[**Expression Add Operators**](https://leetcode.com/problems/expression-add-operators/)

**import** java.util.ArrayList;

**import** java.util.List;

**public** **class** ExpressionAddOperators {

**public** **static** **void** main(String[] args) {

// **TODO** Auto-generated method stub

System.***out***.println(*addOperators*("123" , 6));

}

**public** **static** List<String> addOperators(String num, **int** target) {

List<String> result = **new** ArrayList<>();

**if**(num == **null** || num.length() == 0) {

**return** result;

}

*helper*(result , num , target , "" , 0 , 0 , 0);

**return** result;

}

**public** **static** **void** helper(List<String> result, String num , **int** target , String curr , **long** value , **long** prev , **int** start) {

**if**(start == num.length()) {

**if**(target == value) {

result.add(curr);

}

**return**;

}

**for**(**int** i = start ; i < num.length() ; i++) {

**long** d = Long.*parseLong*(num.substring(start, i + 1));

**if**(i != start && num.charAt(i) == '0') {

**break**;

}

**if**(start == 0) {

*helper*(result, num , target , d+"" , d , d , i + 1);

}

**else** {

*helper*(result, num , target , curr + "+" +d , value + d , d , i + 1);

*helper*(result, num , target , curr + "-" +d , value - d , -d , i + 1);

*helper*(result, num , target , curr + "\*" +d , value - prev + (prev \* d) , prev \* d , i + 1);

}

}

}

}

Time Complexity : O(n \* (4 ^ n)) where n is length of given string

Space Complexity : O(n) where n is length of given string