CHANAKYA UNIVERSITY

DSA assignment – 2

Name:- D. Ganesh kumar

Reg no:- 24UG00507

Section:- D EE(VLSI)

Question:- You are required to model a 2D grid of characters as a graph, where each cell acts as a node, and edges exist between horizontally and vertically adjacent cells (no diagonal connections). Given a target word, your program should search for occurrences of the word in the grid horizontally (left to right) or vertically (top to bottom). For each occurrence of the word found, print the start node and end node coordinates. If there are no occurrences, print Word not found.

Input Format:-

• An integer m number of rows in the grid. • An integer n number of columns in the grid.

• A grid of m rows and n columns containing uppercase letters (A-Z).

• A target word (string) to be searched in the grid. Output Format

• Start: (row_start, col_start) End: (row_end, col_end) • If no occurrence is found, print "Word not found" Additional Requirement

• Design your own grid in such a way that it contains your name (in uppercase) as the target word, arranged horizontally and vertically.

• Use that grid and your name as input to test your program.

• Make sure at least 2 valid occurrences exist. If the length of your name string is too long, you can take first 5 letters of your name and demonstrate the output.

Instructions for submission:

1. Push the code file to your GitHub repo which you have shared for the evaluation.

2. Create a pdf file that contains your name, USN and screenshots of all the outputs. Upload the file to same repo. 3. EVALUATION WILL BE DONE 1-1

4. Submit the assignments on GitHub by 23-10-2025.

Solution:-

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

int main() {
    int m, n;
    printf("Enter number of rows (m): ");
    if (scanf("%d", &m) != 1) {
        fprintf(stderr, "Invalid input for m\n");
        return 1;
    }
    printf("Enter number of columns (n): ");
    if (scanf("%d", &n) != 1) {
        fprintf(stderr, "Invalid input for n\n");
        return 1;
    }

    char **grid = malloc(m * sizeof(char *));
    for (int i = 0; i < m; i++) {
        grid[i] = malloc((n + 1) * sizeof(char));
    }

    printf("Enter the grid rows (each row %d uppercase letters):\n", n);
    for (int i = 0; i < m; i++) {
        scanf("%s", grid[i]);
        if ((int)strlen(grid[i]) != n) {
            fprintf(stderr, "Row %d length is not %d\n", i, n);
            return 1;
        }
    }

    char word[101];
    printf("Enter the target word (uppercase): ");
    scanf("%s", word);
    int L = strlen(word);

    int found = 0;
    for (int i = 0; i < m; i++) {
        for (int j = 0; j + L - 1 < n; j++) {
            int k;
            for (k = 0; k < L; k++) {
                if (grid[i][j + k] != word[k]) break;
            }
            if (k == L) {
                printf("Start: (%d, %d) End: (%d, %d)\n", i, j, i, j + L - 1);
                found++;
            }
        }
    }


    for (int i = 0; i + L - 1 < m; i++) {
        for (int j = 0; j < n; j++) {
            int k;
            for (k = 0; k < L; k++) {
                if (grid[i + k][j] != word[k]) break;
            }
```

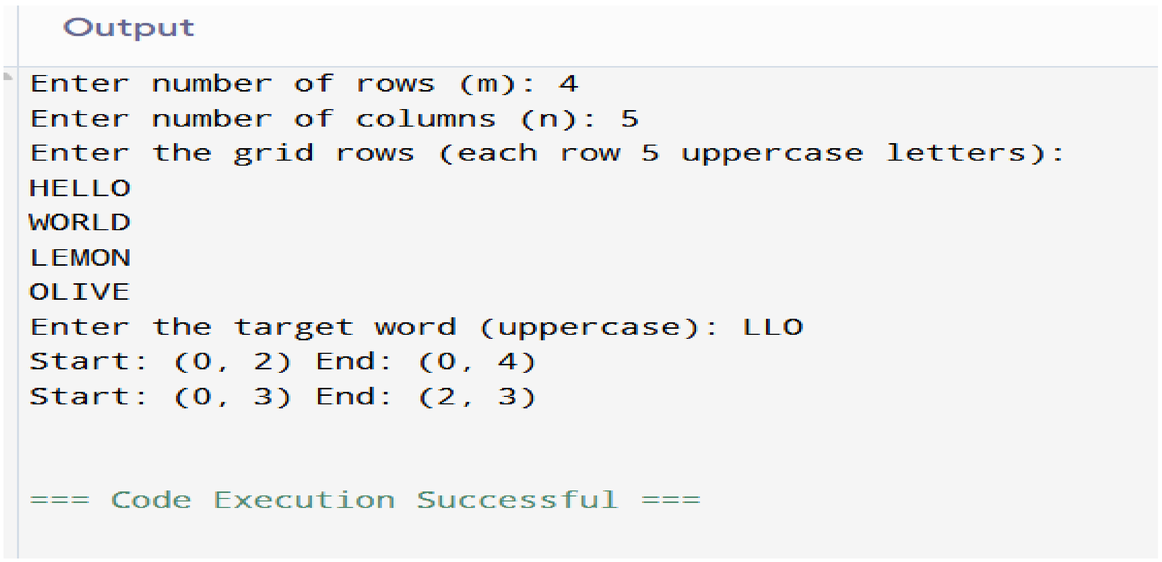
```
            if (k == L) {
                printf("Start: (%d, %d) End: (%d, %d)\n", i, j, i + L - 1, j);
                found++;
            }
        }
    }

    if (!found) {
        printf("Word not found\n");
    }

    for (int i = 0; i < m; i++) {
        free(grid[i]);
    }
    free(grid);

    return 0;
}
```

Out put:-



```
Output
Enter number of rows (m): 4
Enter number of columns (n): 5
Enter the grid rows (each row 5 uppercase letters):
HELLO
WORLD
LEMON
OLIVE
Enter the target word (uppercase): LLO
Start: (0, 2) End: (0, 4)
Start: (0, 3) End: (2, 3)


=== Code Execution Successful ===
```