

2690 Decoding Task

In the near future any research and publications about cryptography are outlawed throughout the world on the grounds of national security concerns. The reasoning for this is clear and widely accepted by all governments - if cryptography literature is public like in the old times, then everybody (even criminals and terrorists) could easily use it to hide their malicious plans from the national and international security forces. Consequently, public cryptographic algorithms and systems have ceased to exist, and everybody who needs strong protection for their secrets is forced to invent proprietary algorithms.

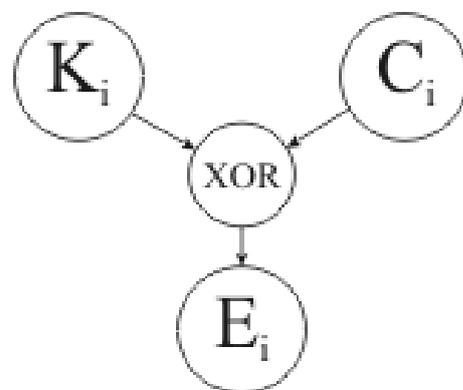
The ACM Corporation has lots of competitors who are eager to learn its trade secrets. Moreover, the job to protect their secrets is complicated by the fact, that they are forced to use intercontinental communication lines which are easy to eavesdrop on, unlike internal lines of the ACM Corporation which are well guarded. Therefore, the ACM Corporation have invented the Intercontinental Cryptographic Protection Code (ICPC) which they are very proud of, and which is considered unbreakable - nobody has even tried to break it yet, but that is about to change.

The group of hackers was hired by the rival company, which does not disclose its name to them, to break ICPC. As the first step, they have bribed one of the programmers who implemented the software for ICPC and have learned how ICPC works. It turns out, the ICPC uses very long key which is a sequence of bytes generated by some sophisticated and random physical process. This key is changed weekly and is used to encrypt all messages that are sent over intercontinental communication lines during the week. This programmer has also proudly told them, that ICPC is the fastest code in the world, because (having the benefit of highly sophisticated code generation) they simply perform bitwise exclusive OR (XOR) operation between the bytes of the message and the key. That is, the i -th byte of the encrypted message $E_i = K_i \text{ XOR } C_i$, where K_i is the i -th byte of the key and C_i is the i -th byte of the original clear-text message.

Having learned how ICPC works, they have started to look for the way to reliably obtain the key every week, which is the only thing that is still missing to listen for all intercontinental communications of the ACM Corporation (eavesdropping on the intercontinental lines themselves has indeed turned out to be an easy task). An attempt to bribe the security officers who guard and distribute the key has failed, because the security officers (having the profession with one the highest salaries of that time) have turned out to be too expensive to bribe.

During the search for alternative solutions, they have stumbled upon a clerk, who sends weekly newsletters to various employees and departments. Fortunately, these newsletters are being sent just after the change of the key and the messages are usually long enough to recover sufficient portions of the key by studying original newsletters and their encoded forms. However, they could not covertly find anyone who will disclose the newsletter contents on a weekly basis, because all the employees were bound by a Non-Disclosure Agreement (NDA) and the penalty for the disclosure of any corporate message according to this NDA is death.

Yet they were able to convince this clerk (for a small reward) to do a seemingly innocent thing. That is, while sending the copies of newsletter throughout the corporation, he was instructed to insert an extra space character in the beginning of some messages but send other copies in their original form.



Now the task to recover the key is straightforward and it is you, who shall create a program for this. The program is given two ICPCed messages where the first message is N bytes, and the second one is $N + 1$ bytes and is the result of encoding the same clear-text messages as the first one, but with one extra space character (represented by the byte with the decimal value of 32) in the beginning. The program shall find the first $N + 1$ bytes of the key that was used to encode the messages.

Input

Input file contains several test cases. Each of them consists of two lines. The first line consists of $2N$ characters and represents the encoded message N bytes long. The second line consists of $2N + 2$ characters and represents the encoded message $N + 1$ bytes long. Here $1 \leq N \leq 10000$. Each message is written on a single line in a hexadecimal form byte by byte without spaces. Each byte of the message is represented by two characters '0'-'9', 'A'-'F' that represent the hexadecimal value of the corresponding byte.

Output

For each input case write a single different output line, that represents $N + 1$ bytes of the recovered key in the same hexadecimal format as in the input file.

Sample Input

```
05262C5269143F314C2A69651A264B
610728413B63072C52222169720B425E
```

Sample Output

```
41434D2049435043204E454552432732
```