

[ICPC2009 国内予選問題 A] [次期町長](#)

- 「ゲームは百万ステップ以内に終了する」と記載されており、また「ゲームのステップ」もそれほど複雑なものではないので、基本的なアルゴリズムとしては「ゲームのステップ」を勝者が決まるまで繰り返すことで良い。すなわち、ゲームのステップをシミュレートするプログラムを書けばよい。
- ゲームの終了条件として「ある候補者が碗に残った最後の小石を取り出したとき、他のどの候補者の手元にも小石がなければ、ゲームは終わり、全部の小石を持っているその候補者が勝者となる」と記載されている。これは「碗から小石を 1 個取り出して手許においたときに手許にある小石の数が小石の総数(p)に等しい」ことと等価である。このことをチェックすれば「他の候補者の手許に小石が無い」ことを確認する必要はない。

プログラム例

```
/* ACM-ICPC2009 国内予選 Problem A */
// http://www.waseda.jp/assoc-icpc2009/preliminary/contest/all_ja.html#section_A
// filename = pa1.c
// コンパイル方法 : cc -O2 pa1.c
// 実行方法 : ./a.out < A0 > A0.result 等
// 確認方法 : diff A0.ans A0.result 等
// アルゴリズム : 「ゲームのステップ」をシミュレートする
//                  碗の中から小石を 1 個取り出して手元に置いたときに、
//                  手元にある小石の個数が小石の総数になればその候補者が勝者となる

#include <stdio.h>

#define MAXN 50          // 候補者数(n)の最大値

int person[MAXN];      // 各候補者が持っている小石の個数
int pos;               // 碗の位置
int wan;               // 碗の中の小石の個数
int n;                 // 候補者数 3<=n<=50
int p;                 // 小石の総数 2<=p<=50

void solve()           // 「ゲームのステップ」を繰り返す関数
{
    while(1) {
        if(wan) {
            person[pos]++; // 碗の中に小石が入っていれば
            person[pos]--; // 碗の中から小石を 1 個取り出し手元に置く
            wan--;        // 碗の中の小石の個数は一つ減る
            if(person[pos]==p) { // この時点ですべての小石が手元にあれば勝ち
                printf("%d\n", pos); // 勝者の番号をプリント
                return;
            }
        }
        else {
            wan = person[pos]; // 碗の中に小石が無ければ
            person[pos] = 0;   // 手元にある小石をすべて碗に入れる
                                // 手元にある小石の個数は 0 になる
        }
    }
}
```

```

    pos++;
    if(pos == n) pos = 0;
}
}

int main()
{
    int i;
    while(1){
        scanf("%d %d", &n, &p);
        if(n==0 && p==0) break;
        for(i=0; i<n; i++) person[i]=0;
        pos = 0; wan = p;
        solve();
    }
}

```