

【解答】

【設問1】 aーf, bーy

【設問2】 cーf, dーf, eーf, fーf, gーf

【解説】

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を変換して出力する、アルゴリズムの問題である。設問は、解析処理のトレースと変換処理で使用する副プログラムの穴埋め問題で構成されている。テーマだけを見ると難しそうに感じるが、問題文中に各処理の手順が示されており、与えられた例を使って一つずつ内容を理解していけば解答は可能である。アルゴリズム問題に付き苦手意識を持っている受験者もいると思われるが、問8は必須問題なので何とか克服する必要がある。問題文を読み取る力、図を描いて処理の過程を追う（トレース）力、配列などのデータ構造に関する知識などを身に付け、過去問題を利用して問題を解くことに慣れておくことが大切である。

【設問1】

文法上の誤りがある代入文を解析する問題である。代入文の解析処理は手順(4)を繰り返して行う。解析には表2の状態遷移表を用いるため、最初にこの表の見方を確認する。なお、解析する代入文は「ans=X1+10×X2」とし、配列 S に次のように格納されているものとする。

	0	1	2	3	4	5	6	7	8	9	10	11	12	13
配列 s の内容	«	A	n	s	=	X	1	+	1	0	*	X	2	»

表2 状態遷移表

S[i] \		列		行	
		英字	数字	=	+, -, *, /
st	“開始”	st ← “左辺”	err ← 12	err ← 13	err ← 14
	“左辺”			st ← “代入”	err ← 24
“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”
	st ← “変数”	st ← “定数”	err ← 63	err ← 64	err ← 65

表2の状態遷移表は、stの状態が行、S[i]の内容が列となっており、解析処理はそれらが交差するセルの内容を実行しながら進む。初期値としてstには“開始”を格納するので、行は“開始”から始まる。また、処理(4)ではiの値は1から始まるので、列はS[1]の内容(A)の「英字」から始まる。この二つが交差するセルの内容が「st ← “左辺”」なので、stは“左辺”が格納され、行は“左辺”に移る。繰返しの都度iは1ずつ増えていくので、S[i]の内容に対応して列も変化する。前図の式は、次図のように遷移しながら解析処理が行われる。なお、errの値が0以外、又はstの値が“終端”となると手順(4)の繰返しを終了する。

行 (st)	列 (S[i])	セル (実行する) の内容
“開始”	A : 「英字」	st ← “左辺”
“左辺”	n : 「英字」	何も実行しない
“左辺”	s : 「英字」	何も実行しない
“左辺”	= : 「=」	st ← “代入”
“代入”	X : 「英字」	st ← “変数”
“変数”	1 : 「数字」	何も実行しない
“変数”	+ : 「+, -, *, /」	st ← “演算”
“演算”	1 : 「数字」	st ← “定数”
“定数”	0 : 「数字」	何も実行しない
“定数”	* : 「+, -, *, /」	st ← “演算”
“演算”	X : 「英字」	st ← “変数”
“変数”	2 : 「数字」	何も実行しない
“変数”	» : 「»」	st ← “終端”

・空欄 a, b : 前述の説明にならって各代入文を順に解析すると解答を導くことができるが、ここではもう少し列挙良く解答を導くために、errの値が51になる場合、64になる場合を、状態遷移表を逆順にたどって確認する。

セルの内容を見ると、errに51が格納されるのは、stが“定数”でS[i]の内容が「英字」の場合である。また、stの値に“定数”が格納されるのは、stが“代入”でS[i]の内容が「数字」、又はstが“演算”でS[i]の内容が「数字」の場合である。更に、stに“代入”が格納されるのは、stが“左辺”でS[i]が「=」の場合であり、stに“演算”が格納されるのは、stが“変数”でS[i]が「+, -, *, /」(算術演算子)のいずれか、又はstが“定数”でS[i]が「+, -, *, /」のいずれかの場合である。これらを整理すると、errの値が51になるのは、次のパターンである。

- ・「=」の後に「数字」、その後に「英字」がある
 - ・「+, -, *, /」のいずれかの後に「数字」、その後に「英字」がある
- 代入文①～④を見ると、②の代入文が「=」の後に「数字」、その後に「英字」があり、1つ目のパターンに当てはまる。よって、空欄 a は (4) が正解となる。

同様にerrの値が64になる場合を、状態遷移表を逆順にたどって確認する。セルの内容を見ると、errに64が格納されるのは、stが“演算”でS[i]の内容が「+, -, *, /」のいずれかの場合である。また、stに“演算”が格納されるのは、stが“変数”でS[i]の内容が「+, -, *, /」のいずれか、又はstが“定数”でS[i]の内容が「+, -, *, /」のいずれかの場合である。これを整理すると、errの値が64になるのは、次のパターンの場合である。

