A. Shuting yard with min/max

time limit per test: 5 seconds
memory limit per test: 256 megabytes

You are given mathematical expressions specified in infix notation with two prefix functions (**min** and **max**). Your task is to convert it to a postfix notation (also known as Reverse Polish notation) using shunting yard algorithm.

Your implementation **must** include Stack ADT (as an interface or an abstract class) and its implementation.

Input

The single line of input contains correct mathematical expression. The expression contains only

- single-digit decimal number (e.g. `0`, `5`, `9`),
- subtraction (`-`), division (`/`), multiplication (`*`), addition (`+`) operators,
- left and right parentheses,
- maximum and minimum functions with two arguments: `max ( <arg1> , <arg2> )` and `min ( <arg1> , <arg2> )`

All tokens are separated by spaces.

Output

Print converted expression. All tokens must be separated by spaces.

Examples

input Скопировать

```
1 + 2 * min ( 3 , 5 ) - 4 / 2
```

output Скопировать

```
1 2 3 5 min * + 4 2 / -
```

input Скопировать

```
1 + 2 + 3 + 4 + 5
```

output Скопировать

```
1 2 + 3 + 4 + 5 +
```

input Скопировать

```
1 * 2 / 3 * 4 / 5
```

output Скопировать

```
1 2 * 3 / 4 * 5 /
```

input Скопировать

```
3 - 2 - 1
```

output Скопировать

```
3 2 - 1 -
```

Note

All operations are *left-associative*: `1 + 2 + 3` is the same as `(1 + 2) + 3`, yielding `1 2 + 3 +` and **not** `1 2 3 + +`.