

平成23年度前期 情報検定

<実施 平成23年9月4日（日）>

プログラミングスキル

（説明時間 10：00～10：10）

（試験時間 10：10～11：40）

- ・ 試験問題は試験開始の合図があるまで開かないでください。
- ・ 解答用紙（マークシート）への必要事項の記入は、試験開始の合図と同時に行いますので、それまで伏せておいてください。
- ・ 試験開始の合図の後、次のページを開いてください。＜受験上の注意＞が記載されています。必ず目を通してから解答を始めてください。
- ・ 試験問題は、すべてマークシート方式です。正解と思われるものを1つ選び、解答欄の○をHBの黒鉛筆でぬりつぶしてください。2つ以上ぬりつぶすと、不正解になります。
- ・ 辞書、参考書類の使用および筆記用具の貸し借りは一切禁止です。
- ・ 電卓の使用が認められます。ただし、下記の機種については使用が認められません。

<使用を認めない電卓>

1. 電池式（太陽電池を含む）以外の電卓
2. 文字表示領域が複数行ある電卓（計算状態表示の一行は含まない）
3. プログラムを組み込む機能がある電卓
4. 電卓が主たる機能ではないもの
 - * パソコン（電子メール専用機等を含む）、携帯電話（PHS）、電子手帳、電子メモ、電子辞書、翻訳機能付き電卓、音声応答のある電卓、電卓付腕時計等
5. その他試験監督者が不適切と認めるもの

＜受験上の注意＞

1. この試験問題は27ページあります。ページ数を確認してください。
乱丁等がある場合は、手をあげて試験監督者に合図してください。
※問題を読みやすくするために空白ページを設けている場合があります。
2. 解答用紙（マークシート）に、受験者氏名・受験番号を記入し、受験番号下欄の数字をぬりつぶしてください。正しく記入されていない場合は、採点されませんので十分注意してください。
3. 試験問題についての質問には、一切答えられません。自分で判断して解答してください。
4. 試験中の筆記用具の貸し借りは一切禁止します。筆記用具が破損等により使用不能となった場合は、手をあげて試験監督者に合図してください。
5. 試験を開始してから30分以内は途中退出できません。30分経過後退出する場合は、もう一度、受験番号・マーク・氏名が記載されているか確認して退出してください。なお、試験終了5分前の合図以降は退出できません。試験問題は各自お持ち帰りください。
6. 2011年度の受験者から、試験後にお知らせする合否結果（合否通知）、および合格者に交付する「合格証・認定証」はすべて、Webページ（PC、モバイル）での認証によるデジタル「合否通知」、デジタル「合格証・認定証」に移行しました。
 - ①団体宛にはこれまでと同様に合否結果一覧ほか、試験結果資料一式を送付します。
 - ②合否等の結果についての電話・手紙等でのお問い合わせには、一切応じられませんので、ご了承ください。

<問題の構成>

必須問題 全員解答

問題 1 ～ 問題 4	2 ページ～13 ページ
-------------------------	--------------

選択問題 次の問題から 1 問選択し解答せよ。

(選択した問題は解答用紙「選択欄」に必ずマークすること)

※選択欄にマークがなく、解答のみマークした場合は採点を行いません。

・ C 言語の問題	15 ページ～18 ページ
・ 表計算の問題	19 ページ～23 ページ
・ アセンブラの問題	24 ページ～27 ページ

必須問題

問題 1 次のランレングス符号化に関する記述中の に入れるべき適切な字句または数値を解答群から選べ。

ランレングス符号化とはデータ圧縮に用いられる符号化の一種で、連続する同じ値と、その個数の組み合わせで次のような形式で表現する。

[データ] [個数] [データ] [個数] …

例：X 5 Y 9 … データ X が 5 個連続し、データ Y が 9 個連続する。

ここで扱うデータは 1 バイトの英字とし、連続するデータが 255 個を超える場合は、255 個で区切り符号化する。これにより、データと個数をそれぞれ 1 バイトで表現できる。

なお、問題の構成上、個数は 10 進数で表記する。

また、ここでは圧縮率を次のように定義する。

圧縮率 = 符号化後のデータ量 ÷ 符号化前のデータ量

ただし、圧縮率は小数第 3 位を四捨五入する。

文字列「AAAAAABBBBAAAAAAAAA」(20 バイト) を符号化すると、文字列は A が 6 個、B が 4 個、A が 10 個なので、符号化すると次のようになる。

A 6 B 4 A 10

符号化前のデータ量は 20 バイトであり、符号化後は 6 バイトであるから、圧縮率は (1) となる。

また、文字列「AABBCDDEFGGH」(12 バイト) を符号化すると、

A 2 B 2 C 1 D 2 E 1 F 1 G 2 H 1

となる。このときの圧縮率は (2) となり、圧縮ではなく、逆に冗長になる。

次に、以下に示す①～⑤の 12 バイトの文字列を符号化したときについて考える。

① AAABCCDDDD

② AABCCDDEEFF

③ AAAAAABCDEEE

④ ABCDDDEEFFG

⑤ AAAABCDEFGHI

①～⑤の文字列のうち、もとのデータ量より圧縮できる (圧縮率が 1 未満である) 文字列は (3) である。この中で最も圧縮効果が高いのは文字列 (4) であり、圧縮率は 0.67 である。

一方、圧縮が有効でなく冗長となる (圧縮率が 1 を超える) 文字列は (5) である。この中で最も冗長性が高くなるのは文字列 (6) であり、圧縮率は 1.50 である。

