

午後問題の重点対策(基本情報技術者) 解答・解説
アルゴリズム

平成 15 年度春 基本情報技術者 午後問 4

【解答】

[設問] a - ウ, b - オ, c - イ, d - オ, e - エ

【解説】

逆ポーランド表記法とスタックはどちらもよく取り上げられるテーマである。これらは午前問題にもよく出題されることから、内容的には比較的取り組みやすかったのではないかと思われる。プログラムは2ページにわたっているが、記号ごとの並列的なアルゴリズムが順に出てくるので、プログラムの構成はつかみやすい。プログラムの構成がつかめたら、空欄を考えていくが、問題の例で挙げられている数式を当てはめながら、図を描いてトレースしていくとよい。

設問に取り組む前に重要なポイントについて整理する。

(1) プログラムの説明の中で、問題を解くうえで特に注意しなければならない点は次のとおりである。

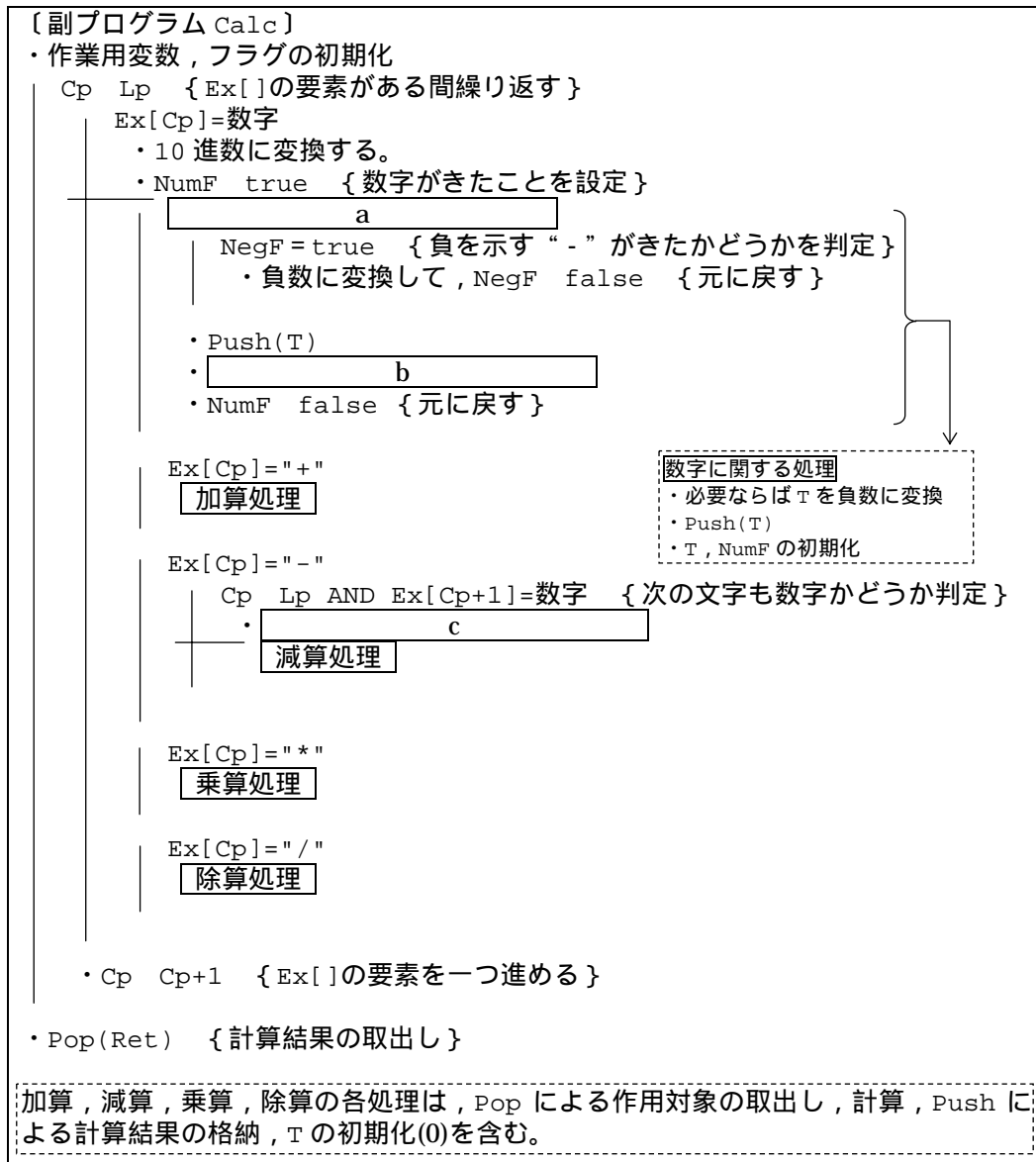
- ・ 正の整数にはプラス記号は付けない。
- ・ 先頭の整数を除き、整数の前には一つの空白文字を置く。
- ・ 数字 1 文字を実数に変換する関数 `ToReal` を使用する。
- ・ 変数 `Sp` はスタックの操作位置を示し、大域変数として定義されている。変数 `Sp` の初期値は 0 である。

