

複合初級問 1 後置表記法変換 (2005 年秋FE午後問 4 改)

問 次のプログラムの説明及びプログラムを読んで、設問 1～3 に答えよ。

[プログラムの説明]

中置表記法による正しい数式を、スタックを用いて後置表記法に変換する副プログラム toPostfix である。

- (1) 例えば、中置表記法による数式 $4 \times (9 + 3)$ は、toPostfix の処理の結果、後置表記法による数式 $4 9 3 + \times$ に変換される。
- (2) 中置表記法によるここでの数式は、1けたの数字“0”～“9”，演算子“+”，“-”，“×”，“÷”及び括弧“(”，“)”の各要素からなる。各要素は表1に示す変換の優先度をもつ。ここで、数値が大きいほど優先度が高い。プログラムでは、各要素を文字として扱い、文字型の配列 Exptext に1要素ずつ格納している。また、表1に示す要素以外に、スタックの底を示すスタック制御文字 EOS を使用し、その優先度を-1 とする。

表1 変換の優先度

数式の要素	優先度
(4
0, 1, ..., 9	3
×, ÷	2
+, -	1
)	0

- (3) 変換結果の後置表記法の数式は、文字型の配列 Postfix に1要素ずつ格納される。
- (4) toPostfix は、スタックを初期化して、Exptext の先頭の要素から順番に次の ①, ② の処理を繰り返し、Exptext の要素をすべて処理したら、スタックに残っているすべての数式の要素を順に取り出して Postfix に格納する。
 - ① Exptext[i] の優先度がスタックの先頭要素の優先度以下で、かつスタックの先頭要素が“(”でない間、スタックから要素を取り出して Postfix に格納する。
 - ② Exptext[i] が“)”でなければ、Exptext[i] をスタックに積み、“)”であれば、スタックから要素を一つだけ取り出す。
- (5) toPostfix の引数の仕様を、表2に示す。すべての配列の添字は、0 から始まる。

表2 副プログラム toPostfix の引数の仕様

変数名	型	入力/出力	意味
Exptext[]	文字型	入力	中置表記法による数式が格納されている1次元配列
Textlen	整数型	入力	中置表記法による数式の要素数
Postfix[]	文字型	出力	後置表記法に変換後の数式が格納される1次元配列
Postfixlen	整数型	出力	後置表記法に変換後の数式の要素数

- (6) toPostfix は、スタックを初期化する手続 initStack, 数式の要素をスタックに積む手続 push, スタックから取り出す関数 pop, スタックの先頭要素の値を返却する関数 top, 及び引数で指定された要素の優先度

を返す関数 `getPriority` を使用する。 `initStack`, `push`, `pop`, `top` 及び `getPriority` の仕様を表3に示す。

