DSA:ASSIGNMENT-02

-MOULYA M

-24UG00447

CODE:

#include <stdio.h>

#include <string.h>

#define MAX 50

void searchWord(char grid[MAX][MAX], int m, int n, char word[]) {

int len = strlen(word);

int found = 0;

// Horizontal search (left to right)

for (int i = 0; i < m; i++) {

for (int j = 0; j <= n - len; j++) {

int k;

for (k = 0; k < len; k++) {

if (grid[i][j + k] != word[k])

break;

}

if (k == len) {

printf("Start: (%d, %d) End: (%d, %d)\n", i, j, i, j + len - 1);

found = 1;

}

}

}

// Vertical search (top to bottom)

for (int i = 0; i <= m - len; i++) {

for (int j = 0; j < n; j++) {

int k;

for (k = 0; k < len; k++) {

if (grid[i + k][j] != word[k])

break;

}

if (k == len) {

printf("Start: (%d, %d) End: (%d, %d)\n", i, j, i + len - 1, j);

found = 1;

}

}

}

if (!found)

printf("Word not found\n");

}

int main() {

int m, n;

char grid[MAX][MAX];

char word[MAX];

printf("Enter number of rows: ");

scanf("%d", &m);

printf("Enter number of columns: ");

scanf("%d", &n);

printf("Enter the grid (each row of uppercase letters):\n");

for (int i = 0; i < m; i++)

scanf("%s", grid[i]);

printf("Enter the target word: ");

scanf("%s", word);

printf("\nSearching for word '%s'...\n", word);

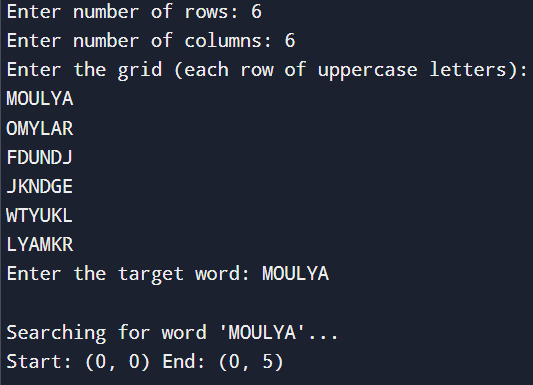
searchWord(grid, m, n, word);

return 0;

}

OUTPUT:

1)HORIZONTAL SEARCH



2)VERTICAL SEARCH

