

次の問 8 は必須問題です。必ず解答してください。

問 8 次のプログラムの説明及びプログラムを読んで、設問 1, 2 に答えよ。

1 行の代入文を解析し、演算の優先順位に従って一つずつ演算を行っていく一連の代入文に変換して出力する。代入文とその出力結果の例を、次に示す。ここで、出力結果にある wk#1 などは、中間結果を保存するための作業変数である。

(代入文の例)
 $Ans = X1 + 10 * X2$

(出力結果の例)
 $wk\#1 = 10 * X2$
 $wk\#2 = X1 + wk\#1$
 $Ans = wk\#2$

代入文の形式は、次のとおりである。

変数 = 式

“=” は代入演算子で、右辺の式の値を評価した結果を、左辺の変数に代入する。

式は、変数又は定数（以下、項という）の一つ以上の並びで、項が二つ以上のときは各項の間に算術演算子を置く。変数は、英字 (“A” ~ “Z”, “a” ~ “z”) で始まる 1 文字以上の英数字の列である。定数は、1 文字以上の数字 (“0” ~ “9”) の列である。

算術演算子は、加算 (“+”), 減算 (“-”), 乗算 (“*”), 除算 (“/”) の 4 種類である。乗除算は加減算に優先する。同一優先順位の算術演算子は、左から順に演算する。

[代入文の構文解析]

解析処理は、次の手順による。

- (1) 代入文を走査して、英字・数字・代入演算子・算術演算子以外の文字が含まれていれば、err に -1 を格納して手順 (5) へ進む。err は、文法上の誤りがあった場合のエラーコードを格納する変数である。
- (2) n 文字からなる代入文を、文字型配列 S の要素番号 1 ~ n に格納し、S[0] に開始マーク “«” を、S[n+1] に終端マーク “»” を、それぞれ格納する。次に、整数型配列 V を用意し、S[i] の内容に対応した表 1 のコードを V[i] に格納する。その例を、次に示す。

配列 S の内容	«	A	n	s	=	X	1	+	1	0	*	X	2	»
配列 V の内容	0	2	2	2	10	2	3	11	3	3	12	2	3	1

表1 文字とコードの対応

S[i]の内容	«	»	英字	数字	=	+, -, *, /
V[i]に格納する内容	0	1	2	3	10	11 12

- (3) 初期値として st に“開始”を、err 及び i に 0 を、それぞれ格納する。st は、解析の状態を格納している変数である。
- (4) i に 1 を加算する。表 2 で、現在の st の状態（行）と着目している文字 S[i] の内容（列）が交差するセルの内容を実行する。ただし、セルが空白の場合は、何も実行しない。実行の結果、err の値が 0 以外となったら手順 (5) へ、st の値が“終端”となったら手順 (6) へそれぞれ進む。それ以外の場合は、この手順 (4) を繰り返す。

表2 状態遷移表

st \ S[i]	英字	数字	=	+, -, *, /	»
“開始”	st ← “左辺”	err ← 12	err ← 13	err ← 14	err ← 15
“左辺”			st ← “代入”	err ← 24	err ← 25
“代入”	st ← “変数”	st ← “定数”	err ← 33	err ← 34	err ← 35
“変数”			err ← 43	st ← “演算”	st ← “終端”
“定数”	err ← 51		err ← 53	st ← “演算”	st ← “終端”
“演算”	st ← “変数”	st ← “定数”	err ← 63	err ← 64	err ← 65

- (5) err の値に応じて適切なエラーメッセージを表示し、処理を終了する。
- (6) 文法上の誤りがなかった旨を表示し、処理を終了する。

設問1 [代入文の構文解析]に関する次の記述中の に入れる正しい答えを、解答群の中から選べ。

文法上の誤りがある次の代入文①～④を、解析処理の手順によって解析した。

代入文① Answer=One+Two+Three+

代入文② HexaSum=7FFF+0001

代入文③ Position=Index++

代入文④ X1+10*X2=Ans

解析処理の手順 (5) に進んだとき、 a の場合は err の値が 51 に、

b の場合は err の値が 64 になっている。

解答群

ア 代入文①

イ 代入文②

ウ 代入文③

エ 代入文④

〔代入文の変換〕

変換処理は、次の手順による。〔代入文の構文解析〕の手順は実行済みで、代入文には文法上の誤りがないものとする。

- (1) 初期値として、w に 0 を格納する。
- (2) 配列 S、V の開始マークから終端マークまでの範囲を検査し、優先順位が最も高い最初の算術演算子の要素番号を変数 next に格納する。その例を、次に示す。算術演算子がないければ、next に 0 を格納する。

