

# CHANAKYA UNIVERSITY

SUBMITTED BY :-

NAME :- KUSHAL .A

SUBJECT :- DATA STRUCTURES AND ALGOTITHM

SECTION :-“D”

## CODE :-

#include <stdio.h>

#include <string.h>

int main() {

int m, n;

printf("Enter number of rows and columns: ");

scanf("%d %d", &m, &n);

char grid[m][n + 1]; // +1 for null terminator

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

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

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

}

char word[50];

printf("Enter the target word: ");

scanf("%s", word);

int wordLen = strlen(word);

int found = 0;

// Search horizontally (left to right)

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

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

int k;

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

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

break;

}

if (k == wordLen) {

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

found = 1;

}

}

}

// Search vertically (top to bottom)

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

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

int k;

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

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

break;

}

if (k == wordLen) {

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

found = 1;

}

}

}

if (!found)

printf("Word not found\n");

return 0;

}

OUTPUT :-

