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

int checkMatch(char **mat, char *str, int i0, int j0, int m, int n, int len, int iInc, int jInc){
    int i;
    for(i = 0; i < len; i++){
        if((i0 + i * (iInc)) >= m)
            return 0; // Oops - we went past the limits of the matrix

        if((j0 + i * (jInc)) >= n)
            return 0; // Oops - we went past the limits of the matrix

        if((i0 + i * (iInc)) < 0)
            return 0; // Oops - we went past the limits of the matrix

        if((j0 + i * (jInc)) < 0)
            return 0; // Oops - we went past the limits of the matrix

        if(mat[i0 + i * (iInc)][j0 + i * (jInc)] != str[i])
            return 0; // Even 1 mismatch means message not present
    }
    return 1; // Hmm ... message must be present
}

int main(){
    int m, n, i, j, len, count = 0;
    scanf("%d %d\n", &m, &n);
    char str[100];
    gets(str);
    char **mat = (char**)malloc(m * sizeof(char*));
    for(i = 0; i < m; i++){
        mat[i] = (char*)malloc(n * sizeof(char));
        for(j = 0; j < n; j++)
            mat[i][j] = getchar();
        getchar();
    }
    len = strlen(str);

    for(i = 0; i < m; i++){
        for(j = 0; j < n; j++){
            // To check diagonally up, decrease row and increase column
            if(checkMatch(mat, str, i, j, m, n, len, -1, 1)){
                printf("FOUND AT (%d, %d) DIAG UP\n", i, j);
                count++;
            }
            // To check flat, keep row constant and increase column
            if(checkMatch(mat, str, i, j, m, n, len, 0, 1)){
                printf("FOUND AT (%d, %d) FLAT\n", i, j);
                count++;
            }
            // To check diagonally down, increase both row and column
            if(checkMatch(mat, str, i, j, m, n, len, 1, 1)){
                printf("FOUND AT (%d, %d) DIAG DOWN\n", i, j);
                count++;
            }
        }
    }
    if(count == 0)
        printf("NOT FOUND");
    return 0;
}
```