Old Watch

When you were younger, you had a *digital* watch you were very fond of. It had 4 modes you selected with a button that could:

- tell the time in the 24hours format (`0000` to `2359`)
- tell the temperature of your suroundings from `000` to `100`
- tell the current month of the year (`01` to `12`)
- tell the current day of the week (`1` to `7`).

One day while cleaning your room, you find this watch and wonder whether it still works. The mode the watch is in can be identified by counting the number of digits that are visible **from the right** (leading spaces are important).

- 4 digits means it's in `24hr` mode
- 3 digits means it's in `Temperature` mode
- 2 digits means it's in `Month` mode
- 1 digit means it's in `Day` mode

From the Information given, it's obvious that there are some values that woudn't make sense in each mode(ex : `2567` isn't a valid time for `24hr` even though there are 4 digits). In all such cases, we consider the watch to be in `Broken` mode.

if there are gaps between the digits, it is also considered to be in `Broken` mode(ex: `0 10` is `Broken`).

If the screen shows anything other than a digit or a space, the watch is considered to be in `Malfunctioning` mode. (*when a malfunnctioning character is there, it doesn't matter if the digits and spaces make sense or not in any other mode. It will be considered malfunctioning)

If the screen shows nothing at all, the watch is considered to be in `Dead` mode.

Identify the mode the watch is in for every test case and print it out.

Input Format

One line containing t, the number of testcases

followed by t lines containing 4 characters each (a character can be a digit from 0 to 9, a space or any other character).

Constraints

$1 < t \leq 100000$

testcases can have leading spaces(see sample testcase).

Output Format

t lines each Containing the mode of the given test case

Sample Input 0

```
6
1247
 070
  13
   2
 .10
 1 1
```

Sample Output 0

```
24hr
Temperature
Broken
Day
Malfunctioning
Broken
```

Explanation 0

`'1247'` digits fits in the 24 hour clock
`' 070'` is a value in the allowed temperature range
`' 13'` isn't a valid month
`' 2'` is a valid day
`' .10'` `.` isn't a digit
`' 1 1'` there can be no gaps

Submissions: 0
Max Score: 100
Difficulty: Medium

Rate This Challenge:

Download problem statement

Download all test cases

Suggest Edits

Collapse

Admin Options

Edit Challenge

View Submissions

C++14

```cpp
#include <cmath>
#include <cstdio>
#include <vector>
#include <iostream>
#include <algorithm>
using namespace std;


int main() {
    /* Enter your code here. Read input from STDIN. Print output to STDOUT */
    return 0;
}

```

Line: 1 Col: 1

Upload Code as File Test against custom input Run Code Submit Code