

## Problem 3. Backup

Input file:           input.txt  
Output file:         output.txt  
Time limit:          2 seconds  
Memory limit:       256 MB



There are many different backup strategies. The easiest to understand are the differential and incremental backups.

With incremental backup, we create a copy of the data which were changed from the moment of the last backup: for instance, after a full backup on Sunday, on Monday we copy the data that were changed during Monday; the same goes for Tuesday, etc. With such strategy, the volume of data copied is relatively small, but it may require a lot of files to restore the data. For instance, to restore data on Friday, we would need backup copies from Sunday (full backup), Monday, Tuesday, Wednesday and Thursday.

With differential backup, we regularly create a full copy — for instance, every Sunday. Next, on each day of the week we make a copy of all data modified after the last full backup: a Monday copy on Monday, a Monday and Tuesday copy on Tuesday, etc. The total volume of the backed up data is relatively large, but you only need two files to restore data: the last differential and the last full backup.

Dump(1), a popular backup freeware, operates with the concept of backup level, which can perform incremental or differential backup as well as more complex strategies, which cannot be reduced to the two former methods. The idea of backup levels as such is pretty simple: when we make an  $N$  level backup, we copy all files modified from the moment of the last backup with the level lower than  $N$ . If there are no preceding lower-level backups, we make a complete backup copy of all data.

Write a program to figure out which backup files should be used for recovery based on the backup schedule.

### Input

The first line of the input file contains the number  $M$  – the total number of tests in the file ( $0 < M \leq 100\,000$ ).

Each following line of the input file consists of eight space-separated integers. The first seven numbers are schedule of backups, i.e. the levels of backup  $N_i$ , which are performed on the corresponding days of the week  $i$ : on Sunday, Monday, Tuesday, etc. ( $0 \leq N_i \leq 9$ ). It is guaranteed that the zero-level backup is performed on Sunday, i.e. the first number is zero.

The eighth number is the day  $d$  when data must be recovered ( $0 \leq d \leq 6$ ). Note that 0 is Sunday and 6 is Saturday.

### Output

For each line of the output file, print a sequence of numbers  $d_j$  corresponding to the days when the backup copies necessary for recovery were made ( $0 \leq d_j \leq 6$ ). Note that the earliest copy is always recovered first, followed by a later copy, hence the following must hold true:  $d_j < d_{j+1}$  and  $N_{d_j} < N_{d_{j+1}}$ . Moreover, the recovery always ends with the latest copy, so the last number in the line must equal  $d$ .

### Examples

| input.txt       | output.txt |
|-----------------|------------|
| 4               | 4          |
| 0 0 0 0 0 0 0 4 | 0 3        |
| 0 8 8 8 8 8 8 3 | 0 1 2 3 4  |
| 0 2 3 4 5 6 7 4 | 0 2 4      |
| 0 7 2 6 3 5 4 4 |            |