- ・「+, -, *, /」のいずれかの後に「+, -, *, /」のいずれかがある

代入文①～④を見ると、算術演算子の後に算術演算子が続く代入文は③である。よって、空欄 b は (7) が正解である。

【設問2】

【副プログラム1】

○整数型関数：プログラム1

○整数型 : i, next, priority

1	・ i ← 1	← α	priority
2	・ next ← 0		
3	・ priority ← 10	← priority の初期値を 10 (= の値) に	
4	■ V[i] ≠ 1	← V[i] の値が 1 (≠) でない間は繰り返す	
5	▲ c		
6	・ next ← i		
7	・ priority ← V[i]		
8		現在の演算子よりも優先順位が高い演算子が出てきたら内容を置き換える	
9	・ i ← i + 1	← 1 文字後へ	
10	■		
11	・ return next	← 優先順位が最も高い最初の算術演算子の要素番号を返す	

【副プログラム2】

○整数型関数：プログラム2

○整数型 : i, next, priority

21	・ i ← Getpos(S[], "»") - 1	← 「»」が格納されている配列の要素番号から 1 を引いた値を i に格納
22	・ next ← 0	
23	・ priority ← 11	← priority の初期値を 11 (+, - の値) に
24	■ V[i] ≠ 0	← V[i] の値が 0 (≠) でない間は繰り返す
25	▲ d	
26	・ next ← i	
27	・ priority ← V[i]	
28		現在の演算子よりも優先順位が高い演算子が出てきたら内容を置き換える
29	・ i ← i - 1	← 1 文字前へ
30	■	
31	・ return next	← 優先順位が最も高い最初の算術演算子の要素番号を返す

・空欄 c, d : 【副プログラム1】、【副プログラム2】は、いずれも代入文の変換処理の手順(2)を行う。手順(2)は、配列 S, V に格納された代入文を 1 文字ずつ検査し、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 に格納する処理である。選択肢を見ると、空欄 c, d はともに変数 priority の内容と配列 V[i] の内容と比較している。[代入文の構文解析] の手順(2)の配列 S、配列 V の例及び表1の「文字とコードの対応表」より、配列 V には S の内容に対応した値が格納されていることが分かる。したがって、算術演算子の場合、「+, -」ならば 11, 「*, /」ならば 12 が格納されている。

【副プログラム1】では、i の初期値は 1 であり、配列 V の内容を先頭から検査する。Priority は現時点で最も優先順位の高い演算子に対応する値を保持する変数であり、初期値に 10 (= の値) を格納する。算術演算子の値は 11 (+, -) 又は 12 (*, /) と priority の初期値よりも大きく、また優先順位は「+, -」や「-」よりも「*」や「/」の方が高いので、priority と next の内容を置き換えるのは、priority < V[i] の場合であることが分かる。なお、最初の位置を格納するので、仮に V[5] と V[8] がともに 12 であった場合、next は 5 となる必要がある。したがって、priority ≤ V[i] にはならないことに注意が必要である。よって、空欄 e は (7) が正解である。

【副プログラム2】では、i の初期値は関数 Getpos (配列、値) を使って求めている。Getpos は値が格納されている配列中の最初の要素番号を返すので、行番号 21 の処理は、配列の最後の要素 (») の要素番号の 1 つ前の値を i に格納することになる。行番号 29 の処理で i を -1 しており、これらの処理から配列 V の内容は後ろから検査することが分かる。priority の初期値は 11 (+, -) であり、算術演算子の値が 12 (*, /) である場合に、priority の内容を置き換えることになる。よって、priority < V[i] が成り立つ場合に priority と next の内容を置き換えるが、優先順位が最も高い最初の算術演算子の要素番号を求めるので、V[5] と V[8] が 12 であった場合、next は 5 となる必要がある。【副プログラム2】は後ろから処理を行うので、先に V[8]、次に V[5] が比較対象となるので、next を 5 とするためには priority ≤ V[i] としなければならぬ。よって、空欄 d は (4) が正解である。

・空欄 e : 繰返し回数を最小にするように i の初期値を変更する。【副プログラム1】では i の初期値は 1 なので代入文の先頭からチェックすることになるが、算術演算子が登場するのは右辺であり、「=」より後ろの文字から検査すればよい。正しい代入文の場合、「=」の直後（一つ後ろ）は必ず「英字」又は「数字」が並ぶため、算術演算子が現れる最初の位置は「=」の二つ後ろになる。よって、空欄 e は (エ) が正解である。

