Secret Agent Key Generation and Decryption

Description

You are a secret agent tasked with implementing a secure key generation and decryption system. Your mission is to create two main functions:

1. `key_gen()` : Generates a legal key array based on a unique number (timestamp).
2. `decipher()` : Accepts a scrambled code and deciphers it to obtain the original password.

As a secret agent, your job is to change the legal scramble key code every 1 minute. The system should produce a different key for each minute, ensuring that no key is repeated until all possible permutations have been used.

In order to avoid too many combinations, the numbers '8' and '9' will not appear in the key of this question.

Initial Conditions

- Initial key: "6 1 3 4 0 7 2 5"
- Password: 4225624

Input

The input consists of a series of timestamps (in seconds), separated by a single space.

The input timestamps will always be less than 8! * 60 seconds (40320 * 60 =2419200seconds) to ensure all permutations can be covered.

Number of input timestamps < 10^5

Output

For each timestamp, output the following information:

- Time in minutes (format: yy.yy)
  - The output time in minutes should be rounded down to the second decimal place.
- The corresponding key (if more than 1 minute has passed since the last key change)
  - To ensure randomness, each key generated for a new minute must be unique, which means Keys should not repeat until all possible permutations have been used.
  - You can invent any method to scramble the key array, as long as it is legal and produces randomly different results with respect to the timestamp.
- The scrambled code
- The deciphered password
  - The `decipher()` function should always be able to recover the original password from the scrambled code.
- Each line of output corresponds to one input timestamp, fields on each line are separated by a single space.

Sample Input 1 📋

```
30 65 75 100
```

Sample Output 1

```
0.50 6 1 3 4 0 7 2 5 0337230 4225624 //no more 1 min
1.08 6 7 5 4 3 2 1 0 3552153 4225624 //new key
1.25 6 7 5 4 3 2 1 0 3552153 4225624 //same as previou
s
1.66 6 7 5 4 3 2 1 0 3552153 4225624 //You can output
any permutation as long as it is legal
```

Hint

Ensure that your `key_gen()` function produces unique keys for each minute until all permutations are exhausted.

Problems

Announcements

Submissions

Rankings

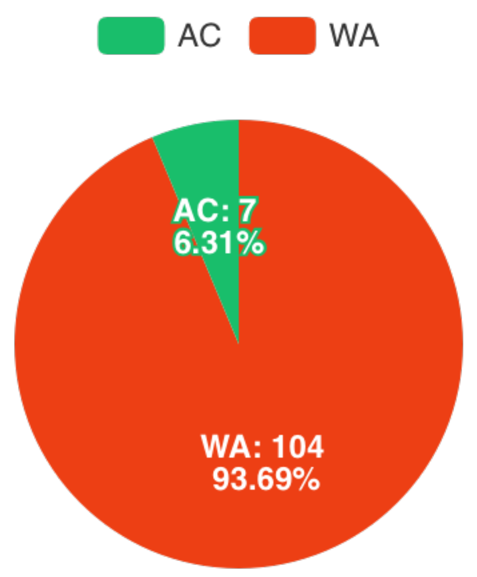
View Contest

Information

ID	mid_12
Time Limit	10000MS
Memory Limit	256MB
IO Mode	Standard IO
Created By	sungod
Level	Low
Score	100
Tags	Show

Statistic

Details


🟩 AC
🟥 WA
AC: 7
WA: 104