

First Occurrence

Two non-negative integers A and B are given. Integer A occurs in integer B at position P if the decimal representation of A is a substring, starting at position P (counting from zero), of the decimal representation of B. Decimal representations are assumed to be big-endian and without leading zeros (the only exception being the number 0, whose decimal representation is “0”).

Examples:

- 53 occurs in 1953786 at position 2.
- 78 occurs in 195378678 at positions 4 and 7.
- 57 does not occur in 153786.

Write a C# function -

```
public class Solution
{
    public int GetIntegerDecimalOccurrence(int a, int b);
}
```

That, given two non-negative integers A and B, returns the leftmost position at which A occurs in B. The function should return -1 if A does not occur in B.

For example, given A = 53 and B = 1953786, the function should return 2, as explained above.

Requirements

- A is an integer within the range [0..999,999,999]
- B is an integer within the range [0..999,999,999]

Notes

1. When implementing the function, implement it at first in the simplest way possible using .NET 4.5 Standard libraries.
2. If time permits, please provide an alternative implementation to your first solution using C# primitives.