



AOJ

2435 - Zero Division Checker

<http://judge.u-aizu.ac.jp/onlinejudge/description.jsp?id=2435>

Problem

- 逆ポーランド記法で表された式を計算
 - 8 bit 符号なし整数 (0 ~ 255)
 - 変数は範囲指定
- 0 除算が発生する場合 error を出力

逆ポーランド記法

3 4 * 6 /



3

逆ポーランド記法

3 4 * 6 /



逆ポーランド記法

4

3 4 * 6 /

4

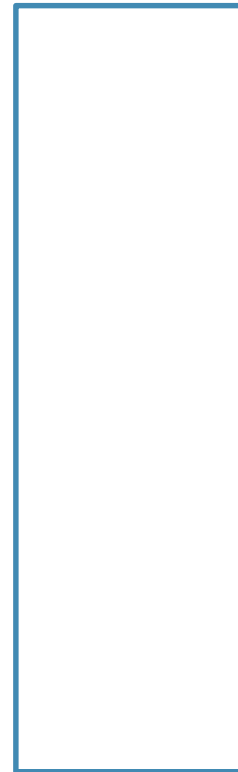
3

逆ポーランド記法

3 4 * 6 /

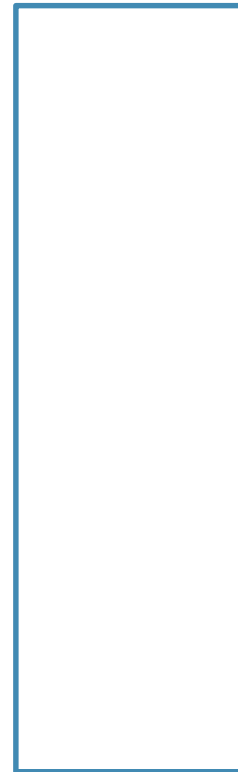
3

4



逆ポーランド記法

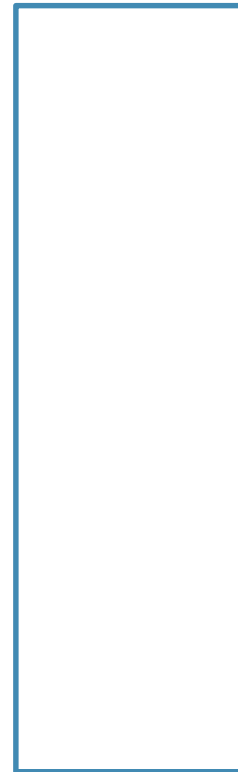
3 4 * 6 /



逆ポーランド記法

3 4 * 6 /

12



逆ポーランド記法

3 4 * 6 /

12

逆ポーランド記法

3 4 * 6 /



逆ポーランド記法

10

3 4 * 6 /

6

12

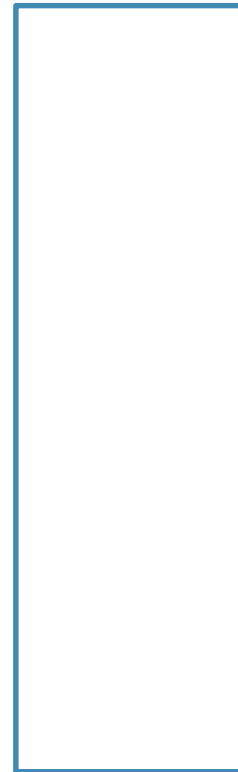
逆ポーランド記法

11

3 4 * 6 /

12

6

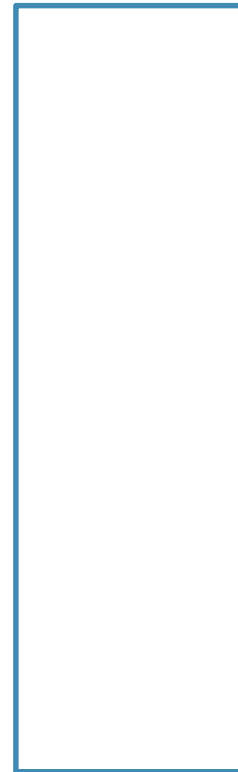


逆ポーランド記法

12

3 4 * 6 /

$$12 \div 6$$

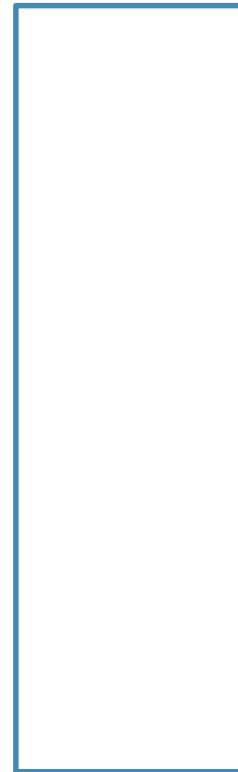


逆ポーランド記法

13

3 4 * 6 /

2



逆ポーランド記法

14

3 4 * 6 /

2

入力

```
2           // 変数は 2 個
a 1 10      // 変数 a は 1 以上 10 以下
b 1 10      // 変数 b は 1 以上 10 以下
5           // 式の要素は 5 個
1 a b - /   // 逆ポーランド記法
```


