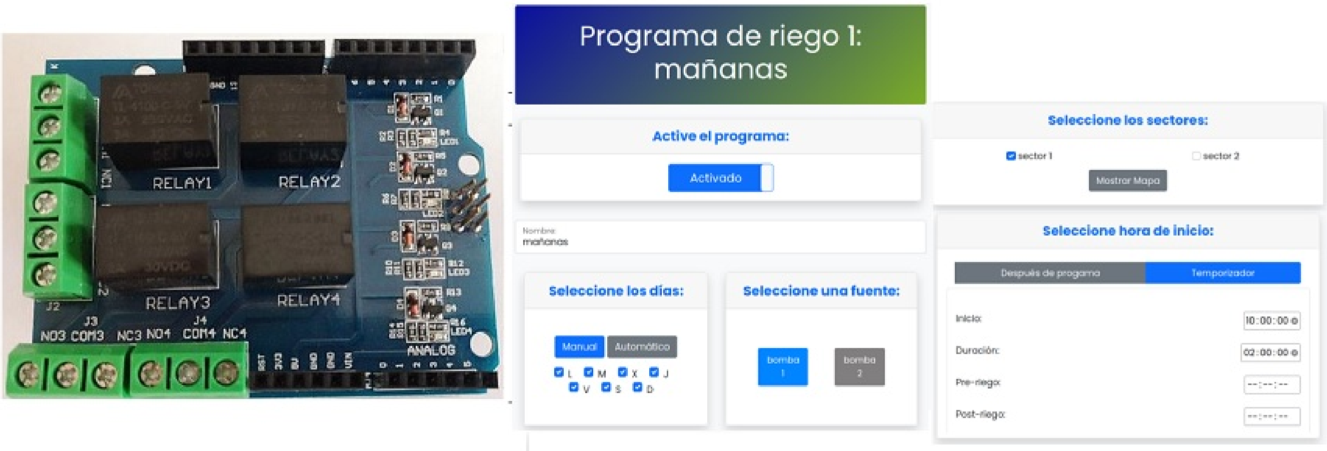
A - Irrigation Programs

Context

We are experiencing a time of little rain and great scarcity of water. But the field of Murcia is enormously technified and the crops are irrigated with optimized and highly efficient systems that give the plants the exact amount of water they need at all times.

These systems are controlled by irrigation programmers, which are electronic devices that receive irrigation programs and transform them into opening and closing commands for the valves.



Nombre: mañanas

☑ sector 1 □ sector 2

☑ L ☑ M ☑ X ☑ J ☑ V ☑ S ☑ D

Inicio: 10:00:00

Duración: 02:00:00

Pre-riego: --:--:--

Post-riego: --:--:--

A sample irrigation microcontroller and a web-based programmer. Taken from the TFG of Carmen María M.P.

The Problem

Given some irrigation programs, your task is to transform them into a series of open/close commands for the valves.

An irrigation program is defined by two values: opening time and closing time. A time is given by a day of the week (Monday, Tuesday...), hour (from 0 to 23) and minute (from 0 to 59). For example, the following irrigation program:

```
L 00:05 L 00:10
```

means: start irrigating every Monday at 00:05, and finish at 00:10. More precisely, during minute 00:05 the valve will be open, and it will close at the beginning of minute 00:10; so, *during* minute 0:10 it will be closed. This is transformed into the following sequence of close (0) / open (1) commands, from minute to minute, starting from minute 00:00 of Monday:

```
00000111110000000000000000...
```

The closing time will always be later than the opening time. The shortest program will be of 1 minute, and it could be for example:

```
S 12:00 S 12:01
```

This means irrigating for 1 minute, all Saturdays at minute 12:00.

Two or more programs can be overlapped, for example:

```
L 00:05 L 00:15
L 00:10 L 00:20
```

In these cases, the valve will be open at a certain minute if it is contained in any of the programs. Thus, the sequence of commands from minute 00:00 of Monday would be:

```
000001111111111111110000000...
```

The same sequence would be obtained, for example, for the programs:

```
L 00:05 L 00:10
L 00:10 L 00:15
L 00:15 L 00:20
```

Finally, a program can start one day and finish some days later, for example:

```
L 20:00 X 19:00
```

which means: start irrigating on Mondays at 20:00, and finish on Wednesdays at 19:00. However, an irrigation program cannot expand from Sundays at 23:59 to Mondays at 00:00. For example, a program cannot begin on Saturdays and finish on Tuesdays.

The Input

The input consists of a series of irrigation programs. The first line of the input contains two integer numbers: *N M*. *N* is the number of irrigation programs that we have to process; it can be from 1 to 1000, inclusive. And *M* is the number of minutes that we want to obtain in the sequence of commands, always staring from minute 00:00 of Monday; it can be from 1 to 10080, inclusive (10080 is the total number of minutes of a week = 7*24*60).

Then, we have *N* lines, where each line contains a program. Each program has two times, and each time has: day of the week, hour (from 0 to 23) and minute (from 0 to 59). The day of the week is a letter: L, M, X, J, V, S, D, for Monday, Tuesday, Wednesday, Thursday, Friday, Saturday and Sunday, respectively. The hour:minute can be written with or without preceding zeros, that is, both 1:2 and 01:02 are valid.

The Output

In the output, you have to write a single line (finishing with a "\n"), with *M* characters. This line contains the close (0) / open (1) commands of the valves for the given programs, always starting from minute 00:00 of Monday. You have to output these 0/1 characters without any separators between them.

Sample Input

```
7 30
M 13:22 M 17:00
L 0:23 L 0:24
L 00:05 L 00:10
L 00:10 L 00:15
L 00:15 L 00:20
L 0:11 L 0:18
L 16:55 D 23:59
```

Sample Output

```
000001111111111111110001000000
```

OMP'22
Facultad de Informática
Universidad de Murcia