配列 S |«|A|n|s|=|X|1|+|1|0|*|X|2|»|
 ↑ next ← 10

- (3) next が 0 なら、配列 S 中の代入文を出力して、処理を終了する。
- (4) next の位置の演算子とその前後の項からなる文字列を抜き出す（図 1 の①）。w に 1 を加算し、出力する代入文を編集して出力する。出力する代入文は、文字列“wk#”，文字列に変換した値 w，文字“=”及び抜き出した文字列をこの順に連結したものである。次に、配列 S について、抜き出した元の文字列を、文字列“wk#”及び文字列に変換した値 w で置き換える（図 1 の②）。置き換えるときに、文字数が増える場合は以降の文字を後ろにずらし、減る場合は以降の文字を前に詰める。さらに、更新された配列 S の内容に応じて配列 V の内容を更新する。文字“#”に対応するコードは、英字のコードと同じとする。

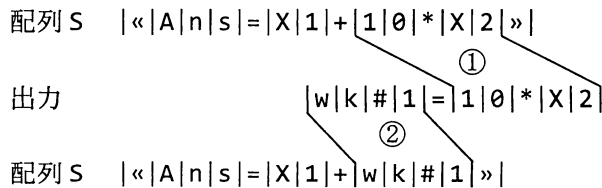


図 1 文字列の置換の例

- (5) 手順 (2) へ戻り、処理を繰り返す。

設問2 「代入文の変換」に関する次のプログラム中及び記述中の に入れる正しい答えを、解答群の中から選べ。

なお、プログラム中の関数 Getpos(配列, 値)は、値が格納されている配列中の最初の要素番号を返す。また、配列 S, V は大域変数として与えられ、副プログラム中から参照できるものとする。

(1) 手順(2)の処理を行う2種類の副プログラム1, 2を作成した。

[副プログラム1]

○整数型関数：プログラム1
 ○整数型：i, next, priority
 • i ← 1 ← **α**
 • next ← 0
 • priority ← 10
 ■ V[i] ≠ 1
 ■ c
 • next ← i
 • priority ← V[i]
 • i ← i + 1
 ■
 • return next

[副プログラム2]

○整数型関数：プログラム2
 ○整数型：i, next, priority
 • i ← Getpos(S[], "»") - 1
 • next ← 0
 • priority ← 11
 ■ V[i] ≠ 0
 ■ d
 • next ← i
 • priority ← V[i]
 • i ← i - 1
 ■
 • return next

プログラム1の **α** の行では、i に1を格納している。

ここで、**α** の行を “i ← e” とすれば、繰返し処理の繰返し回数を最小にすることができる。

(2) 手順(4)の処理中で使用する副プログラム3を作成した。このプログラムは、配列 S, V の要素番号 from 以降（終端マークまで）の要素を前後に移動する。move は、移動する桁数と方向を示し、符号が正の場合は後ろにずらし、符号が負の場合は前に詰める。move の値が -3(左に3桁詰める) の場合の例を、次に示す。ここで、配列 S, V は十分な大きさがあるものとする。

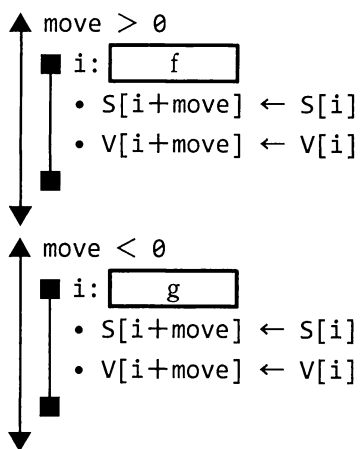
配列 S (移動前) |«|A|n|s|=|w|k|#|2|+|1|0|+|w|k|#|1|»|
 ↑ from
 配列 S (移動後) |«|A|n|s|=|w|k|#|2|+|w|k|#|1|»|

[副プログラム 3]

○プログラム 3(整数型 : from, 整数型 : move)

○整数型 : i, to

• to ← Getpos(S[], "»")



c, dに関する解答群

ア priority < V[i]

イ priority ≤ V[i]

ウ priority ≥ V[i]

エ priority > V[i]

eに関する解答群

ア Getpos(S[], "«") + 1

イ Getpos(S[], "«") + 2

ウ Getpos(S[], "=") - 1

エ Getpos(S[], "=") + 2

f, gに関する解答群

ア from, i < to, 1

イ from, i ≤ to, 1

ウ to, i ≥ from, -1

エ to, i > from, -1