疑似コード

```
s = 空のスタック
n = 式の要素数
for i in 1..n:
    if 式のi番目の要素が整数:
        sにその整数をプッシュする
    if 式のi番目の要素が変数:
        sにその変数の値をプッシュする
    if 式のi番目の要素が演算子:
        sから値をポップし、bとする
        sから値をポップし、aとする
        if 演算子が'+':
            r = (a + b) % 256 とする
        if 演算子が'-':
            r = (a - b + 256) % 256 とする
        if 演算子が'*':
            r = (a * b) % 256 とする
        if 演算子が'/':
            r = (a / b) % 256 とする
        sにrをプッシュする
sから値をポップし、式の計算結果とする
```

疑似コード

```
s = 空のスタック
n = 式の要素数
for i in 1..n:
    if 式のi番目の要素が整数:
        sにその整数をプッシュする
    if 式のi番目の要素が変数:
        sにその変数の値をプッシュする
    if 式のi番目の要素が演算子:
        sから値をポップし、bとする
        sから値をポップし、aとする
        if 演算子が'+':
             $r = (a + b) \% 256$  とする
        if 演算子が'-':
             $r = (a - b + 256) \% 256$  とする
        if 演算子が'*':
             $r = (a * b) \% 256$  とする
        if 演算子が'/':
             $r = (a / b) \% 256$  とする
        sにrをプッシュする
sから値をポップし、式の計算結果とする
```

解法

- 取りうる値全てについて計算
 - 取りうる値は高々 256 通り

変数の扱い

$a \ b^*$

$a = \{ 1, 2 \} \quad b = \{ 2, 3 \}$



変数の扱い

20

$a \times b$

$a = \{1, 2\}$ $b = \{2, 3\}$

$a \times b$

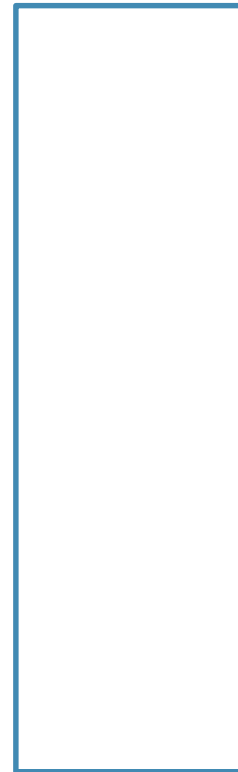


変数の扱い

a b *

$r = \{ 2, 3, 4, 6 \}$

r

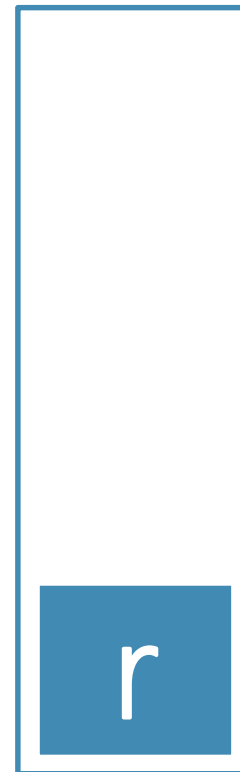


変数の扱い

22

a b *

$r = \{ 2, 3, 4, 6 \}$



コード上での変数の扱い

- 変数 を `vector<bool>` で表現
 - 要素数は 256
 - i 番目の要素が `true` \Leftrightarrow 変数は値 i を取りうる

コード上での変数の扱い

- 変数 を `vector<bool>` で表現
 - 要素数は 256
 - i 番目の要素が `true` \Leftrightarrow 変数は値 i を取りうる
- ついでに定数も `vector<bool>` で表現

コード上での変数の扱い

- 変数 を `vector<bool>` で表現
 - 要素数は 256
 - i 番目の要素が `true` \Leftrightarrow 変数は値 i を取りうる
- ついでに定数も `vector<bool>` で表現
- 演算結果も `vector<bool>` で表現

乗算の場合

```
vector<bool> a, b, r(256, false);  
b = s.top(), s.pop(); a = s.top(), s.pop();  
for(int i = 0; i < 256; i++) {  
    if(a[i] == false) continue;  
    for(int j = 0; j < 256; j++) {  
        if(b[j] == false) continue;  
        r[i * j % 256] = true;  
    }  
}  
s.push(r);
```

ソースコード

- [2435.cpp](#)