【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

41	・ to ← Getpos(S[], "»")	配列 S の最後の要素番号を to に格納
42	▲ move > 0	
43	■ i : f	
44	・ S[i + move] ← S[i]	
45	・ V[i + move] ← V[i]	
46	■	
47	▼	
48	move < 0	
49	■ i : g	
50	・ S[i + move] ← S[i]	
51	・ V[i + move] ← V[i]	
52	▼	
53	▼	

← 配列 S, V の値を後ろに移動

← 配列 S, V の値を前に移動

【解答】

【設問1】 aーf, bーy

【設問2】 cーf, dーf, eーf, fーf, gーf

【解説】

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を変換して出力する、アルゴリズムの問題である。設問は、解析処理のトレースと変換処理で使用する副プログラムの穴埋め問題で構成されている。テーマだけを見ると難しそうに感じるが、問題文中に各処理の手順が示されており、与えられた例を使って一つずつ内容を理解していけば解答は可能である。アルゴリズム問題に付き苦手意識を持っている受験者もいると思われるが、問8は必須問題なので何とか克服する必要がある。問題文を読み取る力、図を描いて処理の過程を追う（トレース）力、配列などのデータ構造に関する知識などを身に付け、過去問題を利用して問題を解くことに慣れておくことが大切である。

【設問1】

文法上の誤りがある代入文を解析する問題である。代入文の解析処理は手順(4)を繰り返して行う。解析には表2の状態遷移表を用いるため、最初にこの表の見方を確認する。なお、解析する代入文は「ans=X1+10×X2」とし、配列 S に次のように格納されているものとする。

	0	1	2	3	4	5	6	7	8	9	10	11	12	13
配列 s の内容	«	A	n	s	=	X	1	+	1	0	*	X	2	»

表2 状態遷移表

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

表2の状態遷移表は、stの状態が行、S[i]の内容が列となっており、解析処理はそれらが交差するセルの内容を実行しながら進む。初期値としてstには“開始”を格納するので、行は“開始”から始まる。また、処理(4)ではiの値は1から始まるので、列はS[1]の内容(A)の「英字」から始まる。この二つが交差するセルの内容が「st ← “左辺”」なので、stは“左辺”が格納され、行は“左辺”に移る。繰返しの都度iは1ずつ増えていくので、S[i]の内容に対応して列も変化する。前図の式は、次図のように遷移しながら解析処理が行われる。なお、errの値が0以外、又はstの値が“終端”となると手順(4)の繰返しを終了する。

行 (st)	列 (S[i])	セル (実行する) の内容
開始	A : 「英字」	st ← “左辺”
左辺	n : 「英字」	何も実行しない
左辺	s : 「英字」	何も実行しない
左辺	= : 「=」	st ← “代入”
代入	X : 「英字」	st ← “変数”
変数	1 : 「数字」	何も実行しない
変数	+: 「+, -, *, /」	st ← “演算”
演算	1 : 「数字」	st ← “定数”
定数	0 : 「数字」	何も実行しない
定数	* : 「+, -, *, /」	st ← “演算”
演算	X : 「英字」	st ← “変数”
変数	2 : 「数字」	何も実行しない
変数	» : 「»」	st ← “終端”

・空欄 a, b : 前述の説明にならって各代入文を順に解析すると解答を導くことができるが、ここではもう少し列挙良く解答を導くために、errの値が51になる場合、64になる場合を、状態遷移表を逆順にたどって確認する。

セルの内容を見ると、errに51が格納されるのは、stが“定数”でS[i]の内容が「英字」の場合である。また、stの値に“定数”が格納されるのは、stが“代入”でS[i]の内容が「数字」、又はstが“演算”でS[i]の内容が「数字」の場合である。更に、stに“代入”が格納されるのは、stが“左辺”でS[i]が「=」の場合であり、stに“演算”が格納されるのは、stが“変数”でS[i]が「+, -, *, /」(算術演算子)のいずれか、又はstが“定数”でS[i]が「+, -, *, /」のいずれかの場合である。これらを整理すると、errの値が51になるのは、次のパターンである。

・「=」の後は「数字」、その後に「英字」がある
・「+, -, *, /」のいずれかの後に「数字」、その後に「英字」がある
代入文①～④を見ると、②の代入文が「=」の後に「数字」、その後に「英字」があり、1つ目のパターンに当てはまる。よって、空欄 a は (4) が正解となる。

同様にerrの値が64になる場合を、状態遷移表を逆順にたどって確認する。セルの内容を見ると、errに64が格納されるのは、stが“演算”でS[i]の内容が「+, -, *, /」のいずれかの場合である。また、stに“演算”が格納されるのは、stが“変数”でS[i]の内容が「+, -, *, /」のいずれか、又はstが“定数”でS[i]の内容が「+, -, *, /」のいずれかの場合である。これを整理すると、errの値が64になるのは、次のパターンの場合である。

