C. Abracadabra

time limit per test: 1 second

memory limit per test: 256 megabytes

input: standard input

output: standard output

Polycarpus analyzes a string called *abracadabra*. This string is constructed using the following algorithm:

- On the first step the string consists of a single character "a".
- On the k-th step Polycarpus concatenates two copies of the string obtained on the $(k - 1)$-th step, while inserting the k-th character of the alphabet between them. Polycarpus uses the alphabet that consists of lowercase Latin letters and digits (a total of 36 characters). The alphabet characters are numbered like this: the 1-st character is "a", the 2-nd — "b", ..., the 26-th — "z", the 27-th — "0", the 28-th — "1", ..., the 36-th — "9".

Let's have a closer look at the algorithm. On the second step Polycarpus will concatenate two strings "a" and insert the character "b" between them, resulting in "aba" string. The third step will transform it into "abacaba", and the fourth one - into "abacabadabacaba". Thus, the string constructed on the k-th step will consist of $2^k - 1$ characters.

Polycarpus wrote down the string he got after 30 steps of the given algorithm and chose two non-empty substrings of it. Your task is to find the length of the longest common substring of the two substrings selected by Polycarpus.

A substring $s[i... j]$ $(1 \le i \le j \le |s|)$ of string s = $s_1 s_2... s_{|s|}$ is a string $s_i s_{i+1}... s_j$. For example, substring $s[2...4]$ of string s = "abacaba" equals "bac". The string is its own substring.

The longest common substring of two strings s and t is the longest string that is a substring of both s and t. For example, the longest common substring of "contest" and "systemtesting" is string "test". There can be several common substrings of maximum length.

Input

The input consists of a single line containing four integers l_1, r_1, l_2, r_2 $(1 \le l_i \le r_i \le 10^9, i = 1, 2)$. The numbers are separated by single spaces. l_i and r_i give the indices of the first and the last characters of the i-th chosen substring, correspondingly $(i = 1, 2)$. The characters of string *abracadabra* are numbered starting from 1.

Output

Print a single number — the length of the longest common substring of the given strings. If there are no common substrings, print 0.

Examples

input

```
3 6 1 4
```

output

```
2
```

input

```
1 1 4 4
```

output

```
0
```

Note

In the first sample the first substring is "acab", the second one is "abac". These two substrings have two longest common substrings "ac" and "ab", but we are only interested in their length — 2.

In the second sample the first substring is "a", the second one is "c". These two substrings don't have any common characters, so the length of their longest common substring is 0.