Musical Scales

Problem ID: musicalscal
CPU Time limit: 1 secon
Memory limit: 1024 MB
Difficulty: 1.6

Author: Zachary Friggsta
Source: Alberta Collegiat
Programming Contest 20
License: (cc) BY-SA

The following are musical notes in “increasing order”:

A, A#, B, C, C#, D, D#, E, F, F#, G, G#

The difference between consecutive notes is a *semitone*, and the sequence wraps around so the note that is one semitone above G# is A. The difference between a *tone* is the same as two semitones. So the note that is one tone above B is C#. The note that is one tone above G is A.

We do not worry about flats such as Cb nor do we worry about adding a # sign to B and E in this problem (they are aliases for notes that are already listed).

A major scale is defined by a note (such as A or C#) and all other notes following that one in an arithmetic progression:

tone, *tone*, *semitone*, *tone*, *tone*, *tone*, *semitone*

The starting note appears in the name of the scale.

For example, the scale A#-major consists of the following notes:

A#, C, D, D#, F, G, A, A#

(by convention, the first note is repeated at the end of the sequence)

Finally, in this problem a song is just a sequence of notes. Your job is to identify all major scales such that the song uses only notes in that scale.

Input

The first line of input is an integer $1 \leq n \leq 100$ denoting the number of notes played in a song. The second line consists of a sequence of notes, separated by spaces.

Output

Output consists of a single line that lists all scales the song may be played in. Consecutive scales should be separated by a single space and the scales must appear in lexicographic order. If the song may not fit in any one of these scales, simply output a line containing the text `none`.

Sample Input 1

```
10
C D F D C D F F F C
```

Sample Output 1

```
A# C D# F
```

Sample Input 2

```
6
A B A F# G# C
```

Sample Output 2

```
none
```