[CS 11] Prac 1n – Expression Evaluation

oj.dcs.upd.edu.ph/problem/cs11prac1n

Problem Statement

A "polynomial" is actually just an object abstracting anything that can be obtained from some variables using addition, subtraction, multiplication and constants only. For example, we know that an expression such as

(x+x·x+5)(x−3)−x $(x + x \cdot x + 5) (x - 3) - x$

will expand to a polynomial, that is, an expression that looks like

a0+a1x+a2x2+…+anxn$a_0 + a_1 x + a_2 x^2 + \ldots + a_n x^n$,

even without actually doing the expansion!

More formally, we can prove the following theorem:

An expression is a polynomial in some variables iff it can be obtained from those variables using only addition, subtraction, multiplication, and constants.

The proof is straightforward. (Exercise for the reader!)

For this problem, we will deal with a special case of such expressions:

- Only addition, multiplication, integer constants between 00 and 99, and the variable xx is allowed.
- There are no parentheses.
- All multiplication operations will be marked with the operator `*`.

Given such an expression as well as the value of xx (an integer), evaluate the expression.

Task Details

Your task is to implement a function called `evaluate`. This function has two parameters `e` and `x`:

- `e` is a `str` representing the expression. No space characters will appear in the input.

- `x` is an `int` denoting the value of the variable x.

The function must return an `int` denoting the evaluation of `e` with the given value of x.

Restrictions

For this problem:

- Assignment is allowed.
- Recursion is allowed.
- Up to 66 function definitions are allowed.
- Comprehensions are **disallowed**.
- `range` is **disallowed**.
- The `abs` symbol is now allowed.
- The source code limit is 10001000.

Example Calls

Example 1 Function Call

Copy

```
evaluate('x*x+6*x+9', 3)
```

Example 1 Return Value

Copy

```
36
```

Example 2 Function Call

Copy

```
evaluate('x*x+6*x+9', -3)
```

Example 2 Return Value

Copy

```
0
```

Example 3 Function Call

Copy

```
evaluate('3+1+4+1+5', 31415)
```

Example 3 Return Value

Copy

```
14
```

Example 4 Function Call

Copy

```
evaluate('x+x', 20)
```

Example 4 Return Value

Copy

```
40
```

Constraints

- The function `evaluate` will be called at most 3030 times.
- The expression is at most 5050 characters long and will be valid.
- $|x| \leq 1010$ | x | $\leq 10^{10}$.

Scoring

- You get 100100 ❤️ points if you solve all test cases where:
  - `+` and `*` do *not* both appear in the expression.
- You get 100100 ❤️ points if you solve all test cases.

Report an issue

Clarifications

No clarifications have been made at this time.