・「+, -, *, /」のいずれかの後に「+, -, *, /」のいずれかがある

代入文①～④を見ると、算術演算子の後に算術演算子が続く代入文は③である。よって、空欄 b は (7) が正解である。

【設問2】

【副プログラム1】

○整数型関数：プログラム1

○整数型 : i, next, priority

```
1  i ← 1
2
3  next ← 0
4  priority ← 10
5  while V[i] ≠ 1
6      next ← i
7      priority ← V[i]
8
9  i ← i + 1
10
11 return next
```

現在の演算子よりも優先順位が高い演算子が出てきたら内容を置き換える

優先順位が最も高い最初の算術演算子の要素番号を返す

【副プログラム2】

○整数型関数：プログラム2

○整数型 : i, next, priority

```
21 i ← Getpos(S[i], "»") - 1
22 next ← 0
23 priority ← 11
24 while V[i] ≠ 0
25     next ← i
26     priority ← V[i]
27
28 i ← i - 1
29
30
31 return next
```

「»」が格納されている配列の要素番号から1を引いた値をiに格納

priorityの初期値を10 (= の値) に

V[i]の値が1 (s) でない間は繰り返す

V[i]の値が0 (d) でない間は繰り返す

現在の演算子よりも優先順位が高い演算子が出てきたら内容を置き換える

1文字前へ

優先順位が最も高い最初の算術演算子の要素番号を返す

・空欄 c, d : 【副プログラム1】、【副プログラム2】は、いずれも代入文の変換処理の手順(2)を行う。手順(2)は、配列 S, V に格納された代入文を1文字ずつ検査し、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 に格納する処理である。選択肢を見ると、空欄 c, d はともに変数 priority の内容と配列 V[i] の内容と比較している。[代入文の構文解析] の手順(2)の配列 S、配列 V の例及び表1の「文字とコードの対応表」より、配列 V には S の内容に対応した値が格納されていることが分かる。したがって、算術演算子の場合、「+, -」ならば11, 「*, /」ならば12が格納されている。

【副プログラム1】では、iの初期値は1であり、配列 V の内容を先頭から検査する。Priority は現時点で最も優先順位の高い演算子に対応する値を保持する変数であり、初期値に10 (= の値) を格納する。算術演算子の値は11 (+, -) 又は12 (*, /) と priority の初期値よりも大きく、また優先順位は「+, -」や「-」よりも「*」や「/」の方が高いので、priority と next の内容を置き換えるのは、priority < V[i] の場合であることが分かる。なお、最初の位置を格納するので、仮に V[5] と V[8] がともに12であった場合、next は5となる必要がある。したがって、priority ≤ V[i] にはならないことに注意が必要である。よって、空欄 c は (7) が正解である。

【副プログラム2】では、iの初期値は関数 Getpos (配列、値) を使って求めている。Getpos は値が格納されている配列中の最初の要素番号を返すので、行番号21の処理は、配列の最後の要素 (s) の要素番号の1つ前の値をiに格納することになる。行番号29の処理でiを-1しており、これらの処理から配列 V の内容は後ろから検査することが分かる。priority の初期値は11 (+, -) であり、算術演算子の値が12 (*, /) である場合に、priority の内容を置き換えることになる。よって、priority < V[i] が成り立つ場合に priority と next の内容を置き換えるが、優先順位が最も高い最初の算術演算子の要素番号を求めるので、V[5] と V[8] が12であった場合、next は5となる必要がある。【副プログラム2】は後ろから処理を行うので、先に V[8]、次に V[5] が比較対象となるので、next を5とするためには priority ≤ V[i] としなければならぬ。よって、空欄 d は (4) が正解である。

・空欄 e : 繰返し回数を最小にするようにiの初期値を変更する。【副プログラム1】ではiの初期値は1なので代入文の先頭からチェックすることになるが、算術演算子が登場するのは右辺であり、「=」より後ろの文字から検査すればよい。正しい代入文の場合、「=」の直後 (一つ後ろ) は必ず「英字」又は「数字」が並ぶため、算術演算子が現れる最初の位置は「=」の二つ後ろになる。よって、空欄 e は (エ) が正解である。

【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

```
41 to ← Getpos(S[i], "»")
42 move > 0
43 i : f
44 S[i+move] ← S[i]
45 V[i+move] ← V[i]
46
47
48 move < 0
49 i : g
50 S[i+move] ← S[i]
51 V[i+move] ← V[i]
52
53
```

配列 S の最後の要素番号を to に格納

配列 S, V の値を後ろに移動

配列 S, V の値を前に移動