(1), (2) の解答群

ア. 0.30 イ. 0.60 ウ. 0.75 エ. 1.33 オ. 3.33

(3), (5) の解答群

ア. ①と② イ. ①と③ ウ. ①と④ エ. ①と⑤ オ. ②と③
 カ. ②と④ キ. ②と⑤ ク. ③と④ ケ. ③と⑤ コ. ④と⑤

(4), (6) の解答群

ア. ① イ. ② ウ. ③ エ. ④ オ. ⑤

問題2 次の流れ図の説明を読み、設問に答えよ。

[流れ図の説明]

マスタファイルの内容をトランザクションファイルの内容で更新し、新マスタファイルを作成する処理である。

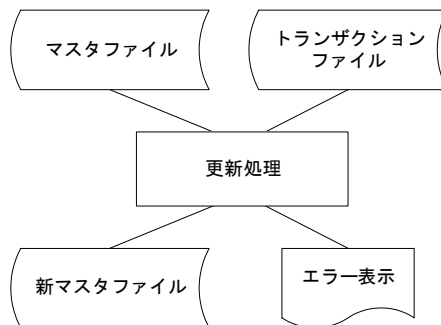


図1 ファイル更新のプロセスチャート

1. マスタファイルは順編成ファイルである。レコード形式は図2のようになっており、商品コードの昇順に整列されている。なお、商品コードが同じ値のレコードは存在しない。

商品コード	商品名	数量
MCODE	MNAME	MSURYO

図2 マスタファイルのレコード形式

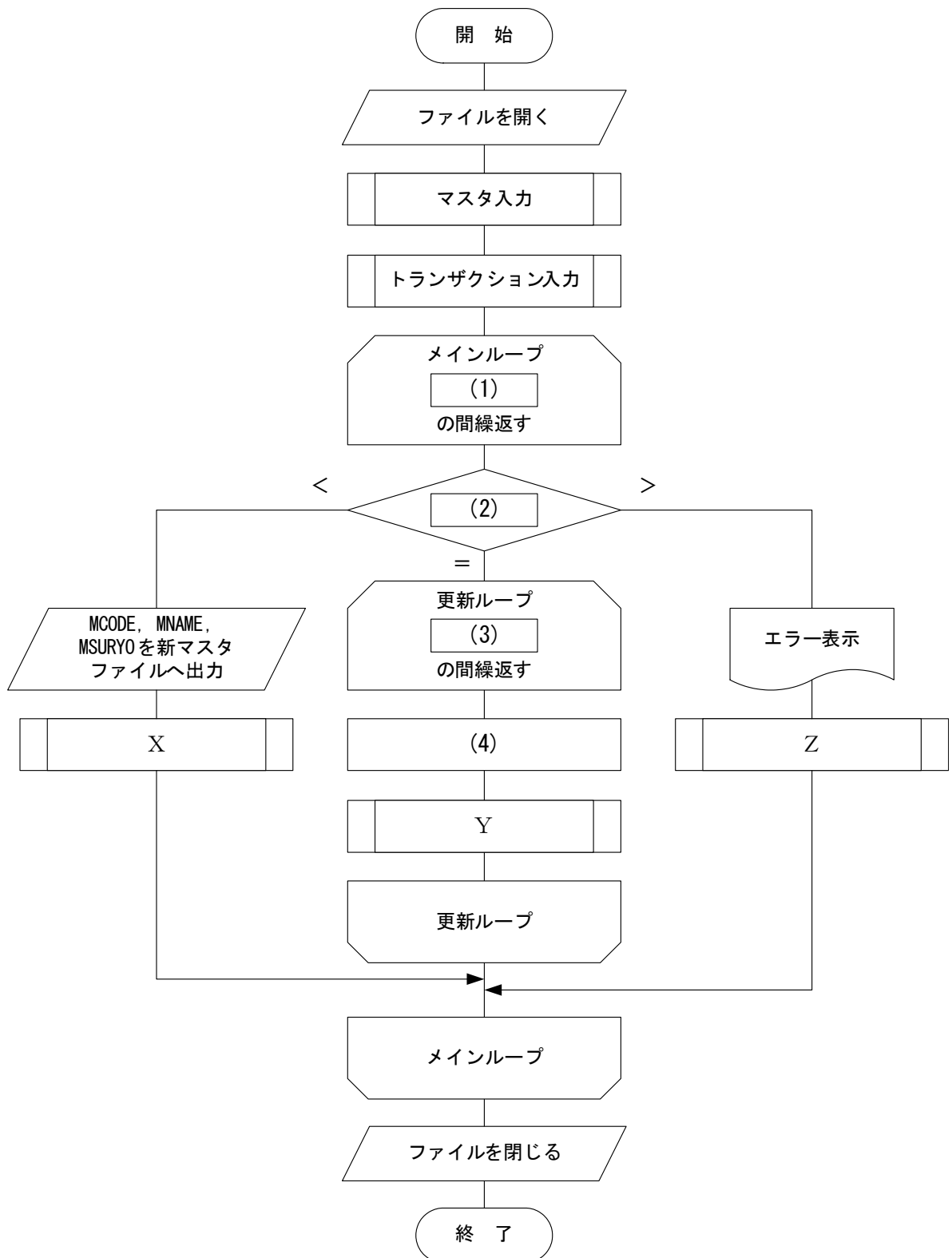
2. トランザクションファイルは順編成ファイルである。レコード形式は図3のようになっており、商品コードの昇順に整列されている。なお、商品コードが同じ値のレコードが複数存在する可能性がある。

商品コード	数量
TCODE	TSURYO

図3 トランザクションファイルのレコード形式

3. マスタファイルの商品コードとトランザクションファイルの商品コードの値が同じであれば、トランザクションファイルの数量をマスタファイルの数量に加えて新マスタファイルへ出力する。
4. マスタファイルの商品コードがトランザクションファイルに存在しない場合は、マスタファイルのレコードをそのまま新マスタファイルへ出力する。
5. トランザクションファイルの商品コードがマスタファイルに存在しない場合は、エラー表示をする。
6. 流れ図中で用いる“最高値”はシステムで表現できる最大の値を意味し、商品コードに使用されていない。

