Problem Statement

The digit sum of an integer is the sum of its digits in decimal notation. For example, the digit sum of 1234 is 1+2+3+4=10, and the digit sum of 3443 is 3+4+4+3=14.

You are given three integers: **A**, **B** and **C**. Return the integer X between **A** and **B**, inclusive, such that the absolute difference between the digit sum of X and the digit sum of **C** is as small as possible. If there are multiple possible values for X, return the smallest among them.

Definition

Class:	MinimalDifference
Method:	findNumber
Parameters:	int, int, int
Returns:	int
Method signature:	int findNumber(int A, int B, int C)
(be sure your method is public)	

Constraints

- **A**, **B** and **C** will each be between 1 and 1,000,000,000, inclusive.
- **B**-**A** will be between 0 and 100,000, inclusive.

Examples

0)

```
1
9
10
Returns: 1
```

The digit sum of C is 1+0=1. Of the integers between 1 and 9, inclusive, only 1 has a digit sum equal to 1.

1)

```
11
20
20
Returns: 11
```

The digit sum of C is 2+0=2. Of the integers between 11 and 20, inclusive, there are two which also have a digit sum of 2: 11 (1+1=2) and 20 (2+0=2). We choose 11 because it is smaller than 20.

2)

```
1
1
999
Returns: 1
```

3)

```
100
1000
99
Returns: 189
```

4)

```
1987
9123
1
Returns: 2000
```