(2) 副プログラム `Calc` で使用される変数(引数として説明されたもの以外)の用途を明らかにする。

変数名	型	説明
<code>Cp</code>	整数型	<code>Ex[]</code> を参照するための変数
<code>X, Y</code>	実数型	副プログラム <code>Pop</code> でスタックから取り出した実数を格納する変数
<code>NumF</code>	論理型	数字がきているかどうかを表すフラグ true : 数字, false : 数字以外
<code>NegF</code>	論理型	"-"がきているかどうかを表すフラグ true : "-", false : "-"以外

(3) 副プログラム `Calc` の概要を整理する。特にフラグ `NumF`, `NegF` の値の設定, フラグに関する条件式が重要であり、空欄を考えるにあたって、その役割を理解する必要がある。

午後問題の重点対策(基本情報技術者) 解答・解説
アルゴリズム



数字に関する処理

- ・必要ならば T を負数に変換
- ・Push(T)
- ・T, NumF の初期化

加算, 減算, 乗算, 除算の各処理は, Pop による作用対象の取出し, 計算, Push による計算結果の格納, T の初期化(0)を含む。

整理したことをもとに設問について考えていく。

午後問題の重点対策(基本情報技術者) 解答・解説 アルゴリズム

[空欄 a]

空欄 a のある位置は、条件式「 $Ex[Cp]=\text{数字}$ 」が偽のときの処理の一部であるから、現在の変数 Cp は数字以外の文字を指している、つまり、数字以外の文字がきているという前提で考える。

空欄 a は数字に関する処理の最初の条件式である。ヒントは、数字に関する処理の最後にある $NumF \text{ false}$ である。フラグ $NumF$ を元に戻すということは、それ以前は $NumF=true$ 、つまり数字がきていたと考えられる。問題の例では、1, 2 と数字がきていたが、空白がきたので、10 進数の数値 12 としてスタックに積む処理をするところが該当する。-, 3, 4, 5, / のところも同様である。

< 例 >

	0	1	2	3	4	5	6	7
Ex	1	2		-	3	4	5	/

Cp
----->↑

数字の区切りは、空白文字とは限らないので、空白文字での判定はできない。ここで、共通して使用できるのが、フラグ $NumF$ である。数字がきていたときは必ず $NumF=true$ となっていることを利用して、空欄 a には(ウ) $NumF=true$ を入れる。

[空欄 b]

$Push(T)$ を行って、変数 T の数値をスタックに積んだ。スタックに積んでしまえば、変数 T の作業用の領域としての役割は終わるので、次の数値のために 0 に初期化する必要がある。したがって、空欄 b には(オ) $T = 0$ が入る。

[空欄 c]

空欄 c は、 $Ex[Cp]$ が“-”のときの処理の中にある。“-”の記号は、負の数の表現のため符号として使用されたり、減算記号として使用されたりする。空欄 c の部分は、次に数字が続くとき、つまり負数の表現に使用される場合の処理である。実際に負数に変換する処理は、空欄 a のところでも述べた数字に関する処理の中にある。そこでは、フラグ $NegF$ を利用して $true$ ならば負数に変換するようにしている。したがって、“-”が負数の表現に使用されることが分かった時点(空欄 c のところ)で、(イ) $NegF = true$ を行えばよい。

[空欄 d]

空欄 d は、副プログラム $Push(T)$ の重要な処理部分である。 T をスタックに積む処理

午後問題の重点対策(基本情報技術者) 解答・解説
アルゴリズム

で、スタックの操作位置を示す変数 Sp の扱いがポイントになる。スタックに積む様子を問題の例で考えてみる。ここでは、最初 $Sp=0$ とする。

<例>

	0	1	2	3	4	5	6	7
Ex	1	2		-	3	4	5	/

Push(12)を行う直前	Push(-345)を行う直前	-345を積んだ後
<p>12</p> <p>Stack structure: Index 2: ... Index 1: ... Index 0: 12 Sp points to index 0.</p>	<p>-345</p> <p>Stack structure: Index 2: ... Index 1: ... Index 0: 12 Sp points to index 0.</p>	<p>Stack structure: Index 2: ... Index 1: -345 Index 0: 12 Sp points to index 1.</p>

以上から、変数 Sp は次に積むことができる $Stack[]$ の位置を示しており、処理の手順としては、 $Stack[Sp]$ T を行った後、変数 Sp を一つ進めること($Sp = Sp+1$)が分かる。したがって、空欄 d には(オ)が入る。

なお、空欄 d は $Sp=Max$ のときも実行されるので、仮に(ア)のように $Sp = Sp+1$ を先に行うと Sp が Max より大きい値で $Stack[Sp]$ T が実行されることになってしまい不適切である。このことから変数 Sp は次に積むことができる $Stack[]$ の位置を示していることが分かる。

[空欄 e]

空欄 e は、副プログラム $Pop(T)$ の重要な処理部分である。変数 Sp は次に積む $Stack[]$ の位置を示しているので、まず Sp から 1 減算して、その後でスタックから取り出す処理を行う。空欄 d で使用した例で検証してみるとよい。したがって、処理の手順としては、 $Sp = Sp-1$, $T = Stack[Sp]$ となり、空欄 e には(エ)が入る。