[流れ図]



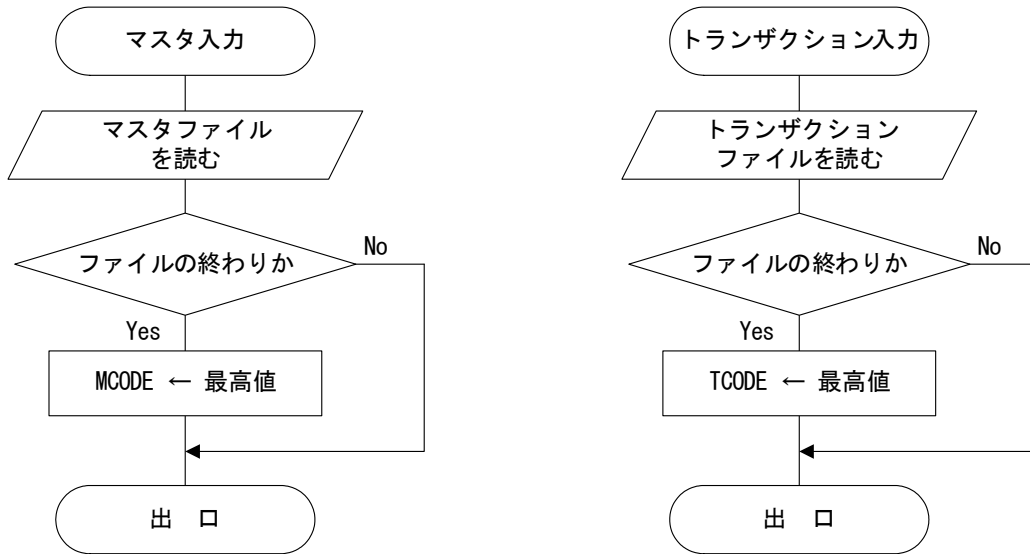


図4 ファイル更新の流れ図

<設問1> 流れ図中の に入るべき適切な字句を解答群から選べ。

(1) の解答群

- ア. MCODE = 最高値 かつ TCODE = 最高値
- イ. MCODE ≠ 最高値 かつ TCODE ≠ 最高値
- ウ. MCODE = 最高値 または TCODE = 最高値
- エ. MCODE ≠ 最高値 または TCODE ≠ 最高値

(2) の解答群

- ア. MCODE : TCODE
- イ. TCODE : MCODE
- ウ. MCODE : 最高値
- エ. TCODE : 最高値

(3) の解答群

- ア. MCODE = TCODE
- イ. MCODE ≠ TCODE
- ウ. MCODE > TCODE
- エ. MCODE < TCODE

(4) の解答群

- ア. MSURYO ← 0
- イ. MSURYO ← TSURYO
- ウ. MSURYO ← MSURYO + TSURYO
- エ. MSURYO ← MSURYO - TSURYO

<設問 2> 流れ図中の X, Y, Z に入るべき適切な処理の組合せを解答群から選べ。

(5) の解答群

	X	Y	Z
ア	マスタ入力	マスタ入力	マスタ入力
イ	マスタ入力	トランザクション入力	トランザクション入力
ウ	トランザクション入力	トランザクション入力	マスタ入力
エ	トランザクション入力	マスタ入力	トランザクション入力

問題3 次の文字列処理に関する説明を読み、流れ図中の に入れるべき適切な字句を解答群から選べ。

[文字列処理の説明]

配列 buff は N 文字を格納する配列であり、この配列中の任意の部分に英単語が 1 つ格納されている。なお、配列 buff の要素位置は、0 から始まる。

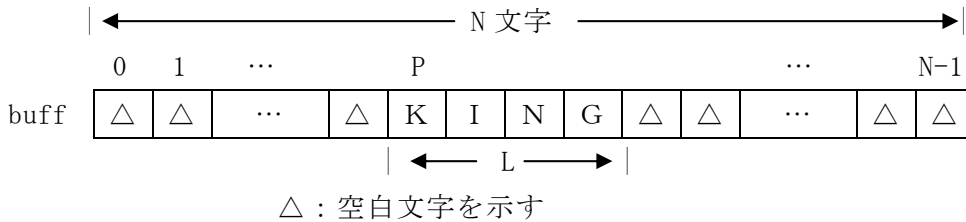


図1 buff に格納される内容

- 英単語中に空白文字は含まない。
- P は、英単語の先頭文字が格納されている配列 buff の要素位置を示す。
- L は、英単語の文字数を示す。
- N, P, L の関係は次のとおりである。
 $1 < L \leq N, 0 \leq P < N$

配列 buff に格納された英単語について、次の 3 つの処理を考える。

1. 左詰め処理

入力された文字列を左詰めにする処理である。

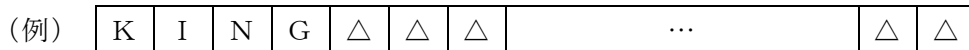


図2 左詰めされた buff の内容

2. 右詰め処理

入力された文字列を右詰めにする処理である。

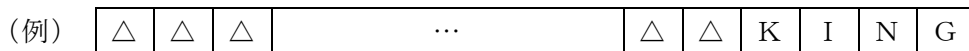


図3 右詰めされた buff の内容

3. 逆順処理

配列 buff に格納されている英単語の位置を変更せずに、逆の順番に入れ替える処理である。

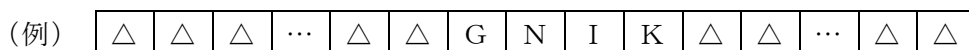


図4 逆順処理された buff の内容

なお、流れ図中で行われる除算は、小数点以下を切り捨てるものとする。

[左詰め処理の流れ図]

