Movie Police (MP) is an international top secret law enforcement agency, controlling illegal movie downloads on internet. With their elite team of programmers, MP has developed a very smart algorithm to produce movie signatures. A *movie signature* is a binary string, one bit for every frame in the movie, so that the i-th bit in the signature corresponds to the i-th frame in the movie. The algorithm is so amazing that it outputs consistently the same signature for versions of the same film with different quality resolutions. One application of this revolutionary technology uses it to detect if a small clip is part of a movie, looking for a high similarity between the clip signature and the movie signature.

Now MP has started to apply this technology and, as a first step, a massive online database of movie signatures was already built. As a new member of the MP crew, you must write a program that, given the signature of a clip, finds the index in the MP database of a movie whose signature *matches* the clip signature at most. That is, a movie whose signature has a substring, of the same length of the clip signature, that is most *similar* to the clip signature.

Similarity between strings of the same length is defined by means of their *Hamming distance* (number of bits that do not match), so that "more similar" means "less Hamming distance".

Input

The first line of the input contains two positive integer numbers M and Q, separated by a blank, where M indicates the number of movie signatures in the database and Q indicates the number of clip signatures to process ($1 \leq M \leq 1000$, $1 \leq Q \leq 500$). Each one of the following M lines contains a binary string s_i describing the i-th movie signature in the database. You may suppose that s_i has length l_i, where $1 \leq l_i \leq 100$. Finally, there are Q lines, each one with a binary string that corresponds to a clip signature to search for maximal similarity in the database. You may assume that, for every clip signature to be searched, there is at least one movie signature in the database whose length is greater or equal than the clip's length.

Output

For each clip signature given in the input, output a single line with the lowest index i of a movie s_i ($1 \leq i \leq M$) that matches the clip at most, as above explained. If there are two movie signatures that match the clip signature maximally, answer the one with lower index in the database.

Sample Input

```
3 1
000011
1101111111
1111100000
1000111
```

Sample Output

```
2
```