

[ICPC2010 国内予選問題 A] 角角画伯, かく悩みき

シミュレーション問題。入力データ "ni di" は、i 番目の正方形を ni 番の正方形に対して di で示される方角に隣接して置くことを指示している。この指示をシミュレートするためには、各正方形の座標が必要となるので、i 番目の正方形の x 座標を格納する配列 x[] と y 座標を格納する配列 y[] を用いる。

最初の正方形はどこに置いてもいいので(0,0)に置くことにする。

di の方角の隣接位置（左側 = 0、下側 = 1、右側 = 2、上側 = 3）に置くには
int dx[] = {1, 0, 1, 0}; int dy[] = {0, -1, 0, 1}; としておいて、

x[i] = x[ni] + dx[di]; y[i] = y[ni] + dy[di];

とすれば switch 文などによる場合分けは不要となる。

正方形を一つ置くたびに、正方形の x 座標の最小値 xmin と最大値 xmax、y 座標の最小値 ymin と最大値 ymax を更新しておく。

最終的に、求める幅は $xmax - xmin + 1$ 、高さは $ymax - ymin + 1$ で与えられる。

プログラム例

```
// ACM-ICPC 2010 Japan Online Contest Problem A
// http://icpc2010.honiden.nii.ac.jp/domestic-contest/problems#section_A
//   ファイル名: pa.c
//   コンパイル方法: cc pa.c
//   実行方法: ./a.out <A0> A0.result など
//   チェック方法: diff A0.ans A0.result など
//
// アルゴリズム概要
// 最初の正方形は(0,0)に置くものとする。
// 2 枚目以降の正方形の置く位置は入力データより求める。
// 各正方形の座標は、配列 x[],y[] に記憶させる。
// x 座標の最小値、最大値、y 座標の最小値、最大値を求める。
// 幅は、x 座標の最大値 - x 座標の最小値 + 1,
// 高さは、y 座標の最大値 - y 座標の最小値 + 1 である。

#include <stdio.h>

#define MAXN 200           // 1 <= N < 200
int x[MAXN];              // x[i] は i 番目の正方形の x 座標
int y[MAXN];              // y[i] は i 番目の正方形の y 座標
int xmin, xmax;          // x 座標の最大値と最小値
int ymin, ymax;          // y 座標の最大値と最小値
int n;                    // 正方形の数

// 左(i=0),下(i=1),右(i=2),上(i=3)に置く時の
// x,y 座標の変化値を dx[i], dy[i]とする
int dx[] = {1,0,1,0};
int dy[] = {0,-1,0,1};
```

```

int main()
{
    int i;
    int ni,di;
    while(1){
        scanf("%d", &n);        // N の読み込み
        if(n==0) break;        // N = 0 なら終了
        // まず最初(0 番目)の正方形を位置(0,0)に置く
        x[0] = y[0] = 0;
        xmin = xmax = ymin = ymax = 0;
        for(i=1; i <= n-1; i++){
            scanf("%d%d", &ni, &di);    // ni と di を入力
            // ni 番目のブロックにたいして di 方向の隣接位置に置く
            x[i] = x[ni] + dx[di];
            y[i] = y[ni] + dy[di];
            // x, y 座標の最大値、最小値を更新
            if(x[i] < xmin) xmin = x[i];
            if(x[i] > xmax) xmax = x[i];
            if(y[i] < ymin) ymin = y[i];
            if(y[i] > ymax) ymax = y[i];
        }
        // 幅と高さを出力
        printf("%d %d¥n", xmax-xmin+1, ymax-ymin+1);
    }
}

```