

PROB4: COMPRESS

**PROB4: COMPRESS**

Reduce the number of digits.

**DESCRIPTION**

An experimental physicist generates a great deal of data from experiments that he performs. The data generated from these experiments has a special property, and he wants to take advantage of this property to reduce the amount of space needed to store the results.

The data is generated in pairs of numbers, where the first number is always less than the second number. The way that the physicist wants to store these numbers is similar to the how some people abbreviate a range of numbers in a book. For example, when they refer to pages 11 through 18 in a book, they will sometimes denote it as 11–8.

Some notation definitions:

Definitions	For example
The first of a pair of numbers is denoted by F.	in "18482–02", F = 18482
The second of a pair of numbers (in compressed form) is denoted by C.	in "18482–02", C = 02
The second of a pair of numbers (in decoded form) is denoted by R.	in "18482–02", R = 18502
MSD(x,y) refers to the 'x' most significant digits of 'y' when 'y' is denoted in base ten, which is the null string for $x \leq 0$	MSD(3,19283) = 192, MSD(0,12)=""
LSD(x,y) refers to the 'x' least significant digits of 'y' when 'y' is denoted in base ten (possibly padded with zeros).	LSD(2, 48290) = 90, LSD(2,3)= 03

The rules for decoding the "compressed" second number are as follows:

Rule	An example
The number C is always written with the fewest possible digits.	
If the number C is larger than F, then R is the same as C.	given "123–283", then F=123, C=283, and R would be 283
If C is less than or equal to F, then the following rules apply:	
LSD(length(C), R) will always be the same as C.	
If LSD(length(C), F) is less than C, then R is equal to MSD(length(F) – length(C), F), prepended to the digits of C.	given "4137–223", then: F=4137, C=223: LSD(length(C),R) = 223 MSD(4 – 3, 4137) = 4

## PROB4: COMPRESS

	R would be 4223
<p>If <math>LSD(\text{length}(C), F)</math> is greater than or equal to <math>C</math>, then <math>R</math> is equal to <math>10^{\text{length}(C)}</math> added to the following value: <math>MSD(\text{length}(F) - \text{length}(C), F)</math>, prepended to the digits of <math>C</math>.</p>	<p>given: "8543-13", then  <math>F=8543, C=13</math>:  <math>LSD(\text{length}(C), R) = 13</math>  <math>MSD(4 - 2, 8543) = 85</math>  <math>10^2 = 100</math>  <math>R</math> would be <math>8513 + 100 = 8613</math></p>

Please note that leading zeros on the number  $C$  are significant. '7' is not the same as '07', and neither of them are the same as '007'. For example:

given: "2839-06", then  $F=2839, C=06$  so  $R$  would be 2906

given: "2839-006", then  $F=2839, C=006$  so  $R$  would be 3006

Your task is to compute the "compressed" second number format from its uncompressed version.

### INPUT: prob4.dat

In the input file, each line of input will consist of a pair of non-negative integers separated by a hyphen, where the second number is always larger than the first number. The second number will always be less than  $2^{31}-1$ .

An example input file would be

```

column  11111111112
12345678901234567890
line 1:10-18[EOL]
      2:83294-84137[EOL]
      3:100-200[EOL]
      :[EOF]
```

### OUTPUT: prob4.out

Other than the standard header and trailer messages, each line of input will produce one line of output. Each line of output will consist of the first number, followed by a hyphen, followed by the "compressed" version of the second number.

The correct output corresponding to the example input file above would be

```

column  11111111112222222223
123456789012345678901234567890
line 1:Program 4 by team 0[EOL]
      2:10-8[EOL]
      3:83294-137[EOL]
      4:100-00[EOL]
      5:End of program 4 by team 0[EOL]
      :[EOF]
```