
yaksha

December 21, 2019

1 MULTIPLES

You are given some magic numbers and their substitutes. Your task is to print numbers from 1 to N according to these rules:

1. If the number is *not* a multiple of any of the given magic numbers, print the *number*.
2. Otherwise, for *every* magic number the number is a multiple of, print the *substitute*.

1.1 INPUT

The first line is N (< 1000). The second line M (< 100) is the number of magic numbers. The next M lines are the magic numbers and their substitutes separated by a space.

1.2 OUTPUT

The numbers 1 to N as per the question.

1.3 EXAMPLE

Input:

```
6
2
2 Idli
3 Sambar
```

Output:

```
1
Idli
Sambar
Idli
5
IdliSambar
```

Explanation: 6 is N (line 1). The number of magic numbers are 2 (line 2). The output is the numbers 1 to N, one per each line. But, since 2, 4 are multiples of 2, *Idli* is printed instead. Similarly *Sambar* instead of 3. 5 is not a multiple of any of the magic numbers 2 and 3, so it is printed as is. Since 6 is a multiple of both 2 and 3, *IdliSambar* is printed.

2 FLIPBOOK

A flip book is a book with a series of pictures that vary gradually from one page to the next, so that when the pages are turned rapidly, the pictures appear to animate by simulating motion or some other change.

Your task is to design a language for describing flipbooks and to implement a compiler to convert these flipbook descriptions into a printable format, say a pdf file.



2.1 EXAMPLE

Let's consider a very minimal flipbook language called **F**, which has a primitive to display an image on a range of pages. An example flipbook in **F** could look like this:

```
// displays the 'baby.jpg' image on pages 1 to 40 at
// the position (0,0) in the reference system
01 40 baby.jpg position (0,0)
11 40 toddler.jpg position (10,0)
21 40 adult.jpg position (20, 0)
31 40 old_person.jpg (30, 0)
```

To generate a printable pdf of this flipbook:

```
gff life_cycle.flip -o life_cycle.pdf
```

where **gff** is a compiler of this language (i.e, a program which can read the flip book language files and produce a pdf)

A more sophisticated language will probably let you move the image over a page range, scale, rotate etc.

2.2 HINTS

1. Think about the primitives that are needed for defining a flipbook, how they can be combined, how they can be abstracted etc.
2. You do not need to define a formal specification for the language. Also, assume that the syntax can be parsed into whatever data structure you choose to use.
3. Use a simple data structure for a page in the flipbook. You can assume that in whatever language you choose to write the implementation, there are primitives to convert this page data structure into a page in a pdf.