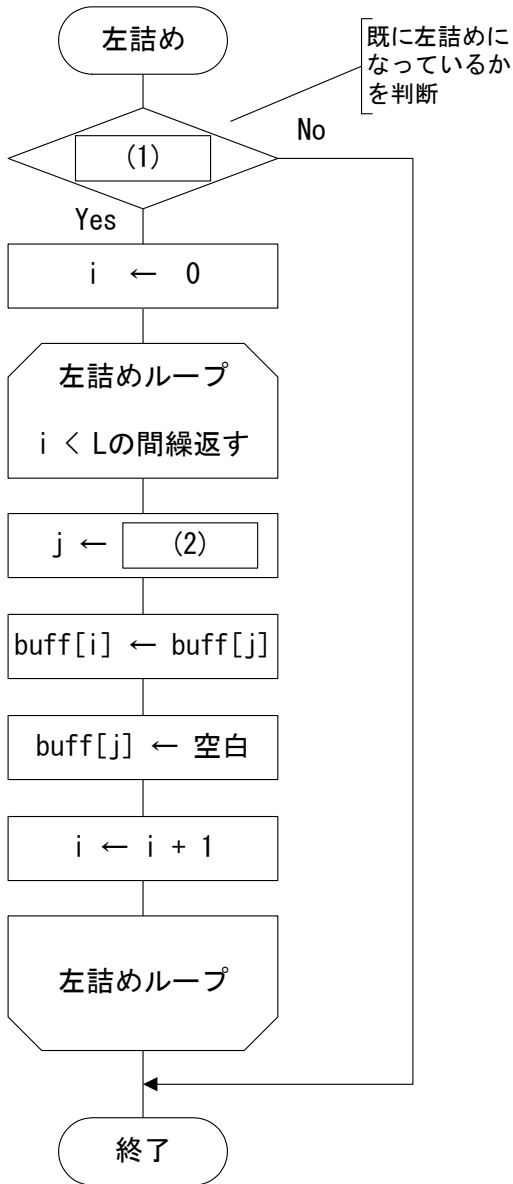


図5 左詰め処理

[右詰め処理の流れ図]

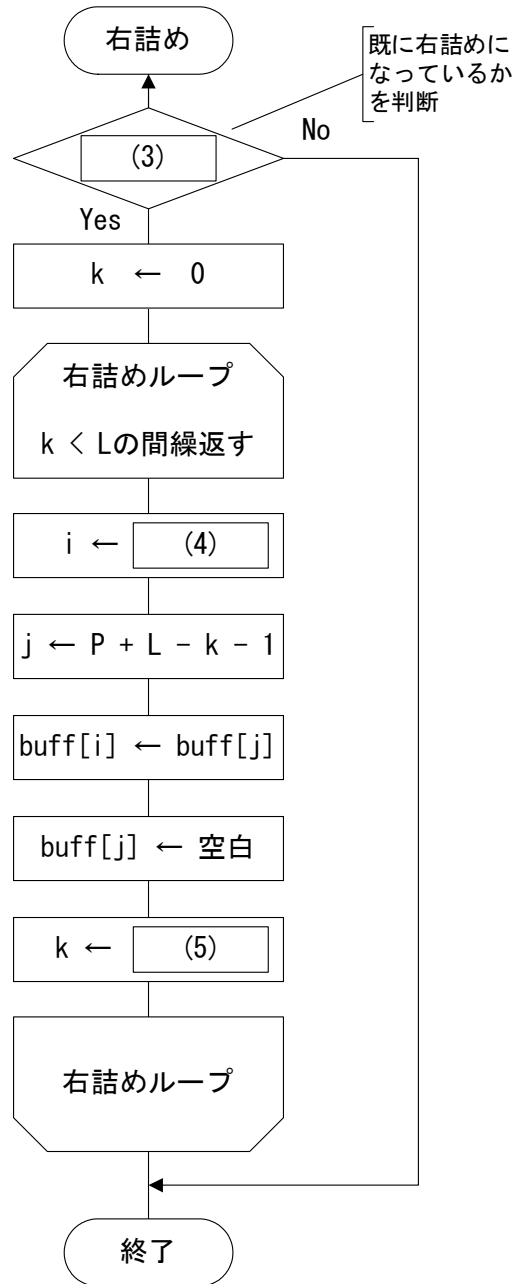


図6 右詰め処理

(1) , (3) の解答群

ア. $P > 0$

エ. $P \leq L$

イ. $P > L$

オ. $P \leq N$

ウ. $P > N$

カ. $P + L < N$

(2) , (4) , (5) の解答群

ア. $k + 1$

エ. $P + k$

イ. $k - 1$

オ. $N - i$

ウ. $P + i$

カ. $N - k - 1$

[逆順処理の流れ図]

