Official Challenge: The Great Menu Catastrophe

The Situation

Welcome to "The Daily Grind," the official (and entirely fictional) coffee shop of DEV CHALLENGE 2025. This morning, our lead barista, the energetic but tragically clumsy Barista Barry, was preparing for the morning rush. His task was simple: take the beautiful, vintage letter tiles and arrange them on the giant magnetic menu board to display today's specials.

He'd just placed the last letter when a rogue Roomba startled him. Barry yelped, tripped over a stray power cord, and sent the entire box of remaining letters flying. The result? A chaotic N x M grid of pure, unadulterated gibberish on the menu board.

The doors open in two hours. There's no time to rearrange the letters. Manager Mel is panicking, the espresso machine is hissing impatiently, and the first caffeine-deprived customers are on their way.

Your Mission

We need a new plan, and we need it fast. Instead of fixing the board, we're going to embrace the chaos. Your mission, should you choose to accept it, is to become a "Menu Archaeologist."

You must write a program that can scan the jumbled menu grid and find all the hidden coffee-related words. To guide the desperate customers, your program must identify not only *which* items are available but also *where* they are on the board so Barista Barry can quickly circle them with a marker.

You are the developers' last hope. Don't let them face the morning without their coffee.

The Inputs (The Evidence)

You will be provided with two essential text files:

1. **grid.txt**: This file represents the chaotic menu board. It's a grid of N rows and M columns of uppercase letters. Each line in the file is a row of the grid.

Example grid.txt (the actual grid will be larger):

```
C A K E S Y
O E T T A L
F H I M E P
F O A A O E
E B C H H I
E C L F Q C
```

2. **dictionary.txt**: This is your word list, containing all possible coffee shop-related words that *might* be on today's menu. It's a simple text file with one word per line, all in uppercase. Your program should only search for words present in this dictionary.

Example dictionary.txt:

```
CAKE
CHAI
COFFEE
FOAM
LATTE
```

The Rules of the Find

A word from the dictionary may appear more than once in the grid. Your program should identify every occurrence for the challenge levels you attempt.

Core Challenge: The Basic Scan

Find all words from the dictionary that appear in their **forward** direction:

- **Horizontally (Left-to-Right):** e.g., C-A-K-E
- **Vertically (Top-to-Bottom):** e.g., C-O-F-F-E-E

Bonus Challenge #1: The Backwards Scan

Impress us by also finding words that appear in **reverse**:

- **Reverse Horizontal (Right-to-Left):** e.g., E-T-T-A-L finds LATTE
- **Reverse Vertical (Bottom-to-Top)**

Bonus Challenge #2: The Diagonal Deep-Dive

For maximum kudos, find all words that appear along the **diagonal axes**, in both **forwards and backwards** directions:

- **All four diagonal directions** are valid (e.g., top-left to bottom-right, bottom-right to top-left, etc.).
- Example: I-A-H-C (from bottom-right to top-left) finds CHAI

The Desired Output (The Solution Map)

Your program must produce a single plain text file named **solution.txt**.

Each line in this file must represent one word found in the grid. The format for each line is the word, followed by its starting and ending coordinates.

Format:
<WORD> <start_x>,<start_y> <end_x>,<end_y>

Important Coordinate System Details:

- The coordinate system is **0-indexed**.
- The **x-coordinate** represents the **column number** (from left, starting at 0).
- The **y-coordinate** represents the **row number** (from top, starting at 0).
- <start_x>,<start_y> are the coordinates of the **first letter** of the word.
- <end_x>,<end_y> are the coordinates of the **last letter** of the word.

Example solution.txt (based on the sample grid.txt):

CAKE 0,0 3,0

COFFEE 0,0 0,5

LATTE 4,1 0,1

CHAI 5,5 2,2

(Note: In the example above, CAKE and COFFEE would be found in the Core Challenge. LATTE would be found in Bonus Challenge #1. CHAI would be found in Bonus Challenge #2. The order of lines in your final text file does not matter.)

Final Words

Barista Barry is counting on you. The reputation of "The Daily Grind" is in your hands.

May your code compile quickly and your coffee be ever-strong. Good luck, developers.