

## Problem B. Board Blitz

Input file:            board-blitz.in  
Output file:           board-blitz.out  
Time limit:            2 seconds  
Memory limit:         256 mebibytes

*In this problem, you have to quickly find the final position in a game on a square board.*

Consider a game on a square checkered board divided into  $8 \times 8$  square cells. Initially, the board is empty. Two players take turns one after another. The first player controls white pieces, the second player controls black pieces. A turn consists in either a movement or a jump of the player's pieces.

To make a movement, a player chooses one of the four directions and shifts all her pieces simultaneously by one cell in that direction. If a piece moves off the board, it disappears. Furthermore, for each of the eight border cells where no piece could have been shifted to during a movement in this direction, the player may place a new piece of her color in that cell. If a piece appears on a cell which contains a piece of opposite color, that piece (of opposite color) disappears from the board.

To make a jump, a player takes all her pieces and simultaneously transfers them to cells which correspond to a 90-degree rotation around the board center. The rotation must be clockwise for white pieces and counter-clockwise for black pieces. Same as after a movement, if a piece appears on a cell which contains a piece of opposite color, that piece (of opposite color) disappears from the board.

Both players agreed to make turns according to the values obtained from a random number generator: the first random number determines the first player's turn, the second number gives the second player's turn, the third number specifies the next turn of the first player, and so on.

You are given the parameters  $a$ ,  $c$  and  $s$  of a linear congruential pseudorandom number generator in a 64-bit signed integer data type. This generator works in the following way: to get the next random number, the assignment  $s_{\text{new}} := s_{\text{old}} \cdot a + c$  is carried out, the result is truncated to a 64-bit signed integer data type, stored as the new value of  $s$  and returned as the next random number.

A turn is determined by the number  $p$  obtained as the highest 11 bits of the random number (in other words,  $p$  is the result of an integer division of  $s$  by  $2^{53}$  rounded towards zero). If  $p < 0$ , the turn will be a jump. Otherwise, the turn will be a movement, bits 8 (multiplied by 1) and 9 (multiplied by 2) form the number of direction (0 is up, 1 is left, 2 is down, 3 is right), and bits 0 through 7 specify which of the eight cells will contain new pieces. The cells are numbered 0 through 7 from top to bottom or from left to right (depending on the direction).

Given numbers  $a$ ,  $c$  and  $s$ , and also the number of half-turns  $n$ , find the final position on the board. A half-turn is a turn by one of the two players. It is guaranteed that  $n$  is even. Additionally, it is guaranteed that in all the judges' tests except the example, the triple of numbers  $a$ ,  $c$  and  $s$  is chosen uniformly at random from all such triples that the linear congruential generator specified by them has a period of exactly  $2^{64}$ .

### Input

The first line of input contains an integer  $n$ , the number of half-turns in the game, and also integers  $a$ ,  $c$  and  $s$  which are the parameters of the random number generator ( $2 \leq n \leq 30\,000\,000$ ,  $n$  is even,  $-2^{63} \leq a, c, s < 2^{63}$ , the period of the generator is exactly  $2^{64}$ ).

## Output

Print eight lines with eight characters on each line: the state of the board after all turns have been completed. Denote empty cells with character ".", white pieces with character "W" and black pieces with character "B".

## Example

board-blitz.in	board-blitz.out
10 12345678901234565 23456789012345679 12345678	.BB..... .....BB ..... WW..... WW..... W.....B W..... .W.....

## Explanation

The following table shows the sequence of values of  $s$ , values of  $p$  and specific bits of  $p$  in the example.

s	p	bit 10	bits 9–8	bits 7–0
8800325836438854357	977	0	11	11010001
-6158326326699848968	-684	1	01	01010100
-8075422879926989273	-897	1	00	01111111
3701688810043832978	410	0	01	10011010
1958856781360031529	217	0	00	11011001
594250524398489244	65	0	00	01000001
-6137059809998356901	-682	1	01	01010110
634828852634288534	70	0	00	01000110
8001533107637205053	888	0	11	01111000
-4716194469178570240	-524	1	01	11110100

The pictures below show all the intermediate states of the board in the example.

