

```

//输入
int N;
int dir[MAX_N]; //牛的方向（0:F, 1:B）

int f[MAX_N]; //区间[i,i+K-1]是否进行反转

//固定 K，求对应的最小操作回数，无解的话则返回-1
int calc(int K) {
    memset(f, 0, sizeof(f));
    int res = 0;
    int sum = 0; // f 的和
    for (int i = 0; i + K <= N; i++) {
        //计算区间[i, i+K-1]
        if ((dir[i] + sum) % 2 != 0) {
            //前端的牛面朝后方
            res++;
            f[i] = 1;
        }
        sum += f[i];
        if (i - K + 1 >= 0) {
            sum -= f[i - K + 1];
        }
    }

    //检查剩下的牛是否有面朝后方的情况
    for (int i = N - K + 1; i < N; i++) {
        if ((dir[i] + sum) % 2 != 0) {
            //无解
            return -1;
        }
        if (i - K + 1 >= 0) {
            sum -= f[i - K + 1];
        }
    }

    return res;
}

void solve() {
    int K = 1, M = N;
    for (int k = 1; k < N; k++) {
        int m = calc(k);
        if (m >= 0 && M > m) {
            M = m;
            K = k;
        }
    }
    printf("%d %d\n", K, M);
}

```