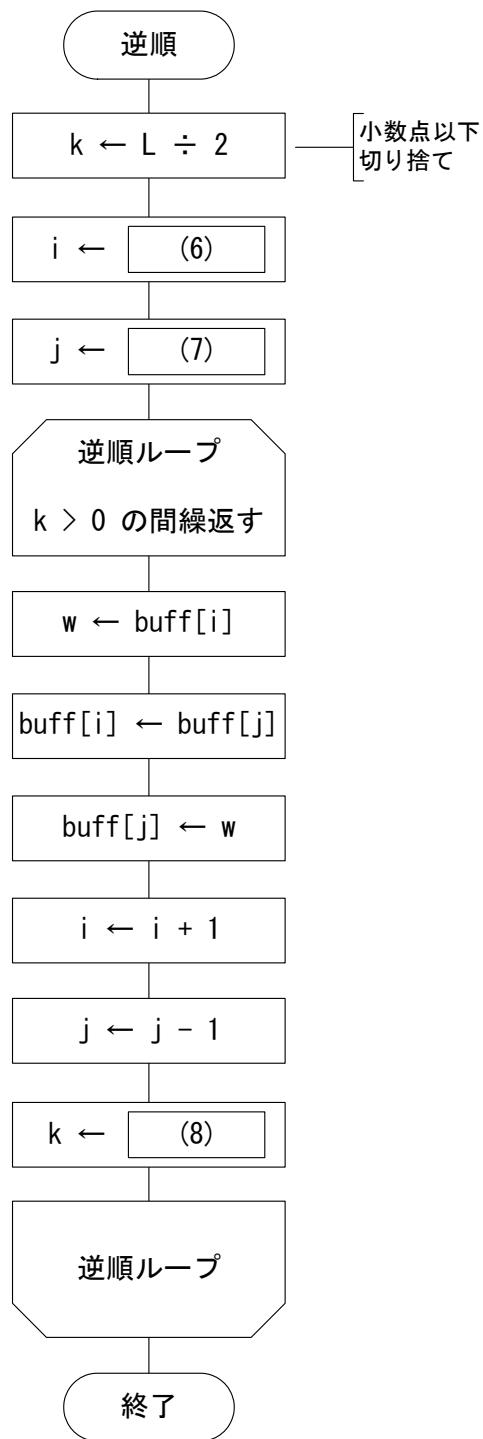


図7 逆順処理

(6) ~ (8) の解答群

ア. $k + 1$

エ. $N - 1$

イ. $k - 1$

オ. P

ウ. N

カ. $P + L - 1$

問題4 次のプログラムの説明および疑似言語の記述形式の説明を読み、設問に答えよ。

[プログラムの説明]

1次元配列 dat の中に格納された数値データを昇順に並び換える関数 SORT である。

1次元配列 dat の要素数は N とし、要素位置は 0 から始まる。dat[0]~dat[N-1]に数値データが格納されている。

	0	1	2	...	N-2	N-1
dat	121	98	180	...	2	86

図1 配列 dat

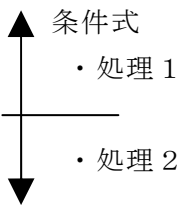
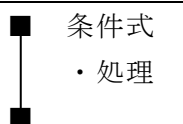
[並び換えの方法]

dat[i]~dat[N-1]の中で一番小さい値が格納されている位置を k に求め、dat[i]と dat[k]を交換する。

この i を 0~N-2 まで変化させることで並び換えが完了する。

変数名	入力/出力	意味
dat	入力/出力	データを格納する 1次元配列
N	入力	1次元配列 dat の要素数

[疑似言語の記述形式の説明]

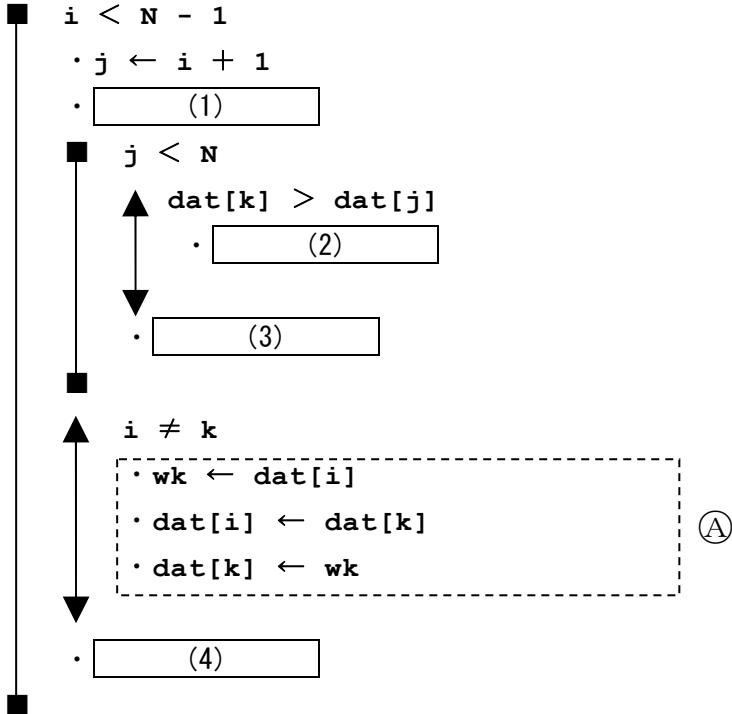
記述形式	説明
○	手続き、変数などの名前、型などを宣言する
・変数 ← 式	変数に式の値を代入する
{文}	注釈を記述する
	選択処理を示す。 条件式が真の時は処理 1 を実行し、 偽の時は処理 2 を実行する。
	前判定繰り返し処理を示す。 条件式が真の間、処理を実行する。

[プログラム]

○SORT (整数型: **dat**, 整数型: **N**)

○整数型: **i**, **j**, **k**, **wk**

・ **i** ← 0



<設問 1> プログラム中の に入れるべき適切な字句を解答群から選べ。

(1), (2) の解答群

ア. **k** ← 0

ウ. **k** ← **i**

オ. **k** ← **i** + 1

イ. **k** ← 1

エ. **k** ← **j**

カ. **k** ← **j** + 1

(3), (4) の解答群

ア. **i** ← **i** + 1

ウ. **j** ← **j** + 1

オ. **k** ← **k** + 1

イ. **i** ← **i** + **k**

エ. **j** ← **j** + **k**

カ. **k** ← **k** + **i**

<設問 2 > 1次元配列 dat に図 2 のようなデータを格納して SORT を呼び出した場合、プログラム中の㉠の部分を実行し終えたとき、dat[0]に格納される値を解答群から選べ。

