Problem Statement

We have an arithmetic expression made up of positive integers, the + and - operators and parentheses. All the parentheses, however, have been erased by the cleaning staff and we want to calculate the minimum value the original expression may have had.

You will be given a String **e** containing the expression without the parentheses. Return the minimum value the original expression could have had before the parentheses were erased.

Definition

Class: LostParentheses
Method: minResult
Parameters: String
Returns: int
Method signature: int minResult(String e)
(be sure your method is public)

Notes

- All operations in the original expression were addition and subtraction; there were no parentheses placed between two consecutive digits.

Constraints

- **e** will contain between 1 and 50 characters, inclusive.
- Each character of **e** will be a digit ('0'-'9'), a plus sign ('+') or a minus sign ('-').
- The first and last characters of **e** will be digits.
- No two operators (characters '+' and '-') will appear consecutively in **e**.
- There will not be a sequence of more than 5 consecutive digits in **e**.

Examples

0)

```
"55-50+40"
Returns: -35
```

The expression can only have two different values: 55-50+40=45 and 55-(50+40)=-35.

1)

```
"10+20+30+40"
Returns: 100
```

Since the sum is associative, any parenthesization of the expression would get the same result.

2)

```
"00009-00009"
Returns: 0
```

Numbers may contain leading zeroes.