```c
#include <stdio.h>
#include <string.h>

void compare(char *str1, char* str2, int *k, int *l, int *isSame){
    char tmp[200];
    int len1 = strlen(str1);
    // Check for every possible rotation of the alphabet
    for(int i = 0; i < 26; i++){
        strcpy(tmp, str1); // Make a copy of str1 as we will modify it
        // Use the rotated alphabet
        for(int j = 0; j < len1; j++)
            tmp[j] = (str1[j] - 'a' + i) % 26 + 'a';

        // Cute trick to check if str2 is a rotated version of tmp
        // The string purkan is a rotated version of the string kanpur
        // However, this can only happen if purkan is a substring of
        // the string kanpurkanpur. Indeed purkan does appear at index
        // 3 inside the string kanpurkanpur
        strcat(tmp, tmp);

        char *p = strstr(tmp, str2);
        if(p != NULL){ // Found it!
            *isSame = 1;
            *k = i;
            *l = (len1 - (p - tmp)) % len1;
            return;
        }
    }
    // No success :(
    *isSame = 0;
    *k = -1;
    *l = -1;
}

int main() {
    char s1[100], s2[100];
    gets(s1);
    gets(s2);

    // If the two strings are not even of the same length
    /// Then there is no way one can be transformed into the other
    if(strlen(s1) != strlen(s2)){
       printf("NO");
       return 0;
    }

    int k, l, isSame;
    compare(s1, s2, &k, &l, &isSame);
    if(isSame)
        printf("YES %d %d", k, l);
    else
        printf("NO");
}
```