	N=5				
	0	1	2	3	4
dat	28	12	8	19	6

図 2 1次元配列 dat に格納したデータ

(5) の解答群

ア. 6

イ. 8

ウ. 12

エ. 19

これより

< 選 択 問 題 >

選択問題は次の問題から 1 問選択し解答せよ。

選択した問題は必ず、解答用紙「選択欄」にマークすること。

※選択欄にマークがなく、解答のみの場合は採点を行いません。

各構成は以下のとおり。

選択問題

- | | |
|------------|---------------|
| ・ C 言語の問題 | 15 ページ～18 ページ |
| ・ 表計算の問題 | 19 ページ～23 ページ |
| ・ アセンブラの問題 | 24 ページ～27 ページ |

選択問題 C言語の問題

次のC言語プログラムの説明を読み、プログラム中の に入れるべき適切な式を解答群から選べ。

[プログラムの説明]

ビット列をBASE64で変換し、その文字列を出力する関数Encodeである。

BASE64とは、データを印字可能な64種類の英数字のみを用いて表し、それ以外の文字を扱うことの出来ない通信環境においてマルチバイト文字やバイナリデータを扱うためのエンコード方式である。

BASE64への変換は、1文字を8ビットで表した3文字を1つの単位とし、この24ビットを6ビットずつ区切って変換表から対応する4文字を出力する。

表 変換表

	2進数	文字		2進数	文字		2進数	文字		2進数	文字
0	000000	A	16	010000	Q	32	100000	g	48	110000	w
1	000001	B	17	010001	R	33	100001	h	49	110001	x
2	000010	C	18	010010	S	34	100010	i	50	110010	y
3	000011	D	19	010011	T	35	100011	j	51	110011	z
4	000100	E	20	010100	U	36	100100	k	52	110100	0
5	000101	F	21	010101	V	37	100101	l	53	110101	1
6	000110	G	22	010110	W	38	100110	m	54	110110	2
7	000111	H	23	010111	X	39	100111	n	55	110111	3
8	001000	I	24	011000	Y	40	101000	o	56	111000	4
9	001001	J	25	011001	Z	41	101001	p	57	111001	5
10	001010	K	26	011010	a	42	101010	q	58	111010	6
11	001011	L	27	011011	b	43	101011	r	59	111011	7
12	001100	M	28	011100	c	44	101100	s	60	111100	8
13	001101	N	29	011101	d	45	101101	t	61	111101	9
14	001110	O	30	011110	e	46	101110	u	62	111110	+
15	001111	P	31	011111	f	47	101111	v	63	111111	/

具体的には、次のように6ビットずつ取出す。

<変換するデータの取出し位置>

- ① 1文字目の上位6ビット
- ② 1文字目の下位2ビットと2文字目の上位4ビット
- ③ 2文字目の下位4ビットと3文字目の上位2ビット
- ④ 3文字目の下位6ビット

なお、ビット列の文字数が3の倍数でなければ、<変換するデータの取出し位置>

②の2文字目, または③の3文字目が存在しない。この場合は, 0を補って変換して終了する。

また, 出力する文字は4文字ずつになるが, ビット数により4文字に満たない場合がある。その場合は, '='を補って4文字にする。

ここでは, 1行に出力する文字数を最大76文字とする。

<変換例>

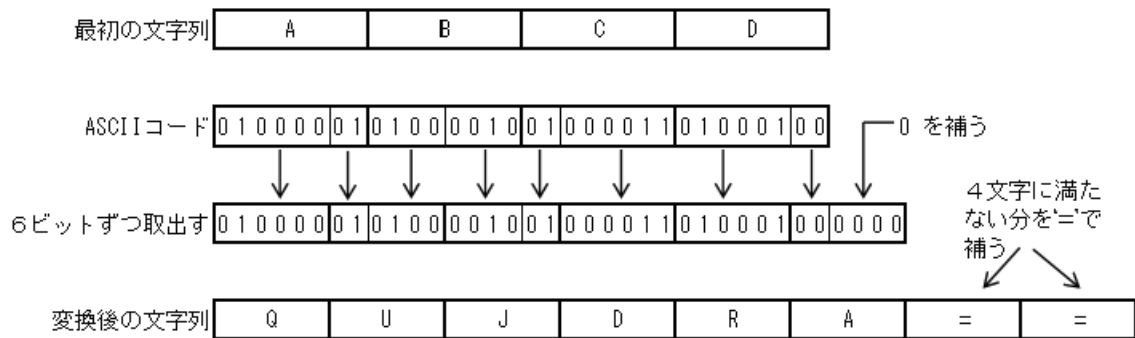


図1 変換例

[関数の説明]

Encode 関数

引 数：文字列。

機 能：BASE64で変換した文字列を出力する。

戻り値：なし。

OutputCheck 関数

引 数：追加する文字の格納位置, 出力文字列, 追加する文字

機 能：出力文字列に文字を追加する。出力文字列の文字数が1行の最大文字数以上になれば, 出力文字列を出力する。

戻り値：次に追加する文字の格納位置

[配列b64tblについて]

