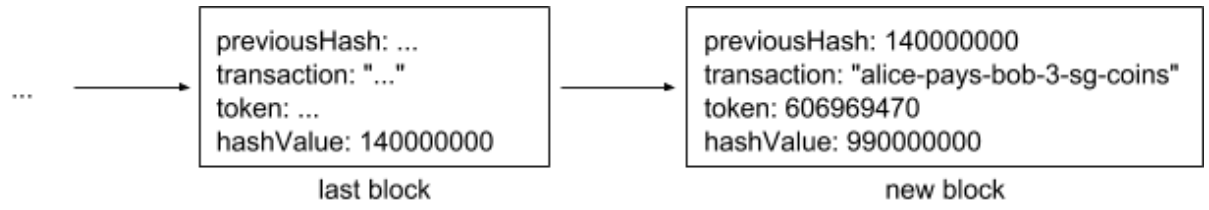
Problem C. SG Coin

Time limit	1000 ms
Mem limit	1048576 kB
OS	Linux

There is a new cryptocurrency called SG coin. Alice purchases an item from Bob and pays in SG coin. The transaction is recorded in a blockchain. A blockchain is a chain of blocks where each block contains the hash value of the previous block, a transaction string, a token, and the hash value of the block (a positive integer with 7 trailing zeros).

Example: Suppose the hash value of the last block in the blockchain is 140 000 000 and the next transaction string to be recorded is "`alice-pays-bob-3-sg-coins`". The new block containing the transaction is appended (chained) to the last block in the blockchain like this:



The token in the new block must be set to a number between 0 to $10^9 - 1$ such that the hashValue of the new block is positive and has 7 trailing zeros. If the token is 606 969 470 then the hashValue of the new block is computed as follow:

```
H(140000000,"alice-pays-bob-3-sg-coins",606969470)=990000000
```

The function `H` is given below (in C++ and Java format only). There can be many other valid values for the token that produces a valid hashValue (with 7 trailing zeros), e.g.,

```
H(140000000,"alice-pays-bob-3-sg-coins",306969470)=690000000
```

In a distributed system, a new block can be added to the blockchain by anyone and then broadcast to everyone. Everyone will receive the new block and verify that it is valid (i.e., the last block's hashValue, transaction string, and token of the new block produces a hashValue with 7 trailing zeros). Everyone keeps track of the longest chain and only appends a new block to the longest chain that they have.

Charlie has a lot of computing power and he sees an opportunity to attack the distributed blockchain system by generating new blocks rapidly, thus producing the longest chain quickly. This allows him to monopolize new transactions to be appended to the blockchain.

Let's see an attack scenario. Alice pays Bob 3 SG coins for purchasing an item, she puts the transaction in the blockchain and broadcasts. Bob sees the transaction in the blockchain and then gives the purchased item to Alice. Charlie then comes in to disrupt this transaction by creating two consecutive new blocks and broadcasting. Everyone sees the two new blocks from Charlie and keeps them, discarding the new block from Alice (which is shorter).

You are Charlie and you want to make life harder for Alice and Bob.

The hash function in C++:

```
long long H(long long previousHash, string &transaction,
            long long token) {
  long long v = previousHash;
  for (int i = 0; i < transaction.length(); i++) {
    v = (v * 31 + transaction[i]) % 1000000007;
  }
  return (v * 7 + token) % 1000000007;
}
```

The hash function in Java:

```
static long H(long previousHash, String transaction,
              long token) {
  long v = previousHash;
  for (int i = 0; i < transaction.length(); i++) {
    v = (v * 31 + transaction.charAt(i)) % 1000000007;
  }
  return (v * 7 + token) % 1000000007;
}
```

Input

The first line is the hashValue of the last block in the blockchain. The hashValue is between 0 to 1 000 000 006 inclusive and guaranteed to have 7 trailing zeros.

Output

Produce two new blocks A and B as fast as possible. A should be chained to the last block (from the given input) and B should be chained to A. Print the transaction strings and the tokens of the two new blocks.

The first line contains the transaction string and the token for A separated by a space and the second line is for B. The transaction string is non-empty and can be anything you want containing only lowercase English characters, digits (0–9), or hyphen with at most 100 characters. The token must be an integer between 0 to $10^9 - 1$.

Explanation

The last three blocks have hashValues of: $140\,000\,000 \rightarrow 930\,000\,000 \rightarrow 730\,000\,000$.

Sample 1

Input	Output
140000000	charlie-pays-to-eve-9-sg-coins 218216710 icpc-sg-2018-at-nus 620658977