Reverse Polish Notation

Reverse Polish Notation

- Reverse Polish Notation (RPN):
  - Other way to write calculations
  - **Parenthesis-free**
  - `stack`-based implementation

Reverse Polish Notation

- Reverse Polish Notation (RPN):
  - Other way to write calculations
  - **Parenthesis-free**
  - `stack`-based implementation

- Example:
  - Classical Notation
  - RPN

```
(13 + 3) / 4
```

```
13 3 + 4 /
```

Reverse Polish Notation

- Evaluation

- **Read next symbol**
  - Case **number**: Put it to stack
  - Case **operator**:
    1. Remove two numbers from stack
    2. Treat these as operands
    3. Put result back to stack
- **Repeat** (until done)

For simplicity: all operators here have *two* operands

Reverse Polish Notation

- **Read next symbol**
  - Case **number**: Put it to stack
  - Case **operator**:
    1. Remove two numbers from stack
    2. Treat these as operands
    3. Put result back to stack
- **Repeat** (until done)

```
13 3 + 4 /
```

RPN

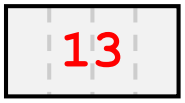
Stack

Reverse Polish Notation

- **Read next symbol**
  - Case **number**: Put it to stack
  - Case **operator**:
    1. Remove two numbers from stack
    2. Treat these as operands
    3. Put result back to stack
- **Repeat** (until done)

```
13 3 + 4 /
```

RPN

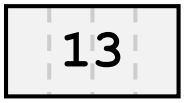


Stack

Reverse Polish Notation

- **Read next symbol**
  - Case **number**: Put it to stack
  - Case **operator**:
    1. Remove two numbers from stack
    2. Treat these as operands
    3. Put result back to stack
- **Repeat** (until done)

`3 + 4 /` **RPN**

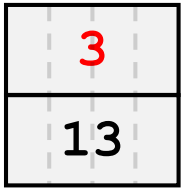


Stack

Reverse Polish Notation

- **Read next symbol**
  - Case **number**: Put it to stack
  - Case **operator**:
    1. Remove two numbers from stack
    2. Treat these as operands
    3. Put result back to stack
- **Repeat** (until done)

`3 + 4 /` **RPN**

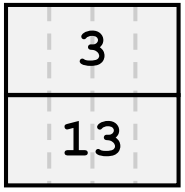


Stack

Reverse Polish Notation

- **Read next symbol**
  - Case **number**: Put it to stack
  - Case **operator**:
    1. Remove two numbers from stack
    2. Treat these as operands
    3. Put result back to stack
- **Repeat** (until done)

`+ 4 /` **RPN**

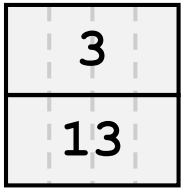


Stack

Reverse Polish Notation

- **Read next symbol**
  - Case **number**: Put it to stack
  - Case **operator**:
    1. Remove two numbers from stack
    2. Treat these as operands
    3. Put result back to stack
- **Repeat** (until done)

`+ 4 /` **RPN**



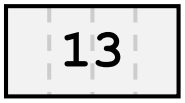
Stack

Reverse Polish Notation

- **Read next symbol**
  - Case **number**: Put it to stack
  - Case **operator**:
    1. Remove two numbers from stack
    2. Treat these as operands
    3. Put result back to stack
- **Repeat** (until done)

RPN

3



Stack

Reverse Polish Notation

- **Read next symbol**
  - Case **number**: Put it to stack
  - Case **operator**:
    1. Remove two numbers from stack
    2. Treat these as operands
    3. Put result back to stack
- **Repeat** (until done)

`+ 4 /` **RPN**

`13 3`

Stack

Reverse Polish Notation

- **Read next symbol**
  - Case **number**: Put it to stack
  - Case **operator**:
    1. Remove two numbers from stack
    2. Treat these as operands
    3. Put result back to stack
- **Repeat** (until done)

`+ 4 /` **RPN**

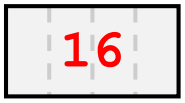
`16`

Stack

Reverse Polish Notation

- **Read next symbol**
  - Case **number**: Put it to stack
  - Case **operator**:
    1. Remove two numbers from stack
    2. Treat these as operands
    3. Put result back to stack
- **Repeat** (until done)

`+ 4 /` **RPN**

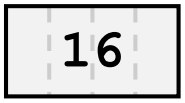


Stack

Reverse Polish Notation

- **Read next symbol**
  - Case **number**: Put it to stack
  - Case **operator**:
    1. Remove two numbers from stack
    2. Treat these as operands
    3. Put result back to stack
- **Repeat** (until done)

RPN



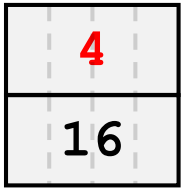
Stack

Reverse Polish Notation

- **Read next symbol**
  - Case **number**: Put it to stack
  - Case **operator**:
    1. Remove two numbers from stack
    2. Treat these as operands
    3. Put result back to stack
- **Repeat** (until done)

4 /

RPN

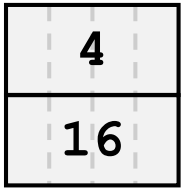


Stack

Reverse Polish Notation

- **Read next symbol**
  - Case **number**: Put it to stack
  - Case **operator**:
    1. Remove two numbers from stack
    2. Treat these as operands
    3. Put result back to stack
- **Repeat** (until done)

/ **RPN**

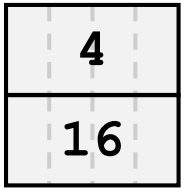


Stack

Reverse Polish Notation

- **Read next symbol**
  - Case **number**: Put it to stack
  - Case **operator**:
    1. Remove two numbers from stack
    2. Treat these as operands
    3. Put result back to stack
- **Repeat** (until done)

/ **RPN**



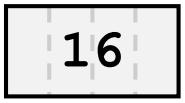
Stack

Reverse Polish Notation

- **Read next symbol**
  - Case **number**: Put it to stack
  - Case **operator**:
    1. Remove two numbers from stack
    2. Treat these as operands
    3. Put result back to stack
- **Repeat** (until done)

RPN

4



Stack

Reverse Polish Notation

- **Read next symbol**
  - Case **number**: Put it to stack
  - Case **operator**:
    1. Remove two numbers from stack
    2. Treat these as operands
    3. Put result back to stack
- **Repeat** (until done)

RPN

16 4

Stack

Reverse Polish Notation

- **Read next symbol**
  - Case **number**: Put it to stack
  - Case **operator**:
    1. Remove two numbers from stack
    2. Treat these as operands
    3. Put result back to stack
- **Repeat** (until done)

RPN

4

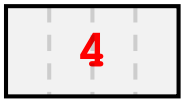
Stack

Reverse Polish Notation

- **Read next symbol**
  - Case **number**: Put it to stack
  - Case **operator**:
    1. Remove two numbers from stack
    2. Treat these as operands
    3. Put result back to stack
- **Repeat** (until done)

/

RPN

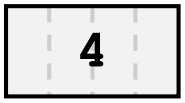


Stack

Reverse Polish Notation

- **Read next symbol**
  - Case **number**: Put it to stack
  - Case **operator**:
    1. Remove two numbers from stack
    2. Treat these as operands
    3. Put result back to stack
- **Repeat** (until done)

RPN



Stack