

I117 (15) 演習 補足

知念

北陸先端科学技術大学院大学 情報科学研究科
School of Information Science,
Japan Advanced Institute of Science and Technology

7/7 レポート課題: I117-15 演習 1)-2)

expreval01.c 拡張 ← expreval01b.c がより良い

1) 減算、除算、剰余に対応せよ★

$$18 - 5 \Rightarrow 13$$

$$34 / 7 \Rightarrow 4$$

$$34 \% 7 \Rightarrow 6$$

2) 括弧に対応して優先度変更を実現せよ★★

$$(1 + 2) * 3 \Rightarrow 9$$

トークン

演算子のマクロ
や値を決める。
括弧は左右ある
ので新たに演算
子が 5 つ必要

名称	マクロ	値	文字
なし	OIG	NUL	なし
加算	OPLUS	+	+
乗算	OTIMES	*	*
加算	OINC	#	++
減算	?	?	-
除算	?	?	/
剰余	?	?	%
左括弧	?	?	(
右括弧	?	?	)

挙動理解

トークンが切り出されている様子を確認しておくこと

```
% cc expreval01b.c
% echo '1+2*3' | ./a.out
; 1+2*3
token type 1 token '1' op 0 ''
token type 2 token '+' op 43 '+'
token type 1 token '2' op 0 ''
token type 2 token '*' op 42 '*'
token type 1 token '3' op 0 ''
reach to end
```

挙動理解 (*cont.*)

括弧に対応していない状況ではこうなる

```
% echo '(1+2)*3' | ./a.out
; (1+2)*3
***ignore token
token type 0 token '' op 0 ''
resolv ***unexpected tokentype 0
***ignore token
token type 0 token '' op 0 ''
0
```

出るべきメッセージを予想してから作ると効率が良い

挙動理解 (cont.)

挙動を理解できたら表示する必要はない
NDEBUG を設定してコンパイルすると、デバッグ関連のメッセージが出なくなる

```
% cc -DNDEBUG expreval01b.c  
% echo '1+2*3' | ./a.out  
; 1+2*3  
7
```

レベル

優先度に応じたレベル毎の関数 (expreval01b.c)

関数名		演算子	演算
4	resolve		token → number
3	L3	なし	resolve を呼ぶだけ
2	L2	*	乗算
1	L1	+	加算

- 求める演算と同じレベルがあればその関数に、その演算処理を追加

レベル (cont.)

- 同じレベルがなければ新しい関数を作って上下から呼ぶ
 - ◇ 減算は？
 - ◇ 除算は？
 - ◇ 剰余は？
 - ◇ 括弧は？

『優先度変更』

括弧は優先度を変更する機能を持つ
今回のレベル毎に関数を設けた場合、優先度変更とは
何を意味するのか

テスト（必須ではないが強く要請）

提出後のバグ発覚を防ぐため...

- 先に挙げた例だけでなく、各自でテスト項目を考えてテストせよ
- 各種プログラム(電卓など)であらかじめ正解を出しておく
- 以前説明したシェルやプログラム内部で式を繰り返し与えると、自動化できる

テスト（必須ではないが強く要請） *(cont.)*

テスト例:

1) $1+2*3$

2) $(1+2)*3$

3) $1+2*(3+4)$

4) $12*7\%5$

5) $132/7/5$

6) $3-13*2$

7) $123*234-345/456+567\%67$