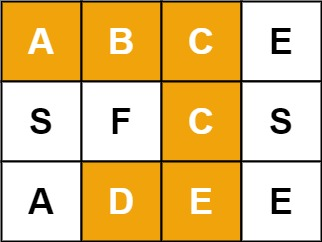
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **National University of Computer and Emerging Sciences, Lahore Campus** | | | | |
| C:\Users\saif\AppData\Local\Microsoft\Windows\Temporary Internet Files\Content.Word\final design.jpg | **Course:** | **Artificial Intelligence Lab** | **Course Code:** |  |
| **Program:** | **BS (Computer Science)** | **Semester:** | **Spring 2023** |
| **Duration:** | **40 MINUTES** | **Total Marks:** | **20** |
| **Paper Date:** | **18-Feb-23** | **Weight** | **5 %** |
| **Section:** | **B1-B2** | **Page(s):** | **2** |
| **Exam:** | **Quiz 1** | **Roll No.** |  |
| **Instruction/Notes:** | * Submit the solution on Google Classroom. * Make sure your submitted file is not corrupted. * In case of ambiguity, take suitable assumptions. * Indentation and commenting holds marks. * There should be no memory leaks or dangling pointers. * Plagiarism will result in severe consequences. | | | |

**Question Marks: 20**

Given an m x n grid of characters board and a string word, return true *if* word *exists in the grid*.

The word can be constructed from letters of sequentially adjacent cells, where adjacent cells are horizontally or vertically neighboring. The same letter cell may not be used more than once.

**For example:**



**Input:** board = [["A","B","C","E"],["S","F","C","S"],["A","D","E","E"]],

**word** = "ABCCED"

**Output:** true

**Constraints:**

* m == board.length
* n = board[i].length
* 1 <= m, n <= 6
* 1 <= word.length <= 15
* board and word consists of only lowercase and uppercase English letters

The input should be prompted from the user.