**Day 2 - Stack, Queue**

**Problem 1:** Implement Stack using Array. The Stack class should support the following methods:

int size()

boolean isEmpty()

int top()

void push(int element)

void pop()

**Problem 2:** Implement a queue using an array. The Queue class should support the following methods:

int size()

boolean isEmpty()

int front()

int back()

void push(int element)

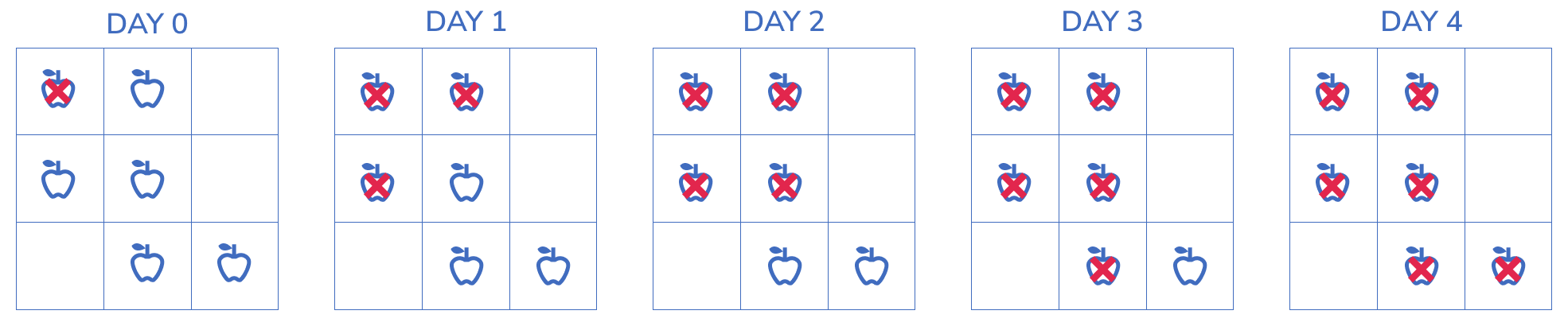
void pop()

Problem 3: You are given an n \* m grid where each position can contain one of the three values:

| **Cell Value** | **Represents** |
| --- | --- |
| 0 | Empty Cell |
| 1 | Fresh Apple |
| 2 | Rotten Apple |

Every day, all fresh apples which are adjacent to any rotten apple become rotten. Two cells are adjacent if they have a common edge, i.e., each cell can have upto four adjacent cells.

Find the minimum number of days required for all the apples to be rotten. If this is not possible return -1.

**Testing**

**Input Format**

The first line contains an integer ‘T’ denoting the number of independent test cases.

For each test case the input has the following lines:

* The first line contains two space separated numbers, ‘n’ and ‘m’, denoting the number of rows and columns in the grid respectively.
* The next n lines contain m elements each, denoting the elements of the grid.

**Output Format**

For each test case, a line containing the minimum number of days for all the apples to be rotten.

If it is impossible, the value is -1.

**Sample Input**

2

3 3

2 1 0

1 1 0

0 1 1

3 3

2 1 1

1 1 0

1 0 1

**Expected Output**

4

-1

**Constraints**

1 <= T <= 100

1 <= n, m <= 200

0 <= cell value <= 2

—--------------------------- END —---------------------------