# Vanya and secret number

| | |
|---|---|
| Input file: | `standard input` |
| Output file: | `standard output` |
| Time limit: | 1 second |
| Memory limit: | 512 megabytes |

Vanya and Jack are playing a game. Jack provides Vanya with 2 streams of numbers and Vanya has to guess the secret code formed by the two streams of numbers. The secret code is a sequence of the elements of the two streams. The secret code starts with the first element of first sequence, then the last element of the second sequence, then second element of sequence 1, then second last element of sequence 2 and so on till integers are left in any of the sequences. (See sample cases for reference) If $L_1$, $L_2$, $L_3$ . . . .. $L_n$ are the numbers in the first stream and $M_1$, $M_2$, $M_3$ . . . .. $M_m$ are the numbers in the second stream then the secret code is formed as follows: $L_1$, $M_n$, $L_2$, $M_{n-1}$, $L_3$, $M_{n-2}$. . . . . .

Calculate the secret code formed by the two streams.

## Input

There are two lines of input: ($1 \leq$ *number of elements in any stream* $\leq 10^6$) The first line contains first input stream containing space seperated integers, -1 at the end marks the end of the stream. The second line contains second input stream containing space seperated integers, -1 at the end marks the end of the stream.

## Output

Print space separated numbers, the elements of the secret code

## Examples

| standard input | standard output |
|---|---|
| 1 2 3 4 5 -1<br>9 8 7 6 -1 | 1 6 2 7 3 8 4 9 5 |
| 1 2 3 4 5 6 7 -1<br>11 12 13 -1 | 1 13 2 12 3 11 4 5 6 7 |
| 1 2 -1<br>10 11 12 13 14 15 -1 | 1 15 2 14 13 12 11 10 |

## Note

Use of *ArrayList* data structure is not allowed. Do not forget to use Fast I/O for Java. (Reader Class) All elements would be positive integers, |element| $\leq 10^9$