**Program name:**

Word Search

**Program requirements:**

Given a 2D board and a word, find if the word exists in the grid. The word can be constructed from letters of sequentially adjacent cell, where “adjacent" cells are those horizontally or vertically neighbouring.

Note: The same letter cell may not be used more than once.

**Requirements analyse:**

Need to design an algorithm to search each character of input in the grid sequentially.

Locate the first character in the grid and then search the next character recursively by using the same function.

In addition, because the same letter cell may not be used more than once, we need to create another grid (shadownFlag grid in program)to store flags that indicate if one letter cell is used.

**Key functions designed:**

1. static boolean initialSearch(char[][] charGrid, String inputString) { }

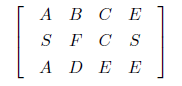
In this function the shadownFlag grid is initialized and location of the first letter can be found. If the first letter can’t be found in grid then this function returns false. If the first letter is in grid, the function for recursive search will be called.

1. static boolean searchInGrid(char[][] charGrid, int i,int j,String inputString,boolean[][] shadowFlag){}

This function is used for recursive search by comparing the current target letter with four elements in grid around the location of previous target letter. The recursion ends when the last letter of input is found or the target letter can’t be found around.

**Program test:**

2D Matrix given:



Given a word: ASADFBCCEESE 🡨 The whole matrix is searched

Result: True

Given a word: ABCB 🡨 B is searched twice

Result: False

Given a single letter: A

Result: True

