道自己站在几号格子上

小和尚需求大家程序的帮助啊!!!

（老和尚很阴险，有可能让小和尚走出方阵外，这种时候输出"WanQuanGaoBuDong!"（不带引号）就好了

输入格式:

第一行为组数 T, T 为整数且 $T \leq 10$;

接下来 T 组，每组数据为 m+1 行，

第一行包含两个整数 n, m，表示方阵边长 n ($1 \leq n \leq 10000$)，指令数量 m ($1 \leq m \leq 10000$);

下面 m 行，每行一个字母 x 和一个整数 k，字母 x 为上(U)、下(D)、右(R)、左(L)其中的一个， k ($1 \leq k \leq 10000$) 为整数

输出格式:

对于每组测试数据，输出一行，为最后所在位置的数字，
如果在过程中出界了，输出"WanQuanGaoBuDong!"（不带引号）

sample in:

```
2
10 2
R 5
D 5
10 1
U 1
```

sample out:

```
56
WanQuanGaoBuDong!
```

解题分析

首先看这一句——表示方阵边长 $n(1 \leq n \leq 10000)$ 一万乘以一万的二维数组大家可以尝试一下，是开不下的，编译器会报错。

然而本题实际上完全不需要二维数组，观察数字排列的顺序，不难看出若一以左上角坐标为 $(0,0)$ ，任意一点 (x,y) 的数值是关于 x 和 y 的函数，即：

$$\text{num}(x,y) = (x - 1) * n + y$$

而指令也都是对这横纵坐标 x 、 y 的变换，因此只需要定义两个变量来记录当前所在位置就可以了。

然而要注意两个问题

- ① 越界的判断是在过程之中每一步都要判断的，否则如果中间走了出去然而之后的操作又使其走回来，就会出错。这个可以通过在中间使用一个 **flag** 布尔变量来记录，详见代码
- ② 即使在某一步不合法，也不可以 **break**。除非你在这之前已经把这一组操作全部读了进来存在数组里待用，否则你在中间某一步 **break** 将会导致下一组数据仍然会从你这一组数据里未被读入的部分开始读入，这样也会出错

参考代码

```
#include<iostream>
using namespace std;
bool isLegal(int x,int y,int n)//判断函数
{return x<=n&&x>=1&&y<=n&&y>=1;}//越界情况一共四种
bool shift(bool &legal,int &x,int &y,char c,int l,int n)
{
    switch(c)//根据方向来判断如何走
    {
        case 'U':y-=1;break;//上
        case 'D':y+=1;break;//下
        case 'R':x+=1;break;//右
        case 'L':x-=1;break;//左
    }
    if(!isLegal(x,y,n))    legal=false;
}    //如果不合法，将 legal 置为 false
int main()
{
    int T,n,m,l;cin>>T;
    while(T--)
    {
        char c;    //方向
        int x=1,y=1;//初始坐标
        bool legal=true;//legal 默认为 true，为判断合法的 flag
        cin>>n>>m;
        for(int i=0;i<m;i++)
        {
            cin>>c>>l;
            shift(legal,x,y,c,l,n);
        }    //一次变换，注意这里的一些变量是按引用传递
        (legal)?cout<<x*n+y-n<<endl:cout<<"WanQuanGaoBuDong!"<<endl;
    } //根据最终结果决定如何输出
}
```