表3 toPostfix で使用する手続／関数の仕様

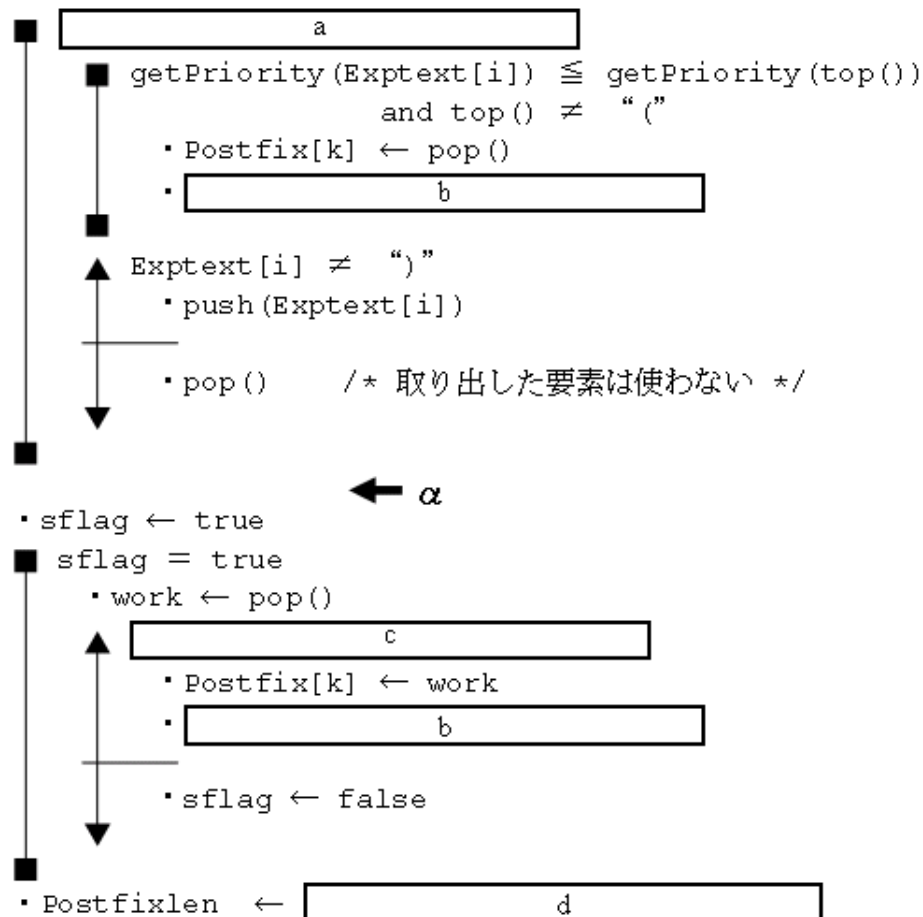
手続／関数	仕様
<code>initStack()</code>	スタックを初期化し、スタックに EOS を格納する。
<code>push(文字型: element)</code>	スタックに <code>element</code> を格納する。
文字型: <code>pop()</code>	スタックから要素を取り出し、その要素を返却する。
文字型: <code>top()</code>	スタックの先頭要素の値を返却する。
整数型: <code>getPriority</code> (文字型: <code>element</code>)	引数で指定された要素 <code>element</code> の優先度を返却する。

[プログラム]

○ `toPostfix(文字型: Exptext[], 整数型: Textlen,
 文字型: Postfix[], 整数型: Postfixlen)`

- 整数型: `k, i`
- 論理型: `sflag`
- 文字型: `work`

・ `initStack()`
 ・ `k ← 0`



設問1 プログラム中の に入れる正しい答えを記述せよ。ただし、繰り返し処理において、ある変数の値を1回ごとに变化させていく場合には、変数:初期値, 継続条件, 増分の順番で記述せよ。例えば、変数 i を用いて i が初期値0で1ずつ増加させ、 N になったら終了する場合には、 $i:0, i < N, 1$ という形で記述する。

設問2 Exptext の内容が次の数式のととき、プログラム中の α で示す箇所での Postfix の内容を記述せよ。

Exptext の内容

(2	+	4)	×	3	+	5	×	6
---	---	---	---	---	---	---	---	---	---	---

設問3 引数で指定された要素の優先度を返す関数 `getPriority` のプログラムを次に示す。プログラム中の に入れる正しい答えを記述せよ。

[プログラム]

```
○ 整数型: getPriority ( 文字型: element )
/* 関数 getPriority は、数字、演算子、括弧又はスタック制御文字を
   element で受け取り、その優先度を整数型で返す */

/* 文字型の配列 elem を初期化する */
/* 添字の範囲は 0 ~ 16 である */
○ 文字型: elem[0:16] ← { "0", "1", "2", "3", "4", "5", "6", "7",
                        "8", "9", "+", "-", "×", "÷", "(", ")" , EOS }
/* 整数型の配列 priority を elem の各要素の優先度で初期化する */
/* 添字の範囲は 0 ~ 16 である */
○ 整数型: priority[0:16] ← { 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 1, 1, 2, 2, 4, 0, -1 }

○ 整数型: i

  • i ← 0
  ■ 
  |
  | • i ← i + 1
  ■
  • return priority[i] /* element の優先度を返す */
```