

# コンパイラ演習：作成問題 1

プログラムの提出に関しては下記のガイドラインに従うこと。レポートは通常通り提出する。今回は、プログラムは**(2)で作成したコンパイラのソースのみを ssh.cis.k....に提出**すること。(レポートには(1)(2)の両方のソースプログラムを印刷したものを添付すること)

<http://cis.k.hosei.ac.jp/~asasaki/lectureCompiler/guideline.htm>

(0) ハンドコンパイル問題集。

疑似命令を使った hsm のコードを示せ。余力のある人は、疑似命令を使わないバージョンも示せ。(a)は疑似命令を使う必要は無い。

```
(a)
int main() {
    putint(16 * 16);
}

(b)
int main() {
    int x;
    x = 16 * 16;
    putint(x);
}

(c)
int main() {
    int x, y;
    x = 16;
    y = 8;
    putint( (x + y * 2) * 8 );
}

(d)
int main() {
    int x;
    x = 16 * 16;
    putint( x * x );
}
```

(1) 整数、四則演算、括弧からなる中置記法の式を hsm マシン語プログラムに翻訳するコンパイラを JavaCC 使って作成せよ。下記の中置記法の式をテストプログラムとして、作成したコンパイラでコンパイルし、その結果 (hsm マシン語) を hsm 仮想機械で実行して答えが一致するか確認すること。

$(4-2)*4 - ((22-12)/2 - (15-12)) + 5*(1-0) - 5 - (6 - (2+1)) * (10 - ((8-2)*2+4)/(15/3-3))$

答え : 0

$((8*(6+3))/12/3-2) + 18/(12-9)/2/(2+1) - 1$

答え : 0

$3*(10-5-21*5)/2-50$

答え : -200

$9/(5*2-(17-10))$

答え : 3

上記で、「答え」とは、hsm(仮想機械)で実行して求められた結果、すなわちスタックに残っている値のことを指す。

(2) 下記のような言語のコンパイラを作成せよ。

機能：

- 整数をオペランドとし、四則演算、単項マイナス、括弧を含む算術式を処理できる。
  - putint 文を処理できる。
- (変数機能は次回以降)

文法：

```
<PROGRAM> ::= <MAIN>
<MAIN> ::= 'int' 'main' '(' ')' <BLOCK>
<BLOCK> ::= '{' <STATEMENTLIST> '}'
<STATEMENTLIST> ::= empty
                    | <STATEMENTLIST> <STATEMENT>
<STATEMENT> ::= 'putint' '(' <EXPRESSION> ')' ';'
<EXPRESSION> ::= <TERM>
                | <EXPRESSION> '+' <TERM>
                | <EXPRESSION> '-' <TERM>
<TERM> ::= <UNARY>
          | <TERM> '*' <UNARY>
          | <TERM> '/' <UNARY>
<UNARY> ::= <FACTOR>
          | '-' <UNARY>
<FACTOR> ::= <IDENT>
            | <NUMBER>
            | '(' <EXPRESSION> ')'
```

字句の定義 (終端記号)

- 空白、タブ、改行 (これらは構文の中では使われない。スキップする)
- <NUMBER> ::= 数字の1回以上の繰り返し文字列
- 括弧記号、区切り記号など
- 演算記号
- その他キーワード (予約語) int, main, putint など  
(empty は  $\epsilon$  (空の記号列) を意味する。)

---

putint <EXPRESSION>の値を整数で出力

例：

```
int main() {
    putint(1);
    putint(-30);
    putint(9/(5*2-(17-10)));
}
```

次のような hsm コードを出力するようにせよ。

```
LDC 0 1
WRI 0 0
LDC 0 30
NEG 0 0
WRI 0 0
... ← (9/(5*2-(17-10)))を計算するコードをここに入れる (省略)
WRI 0 0
```

## HLT 0 0

WRI はスタックトップの内容を数値として表示し、NEG はスタックトップの数を反転させる `hsm` 命令である。

`hsm` で実行すると

## 1-303

と表示される。(改行を入れていないので、結果がつながって出てくる。)

ヒント: この文法は左再帰性をもった文法であり、LL 文法ではない。括り出し (繰り返しを許す書き方) あるいは左再帰除去を行え。なお、括り出しの手法がよりシンプルであるため、こちらを推奨する。