```c
#include <stdio.h>

// All the winning configurations
int win[8][3] = {{0, 1, 2}, // Check first row.
                 {3, 4, 5}, // Check second Row
                 {6, 7, 8}};

// Is the character c winning the given board?
int isCWin(char *board, char c){
    for(int i=0; i<3; i++)
        if((board[win[i][0]] == c) && (board[win[i][1]] == c) && (board[win[i][2]] == c))
            return 1;
    return 0;
}

// A board is valid if and only if it has 5 X and 4 O
int isValid(char *board){
    int xCount=0, oCount=0;
    for(int i = 0; i < 9; i++){
        if(board[i]=='X')
            xCount++;
        if(board[i]=='O')
            oCount++;
    }
    if(xCount == oCount+1)
        return 1;
    else
        return 0;
}

void genWinX(char *board, int *blanks, int b, int done){
    if(done == b){ // All blanks filled
        // Validate and print if X is winning
        if(isValid(board) && isCWin(board, 'X'))
            printf("%s\n", board);
        return;
    }
    board[blanks[done]] = 'O'; // Lexicographic order: O comes before X
    genWinX(board, blanks, b, done+1);
    board[blanks[done]] = 'X';
    genWinX(board, blanks, b, done+1);
}

int main(){
    char board[10];
    board[9] = '\0';
    int blanks[9];
    int b = 0;
    char c;
    for(int i = 0; i < 9; i++){
        c = getchar();
        if(c == 'B')
            blanks[b++] = i; // This is the next blank position
        board[i] = c;
    }
    genWinX(board, blanks, b, 0);
}
```