変換表にある文字を順番に格納しており, 関数 Encode が呼び出される前に配列内に設定される。

	0	1	2	3	...	61	62	63
b64tbl	A	B	C	D	...	9	+	/

[プログラム]

```
#include <stdio.h>
#define MAXCHARS 76
unsigned char b64tbl[64];
void Encode(unsigned char *);
unsigned int OutputCheck(unsigned int,
```

```

                unsigned char *,
                unsigned char);

void Encode(unsigned char *buff) {
    unsigned int    i, j, scnt;
    unsigned char  n, out_tbl[MAXCHARS+1];
    i = 0;          /* 受取った文字列の添え字 */
    j = 0;          /* 出力文字列の添え字 */
    scnt = 0;      /* 受取った文字列の取出し位置を判断するカウンタ */
    while( (1) ) { /* 受取った文字列の最後まで繰り返す */
        switch (scnt) {
            case 0:
                n = ( buff[i] & 0xfc ) >> 2;
                break;
            case 1:
                n = ( (2) ) | ( (buff[i+1] & 0xf0) >> 4 );
                i++;
                break;
            case 2:
                n = ( (3) ) | ( (buff[i+1] & 0xc0) >> 6 );
                i++;
                break;
            case 3:
                n = (4) ;
                i++;
        }
        (5) ;
        scnt = (scnt + 1) (6) ;
    }
    /* 出力する4文字に満たない部分に"="を設定する */
    if (scnt > 0) for( i=scnt; i<4; i++) (7) ;
    /* 出力しきれていない文字列が存在する場合に出力する */
    if (j > 0) {
        out_tbl[j] = '¥0';
        puts(out_tbl);
    }
}

```

```

unsigned int OutputCheck(unsigned int n,
                          unsigned char *buff,
                          unsigned char ch) {
    buff[n] = ch;
    n++;
    if (n >= MAXCHARS) { /* 最大出力文字数に到達した場合は出力する */
        buff[n] = '\0';
        puts(buff);
        n = 0;
    }
    (8) ;
}

```

(1) の解答群

- | | |
|---------------------------------|---------------------------------|
| ア. <code>buff[i] == '\0'</code> | イ. <code>buff[i] != '\0'</code> |
| ウ. <code>i < MAXCHARS</code> | エ. <code>i <= j</code> |

(2) ~ (4) の解答群

- | | |
|---|---|
| ア. <code>buff[i] & 0x3f</code> | イ. <code>buff[i] & 0xc0</code> |
| ウ. <code>(buff[i] & 0xf0) >> 2</code> | エ. <code>(buff[i] & 0xfc) >> 2</code> |
| オ. <code>(buff[i] & 0x0f) << 2</code> | カ. <code>(buff[i] & 0xf0) << 2</code> |
| キ. <code>(buff[i] & 0x03) << 4</code> | ク. <code>(buff[i] & 0x0f) << 4</code> |
| ケ. <code>(buff[i] & 0x03) << 6</code> | コ. <code>(buff[i] & 0xc0) >> 6</code> |

(5) , (7) の解答群

- ア. `j = OutputCheck(j, out_tbl, '0')`
- イ. `j = OutputCheck(j, out_tbl, '=')`
- ウ. `j = OutputCheck(j, out_tbl, n)`
- エ. `j = OutputCheck(j, out_tbl, b64tbl[i])`
- オ. `j = OutputCheck(j, out_tbl, b64tbl[j])`
- カ. `j = OutputCheck(j, out_tbl, b64tbl[n])`

(6) の解答群

- | | | | |
|---------------------|---------------------|---------------------|---------------------|
| ア. <code>% 2</code> | イ. <code>/ 2</code> | ウ. <code>% 4</code> | エ. <code>/ 4</code> |
|---------------------|---------------------|---------------------|---------------------|

(8) の解答群

- | | |
|--------------------------|------------------------------|
| ア. <code>return 0</code> | イ. <code>return 1</code> |
| ウ. <code>return n</code> | エ. <code>return n + 1</code> |

次の表計算ソフトに関する設問に答えよ。

この問題で使用する表計算ソフトの仕様は下記のとおりである

SUMIF 関数

指定された範囲に含まれるセルのうち、検索条件に一致するセルに対応する合計範囲の合計を求める。なお、合計範囲は省略可能であり、省略された場合は範囲の合計を求める。

書式：SUMIF(範囲, 検索条件, [合計範囲])

IF 関数

条件が真の時に 真の場合 , 偽の時に 偽の場合 の計算結果や値を返す。

IF 関数の中に IF 関数を入れることができる。

書式：IF (条件式 , 真の場合 , 偽の場合)

VLOOKUP 関数

検査値と検査範囲のセルの値 (範囲値) とを比較し、「範囲値=検査値」となるときの行に対し、列番号で指定された列内にある値を返す。

書式：VLOOKUP (検査値, 検査範囲, 列番号)

検査値 : 検査の値または対象セル番地

検査範囲 : 検査開始セル番地 : 検査終了セル番地

列番号 : 検査範囲の列を 1 として数えたときの列の番号

RANK 関数

順序に従って範囲内の数値を並べ替えたとき、数値が何番目に位置するかを返す。

順位は 0 を指定すると降順, 1 を指定すると昇順になる。

書式：RANK (数値, 範囲, 順位)

式

=に続けて計算式や関数などを入力する。

セル番地の参照

セル番地に \$ を付けることで、絶対番地 (絶対参照) を表す。

J商店では、10種類の商品を扱い販売している。商品の売上は1日ごとに商品別に集計し売上表に入力している。9月の売上表は表1のようになっている。また、表はすべて同一シート上に作成される。

表1 売上表

	A	B	C	D
1				
2		日付	商品コード	売上数量
3		9月1日	C001	15
4			C002	8
5			C006	12
6			C007	4
7			C008	3
8			C009	16
9		9月2日	C002	5
10			C003	6
11			C010	2
		⋮	⋮	⋮
199		9月30日	C001	3
200			C002	16
201			C005	6
202			C007	20

