#### 배짱이 당첨자 추첨 제출 완료

우아한형제들은 배짱이(배민을 짱 좋아하는 이들의 모임)라는 팬클럽이 있습니다. 배짱이 팬클럽 모임을 가지던 중, 한 분을 뽑아서 365일 치킨 무료 쿠폰을 증정하기로 했는데요. 쿠폰을 받게 될 사람은 다음 규칙을 통해 선정합니다. 입력으로 n(배짱이 인원 수), d(방향을 나타내는 수), k, j가 주어졌다고 가정합니다.

1. 배짱이 분들에 1부터 n까지 번호를 매깁니다.
2. 1번부터 n번 사람까지 시계 방향으로 원을 만들어서 자리를 잡습니다. n번째 사람 옆에는 1번 사람이 있게 됩니다.
3. 탈락자를 뽑을 때, 시계 방향으로 진행할 지 반시계 방향으로 진행할 지를 정합니다. 방향이 1이면 시계 방향, 0이면 반시계 방향입니다.
4. 한 사람이 남을 때까지, 다음과 같은 과정을 반복하면서 탈락자를 선정합니다. 처음의 시작점은 1번 사람입니다.
   1. 현재의 지점으로부터 ‘규칙 3’에서 정한 방향으로 k번째 있는 사람이 탈락합니다. 예를 들어 8명으로 시작했을 때(1번이 현재 지점)
      1. 방향이 1이고(시계 방향) k가 2이면 첫번째 탈락자는 3번 사람이 되고,
      2. 방향이 0이고(반시계 방향) k가 3이면 탈락자는 6번 사람이 됩니다.
   2. 탈락자는 탈락 표시를 붙이고, 그 자리에 서 있으면서 다음 번 탈락자를 정하기 위한 출발점이 됩니다.
   3. 탈락자가 정해지면, k를 j만큼 증가시킵니다. 예를 들어 k가 3이고, j가 2이면 그 다음 탈락자를 정할 때는 정해진 방향으로 돌면서 5번째 있는 사람이 탈락자가 되고, 그 다음 번에는 7번째 있는 사람이 탈락자가 됩니다.
   4. k만큼 이동할 때는, 탈락자는 제외하고 남아 있는 사람들만 고려하여 이동합니다.
5. 최후의 한 사람이 남으면, 그 사람이 당첨자가 됩니다.

예를 들어 배짱이가 6명, 방향이 1이고, k와 j가 모두 1이라고 하면 다음과 같이 탈락자가 결정됩니다.

* 1번에서 출발하여 k만큼(1만큼) 이동하여 2번 탈락. k = k + j(1) = 2
* 2번에서 출발하여 k만큼(2만큼) 이동하여 4번 탈락. k= k + j = 3
* 4번에서 출발하여 k만큼(3만큼) 이동하여 1번 탈락. k = 4
* 1번에서 출발하여 k만큼(4만큼) 이동하여 3번 탈락
  + 1 -> 3 -> 5 -> 6-> 3 (1, 2, 4는 이미 탈락했으므로 제외)
* 3번에서 출발하여 k만큼(5만큼) 이동하여 5번 탈락
  + 3 -> 5 -> 6 -> 5 -> 6 -> 5
* 마지막에 6번이 남았으므로 6번이 당첨자가 됨

배짱이 팬클럽 인원 수 n, d(방향: 1이면 시계 방향, 0이면 반시계 방향), k, j가 공백 문자로 구분되어 한 줄로 표준 입력으로 주어졌을 때, 최종 당첨자가 누구인지 출력하는 프로그램을 작성하십시오.

* n는 1 이상 10,000,000 이하의 자연수가 주어집니다.
* d는 1이면 시계방향, 0이면 반시계 방향입니다.
* k와 j는 1 이상 1,000,000 이하의 자연수가 주어집니다.

입/출력 예시

⋇ 입출력 형식을 잘 지켜주세요.

␣

 : 공백

↵

 : 줄바꿈

보기 입력 1

6 ␣1 ␣1 ␣1

6 1 1 1

출력 1

6

6

보기 입력 2

4 ␣0 ␣2 ␣2

4 0 2 2

출력 2

1

1

//Please don't change class name 'Main'

import java.util.Scanner;

class Main {

public static void main(String[] args) {

try(Scanner s = new Scanner(System.in))

{

int n = s.nextInt();

int d = s.nextInt();

int k = s.nextInt();

int j = s.nextInt();

//여기부터 작성해 주세요

int move = 0;

int site = 1;

int number = 0;

int[] life = new int[n+1];

int reverse = n+2;

while(true){

move = k;

while(true){

site = ( site + 1 ) % n;

if( site == 0 ){

site = n;

}

if( life[site] == 0 ){

move--;

if( move == 0 ){

break;

}

}

}

life[site]++;

number++;

if( number == n-1 ){

break;

}

k = k + j;

}

if( d == 1 ){

for(int i = 1 ; i < n+1 ; i++){

if( life[i] == 0 ){

System.out.println(i);

}

}

}else{

for(int i = 1 ; i < n+1 ; i++){

reverse--;

if( life[i] == 0 ){

if(i == 1){

System.out.println(i);

break;

}

if( life[i] == 0 ){

System.out.println(reverse);

}

}

}

}

}

}

}