

Game

[English](#)

[Vietnamese](#)

Note: C, S, B are used instead of H, F, I, respectively, in test case.

Hal and Dave are playing an interesting game on a rectangular chess-like board consisting of squares arranged in R rows and C columns. Here are the rules of the game:

- There is only one piece on a board alternately moved by the players.
- One move of the piece consists of moving it to an adjacent square in one of following directions: down, right or diagonally right–down.
- Some squares on a board are ‘forbidden’, i.e. the piece cannot enter such squares.
- A square may contain at most one of the following: a hamburger, french fries, ice cream. A player who moves the piece to a square with a hamburger receives 1 point, with french fries 3 points and with ice cream 5 points.
- The game ends when a player who should make a move cannot make it (because piece would fall off of board or enter a forbidden square in all three directions).
- If at the end of a game both players have same number of points, then the one who cannot make a legal move lost that game.
- If at the end of a game players have different numbers of points, then the player with more points won the game.
- Both players have 0 points at the beginning of a game. Hal makes first move. The initial position of the piece is a square that is not forbidden and it does not contain any food.

Since there is a finite number of sequence of moves finishing any given game, it can be proved that for any given initial position of the piece either Dave or Hal can win no matter how the other player plays, i.e. he has a winning strategy.

A board, positions of forbidden squares, positions of squares with food and some initial positions are given. Write a program that will determine for each given initial position which player has a winning strategy.

Input

The first line of input file contains two integers R, number of rows ($2 \leq R \leq 100$) and C, number of columns ($2 \leq C \leq 100$) of a board, separated by a space character.

Each of next R lines contains a sequence of C characters representing a corresponding row of the board. Forbidden squares are represented with ‘#’ character. Squares containing food are represented with letters H (hamburger), F (french fries) and I (ice cream). Other squares are represented with a dot character (‘.’).

The next line contains an integer N, $1 \leq N \leq 100$, the number of given initial positions for which a program should find which player has a winning strategy. Each of the following N lines contain two integers A ($1 \leq A \leq R$) and B ($1 \leq B \leq C$), separated with a space character, which determine a row and a column of an initial position of a piece.

Rows are numbered from top to bottom with number from 1 to R, and columns are numbered from

left to right with numbers from 1 to C.

Output

Output file should contain N lines. The i^{th} line should contain name of a player (i.e. DAVE or HAL) having a winning strategy for i^{th} given initial position.

Sample

Input:

```
3 4
.H#
l...
##H.
3
1 1
1 4
2 3
```

Output:

```
HAL
DAVE
HAL
```

Input:

```
4 5
.#...
#.#.F
.#.F
.#...
3
3 1
3 3
1 5
```

Output:

```
HAL
HAL
HAL
```

Input:

```
5 6
##..#
..#FH#
..#..#
###...
.....l
4
2 1
5 1
1 4
1 6
```

Output:

```
HAL
HAL
DAVE
DAVE
```

