```c
#include <stdio.h>

void generatePalindromes(char* str, int k, int n, int next){
    // If the next array location to be filled in is the same as its
    // mirror image, then k must be odd and we have just the middle
    // letter to fill in e.g. mala*alam (only * remains to be filled)
    if(next == k - next - 1){
        // Fill the missing space with alphabets in increasing order
        // to satisfy the lexicographic order mentioned in the question
        for(int i = 0; i < n; i++){
            str[next] = 'a' + i;
            printf("%s\n", str);
        }
        return;
    }
    // This means that k is even and only the middle two elements are left
    // For example abc**cba
    if(next + 1 == k - next - 1){
        for(int i = 0; i < n; i++){
            // Make sure both places are the same alphabet
            str[next] = str[next+1] = 'a' + i;
            printf("%s\n", str);
        }
        return;
    }
    // General case: just fill the position and its mirror image
    // with the same alphabet
    for(int i = 0; i < n; i++){
        str[next] = str[k - next - 1] = 'a' + i;
        generatePalindromes(str, k, n, next + 1);
    }
}

int main(){
    int n, k;
    scanf("%d %d", &n, &k);
    char str[k+1];
    str[k] = '\0'; // All palindromes are of length k
    // Generate palindromes of length k with the first n letters
    // Till now 0 letters have been filled in so the next array
    // index to be filled in is location 0
    generatePalindromes(str, k, n, 0);
    return 0;
}
```