

情報 III

次の文章の空欄 (20) ~ (54) には、選択肢からもっとも適切なものを選び、その番号を解答欄にマークしなさい。

加算を +、減算を -、乗算を *、除算を / という演算子で表し、 a, b, c を数値とすると、

$$a + b * c \quad (1)$$

という計算式の計算をコンピュータで行う場合には $b * c$ の乗算を先に計算して、その結果を a と加算して、計算結果を得る。この計算式表記法を自然記法と呼ぶ。計算式全体をまず調査して、計算の実行順を決めることで、自然記法のままだでも計算できるが処理アルゴリズムは複雑となる。

そこで、自然記法の計算式を中間的記法に書き換え、書き換えた計算式に、スタックと呼ぶ一時的記憶領域を用いた単純な手順を繰り返し当てはめることで、計算結果を得ることを考える。スタックは書き込み (push) と取り出し (pop) を用いた記憶方式である。pop すると、最後に push した要素が取り出される。中間的記法における要素は演算子あるいは数値である。スタックの記憶領域は必要なだけ得られるものとする。

コンピュータは中間的記法で記述された計算式に対して、その先頭から要素を一つずつ取り出し、以下の手順を繰り返すことで計算を行う。

1. 取り出した要素が数値の場合には、その数値をスタックに push する。
2. 取り出した要素が演算子の場合には、スタックから 2 つ数値を pop して、その演算子をあてはめた計算結果を push する。計算時には先に pop した数値が演算子の後に配置されるとする。
3. これを計算式の最後まで繰り返し、最後にスタックに残った値が計算式の計算結果となる。

自然記法計算式 $a + b * c$ は

$$abc * +$$

と中間的記法に変換したのち、この手順を適用することで計算が可能となる。この記法変換を \Rightarrow を用いて以下のように表す。

$$a + b * c \Rightarrow abc * +$$

この中間的記法は逆ポーランド記法と呼ばれており、コンピュータ内部での中間的記法として使われる。図 1 に処理中の逆ポーランド記法計算式とスタックの状態例を示す。図中「計算」と書いてある部分は pop した 2 つの数値を所定の演算子を用いて計算することを意味している。push(a) は a をスタックに push すること、push($b * c$) は $b * c$ の計算結果をスタックに push することである。

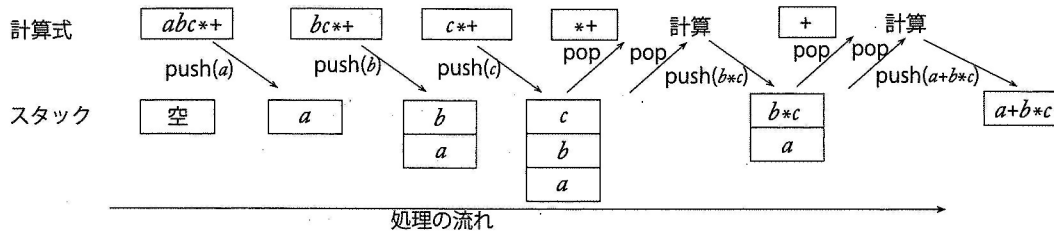


図1 逆ポーランド記法を用いた計算例

式 (1) は

$$a + b * c \Rightarrow bc * a +$$

などに変換しても同じ計算結果が得られるが、以下の (ア)、(イ) とも、数値の出現順は自然記法と逆ポーランド記法で変えないことにする。

自然記法における要素は、演算子、数値、括弧のいずれかとなる。自然記法で括弧がある計算式は、逆ポーランド記法を用いると括弧なしで表記できる。例えば以下のようなものである。

$$(a + b) * c \Rightarrow ab + c *$$

(ア) 以下の自然記法の計算式を4つの数値と3つの演算子から成る逆ポーランド記法に書換えなさい。

- $a + b * c + d \Rightarrow$ (20) (21) (22) (23) (24) d (25)
- $a + b * (c + d) \Rightarrow$ (26) (27) (28) (29) (30) (31) (32)
- $(a + b) * (c + d) \Rightarrow$ (33) (34) (35) (36) (37) (38) (39)
- $a * b / c + d \Rightarrow$ (40) (41) (42) c (43) (44) (45)

【(20)~(45)の選択肢】

- (1) a (2) b (3) c (4) d
- (5) $+$ (6) $-$ (7) $*$ (8) $/$

(イ) 次に自然記法で書かれた計算式を入力とし、その先頭から要素(演算子、数値、括弧のいずれか)を順次取り出してスタックを用いて処理し、逆ポーランド記法計算式を出力するアルゴリズムを作成する。アルゴリズムを以下の処理Aと処理Bで構成する。自然記法で書かれた計算式から順次取り出す要素を x 、popするとスタックから取り出せる要素を p とする。

[処理 A] : 数値を自然記法計算式に現われる順で出力し、括弧や演算の優先順を考慮して、演算子の出力順をスタックで調整する。括弧は逆ポーランド記法出力に含まれないので適切なタイミングで破棄する。

[処理 B] : 処理 A を行った後にスタックに要素が残る場合は、順に pop して出力する。

処理 A において、次の条件の場合に最も適した操作を下の選択肢から選び、その番号を解答欄にマークしなさい。ただし、 p が何であるかは pop せずに分かるものとする。また、処理が p に対してのみ行われる場合には x は保持し、新たに自然記法計算式から要素を取り出さないとする。さらに、与えられた自然記法計算式に誤り（式として成り立たないような演算子、数値、括弧の並び）はないものとする。

条件	処理
x が数値の場合	(46)
x が (の場合	(47)
x が)、かつ p が (の場合	(48)
x が)、かつ p が (でない場合	(49)
x が演算子、かつ スタックが空の場合	(50)
x が * あるいは /、かつ p が * あるいは / の場合	(51)
x が * あるいは /、かつ p が * や / でない場合	(52)
x が + あるいは -、かつ p が演算子の場合	(53)
x が + あるいは -、かつ p が演算子でない場合	(54)

【(46)～(54)の選択肢】

- (1) x を push する
- (2) x を出力する
- (3) p を pop して廃棄し、 x を廃棄する
- (4) x を廃棄する
- (5) p を pop して出力する
- (6) p を pop して出力し、 x を出力する
- (7) x を出力し、 p を pop して出力する