

Вычислитель

Требуется реализовать приложение, принимающее на вход список инструкций.

Существуют два вида инструкций:

1. **calc**: вычислить арифметическую операцию (умножение, сложение, вычитание) над двумя сущностями и сохранить результат в переменную. Сущность может быть либо литерал типа int64, либо имя переменной. В одну и ту же переменную записывать результат можно только один раз.
2. **print**: вывести значение указанной переменной, например, **x = 5**
Считаем, что каждая арифметическая операция из инструкций вычисляется долго, например 50ms. Требуется наиболее быстро вычислять результаты списка инструкций, т.е. выводить требуемые переменные.

Примеры

1

input

```
[
  { "type": "calc", "op": "+", "var": "x", "left": 1, "right": 2 },
  { "type": "print", "var": "x" }
]
```

output

```
{ "items": [
  { "var": "x", "value": 3 }
]
```

2

input

```
[
  { "type": "calc", "op": "+", "var": "x", "left": 10, "right": 2 },
  { "type": "print", "var": "x" },
  { "type": "calc", "op": "-", "var": "y", "left": "x", "right": 3 },
  { "type": "calc", "op": "*", "var": "z", "left": "x", "right": "y" },
  { "type": "print", "var": "w" },
  { "type": "calc", "op": "*", "var": "w", "left": "z", "right": 0 }
]
```

output

```
{ "items": [
  { "var": "x", "value": 12 },
  { "var": "w", "value": 0 }
]
```

3

input

```
[
  { "type": "calc", "op": "+", "var": "x",      "left": 10, "right": 2 },
  { "type": "calc", "op": "*", "var": "y",      "left": "x", "right": 5 },
  { "type": "calc", "op": "-", "var": "q",      "left": "y", "right": 20 },
  { "type": "calc", "op": "+", "var": "unusedA", "left": "y", "right": 100 },
  { "type": "calc", "op": "*", "var": "unusedB", "left": "unusedA", "right": 2 },
  { "type": "print", "var": "q",                },
  { "type": "calc", "op": "-", "var": "z",      "left": "x", "right": 15 },
  { "type": "print", "var": "z",                },
  { "type": "calc", "op": "+", "var": "ignoreC", "left": "z", "right": "y" },
  { "type": "print", "var": "x",                }
]
```

output

```
{ "items": [
  {"var": "q", "value": 40},
  {"var": "z", "value": -3},
  {"var": "x", "value": 12}
]
```

Требования

Реализовать в виде сервиса, который обязательно:

- 1) написан на go;
- 2) любой паттерн проектирования;
- 3) поднимается через docker compose;
- 4) unit tests coverage > 20%;
- 5) реализует grpc и http endpoint, есть swagger;
- 6) примеры запросов и ответов;
- 7) ссылка на gh, смотрю на master, или архив с кодом;
- 8) краткая записка—документация по реализации, до двух A4, в свободной форме. pdf, md, txt;