

Problem A

Atlantic Email

Time limit: 1 second

The year is 1984. Your boss, Professor Werner Zorn, has been pushing for establishing connections to international networks. His newest project is for Karlsruhe University¹ to receive the first email from the US Computer Science Network (CSNet).

The email will be sent by Laura Breeden (Massachusetts Institute of Technology) to the Karlsruhe University computers. For simplicity, the email only consists of n lowercase letters. It should be encoded so it can be sent across 5 transatlantic binary channels.



CC BY-SA 4.0 by Reinhard Ferdinand on Wikimedia Commons

You want the communication to be done efficiently, so you require that across each channel, no more than $n + 10$ bits are sent. The channels are working flawlessly and losslessly, so for each of the 5 channels, the bits will arrive intact in Karlsruhe.

Everything was almost ready, when you noticed that the wires coming into the computer in Karlsruhe are a mess, and so you do not know which of the 5 channels is which. Thus, as the receiver, you will receive the binary strings of the 5 channels in arbitrary order. Instead of untangling the mess, you think it is a better idea to just fix it in software. Design a communication protocol that can transfer the email correctly and efficiently.

Input

The input consists of:

- One line with the action that your program needs to perform: either the string “send” or “receive”, which denotes whether you are at the sending or receiving end.
- If the action is “send”:
 - One line with one integer n ($1 \leq n \leq 10\,000$), the length of the email.
 - One line with a string s of length n , the email to send. The email only consists of English lowercase letters (a–z).
- If the action is “receive”:
 - Five lines, each with a string b ($1 \leq |b| \leq n + 10$), only consisting of digits 0 and 1, the five binary strings you used to encode the email, in an arbitrary order.

Note that when receiving, you have to infer n from the received binary strings, as this is not known to you up front.

This is a multi-pass problem. Your program will be invoked multiple times, possibly more than twice. It is guaranteed that the first pass is a “send” action, and that each subsequent pass is a “receive” action.

For testing purposes, the number and input of subsequent passes will depend on the output of your submission.

A testing tool is provided to help you develop your solution.

¹Former university in Karlsruhe that later merged to become Karlsruhe Institute of Technology.

Output

If the action is “send”, output 5 bit strings b_1, \dots, b_5 ($1 \leq |b_i| \leq n + 10$ for each i), only consisting of digits 0 and 1, the bit strings to send.

If the action is “receive”, output the original email s , a string only consisting of English lowercase letters (a–z).

Sample Case 1

Sample Input	Pass 1	Sample Output
send 17 willkommenimcsnet	0 01 111111 1001 1	

Sample Input	Pass 2	Sample Output
receive 1 01 1001 0 111111	willkommenimcsnet	

Sample Case 2

Sample Input	Pass 1	Sample Output
send 3 kit	000000000 111111111 000000000 111111111 000000000	

Sample Input	Pass 2	Sample Output
receive 111111111 000000000 000000000 000000000 111111111	kit	