<設問1> 売上表（表1）から売上分析表（表2）を作成する。売上分析表にはあらかじめ商品コード、商品名、単価が入力されており、各商品の月間売上数量、売上金額、売上順位を求める。以下の問に答えよ。

表2 売上分析表

	F	G	H	I	J	K
1						
2	商品コード	商品名	単価	月間売上数量	売上金額	売上順位
3	C001	商品P	1,000	246	246,000	5
4	C002	商品Q	1,200	346	415,200	1
5	C003	商品R	1,500	152	228,000	6
6	C004	商品S	800	168	134,400	8
7	C005	商品T	300	439	131,700	9
8	C006	商品U	2,500	166	415,000	2
9	C007	商品V	3,100	104	322,400	3
10	C008	商品W	2,200	131	288,200	4
11	C009	商品X	900	71	63,900	10
12	C010	商品Y	1,700	117	198,900	7
13	合計				2,443,700	

(1) 売上分析表の I 列に商品ごとの月間売上数量を求めるため、I3 に入力する式を解答群から選べ。ただし、I3 の式は I4～I12 に複写する。

(1) の解答群

- ア. =SUMIF(C\$3:C\$202, F3, D\$3:D\$202)
- イ. =SUMIF(\$C3:\$C202, G3, D\$3:D\$202)
- ウ. =SUMIF(D\$3:C\$202, F3, C\$3:C\$202)
- エ. =SUMIF(\$D3:\$D\$202, G3, C\$3:C\$202)

(2) 売上分析表の J 列に商品ごとの売上金額を求め、売上金額の順位（降順）を求めるため K3 に入力する式を解答群から選べ。ただし、K3 の式は K4～K12 に複写する。

(2) の解答群

- ア. =RANK(\$J3, \$J3:\$J12, 0)
- イ. =RANK(J3, I\$3:I\$12, 0)
- ウ. =RANK(J\$3, J\$3:J\$12, 0)
- エ. =RANK(J3, J\$3:J\$12, 0)

<設問 2> 各商品から生じる利益を求めるため、各商品の利益率を調査し利益率表を作成した。各商品の利益は売上金額から消費税額を差し引き、その結果に利益率をかけて求める。以下の問に答えよ。

表 3 利益率表

	F	G	H
14			
15	消費税	0.05	
16			
17	商品コード	商品名	利益率
18	C001	商品P	0.10
19	C002	商品Q	0.12
20	C003	商品R	0.15
21	C004	商品S	0.07
22	C005	商品T	0.20
23	C006	商品U	0.18
24	C007	商品V	0.05
25	C008	商品W	0.10
26	C009	商品X	0.12
27	C010	商品Y	0.15

表4 売上分析表

	F	G	H	I	J	K	L
1							
2	商品コード	商品名	単価	月間売上数量	売上金額	売上順位	利益
3	C002	商品Q	1,200	346	415,200	1	47,333
4	C006	商品U	2,500	146	365,000	2	62,415
5	C007	商品V	3,100	104	322,400	3	15,314
6	C008	商品W	2,200	131	288,200	4	27,379
7	C001	商品P	1,000	231	231,000	5	21,945
8	C003	商品R	1,500	149	223,500	6	31,849
9	C010	商品Y	1,700	117	198,900	7	28,343
10	C004	商品S	800	168	134,400	8	8,938
11	C005	商品T	300	439	131,700	9	25,023
12	C009	商品X	900	66	59,400	10	6,772
13	合計				2,369,700		

(3) 表4は表2の売上分析表を売上順で並べ替え、利益を求める列を追加した表である。売上分析表のL列に商品ごとの利益を求めるためL3に入力する式を解答群から選べ。ただし、L3の式はL4~L12に複写する。

(3) の解答群

- ア. $= (1 + G15) * J3 * VLOOKUP (F3, F18:H27, 3)$
- イ. $= (1 + G15) * J3 * VLOOKUP (F3, F18:H27, 3)$
- ウ. $= (1 - G15) * J3 * VLOOKUP (F3, F18:H27, 3)$
- エ. $= (1 - G15) * J3 * VLOOKUP (F3, F18:H27, 3)$

<設問3> 表5は表を利益の降順に並べかえ、利益構成比率と利益構成比率累計を求め、次のように商品をグループ化した表である。

- ・Aグループ…利益の累積構成比率が70%未満の商品
- ・Bグループ…利益の累積構成比率が70%以上90%未満の商品
- ・Cグループ…利益の累積構成比率が90%以上の商品

表5 売上分析表

	F	G	H	I	J	K	L	M	N	O
1										
2	商品コード	商品名	単価	月間売上数量	売上金額	売上順位	利益	利益構成比率	利益構成比率累計	グループ
3	C006	商品U	2,500	146	365,000	2	62,415	23%	23%	A
4	C002	商品Q	1,200	346	415,200	1	47,333	17%	40%	A
5	C003	商品R	1,500	149	223,500	6	31,849	12%	51%	A
6	C010	商品Y	1,700	117	198,900	7	28,343	10%	62%	A
7	C008	商品W	2,200	131	288,200	4	27,379	10%	72%	B
8	C005	商品T	300	439	131,700	9	25,023	9%	81%	B
9	C001	商品P	1,000	231	231,000	5	21,945	8%	89%	B
10	C007	商品V	3,100	104	322,400	3	15,314	6%	94%	C
11	C004	商品S	800	168	134,400	8	8,938	3%	98%	C
12	C009	商品X	900	66	59,400	10	6,772	2%	100%	C
13	合計				2,369,700		275,310	100%		

(4) L3 に利益の合計を求めた後、利益の構成比率を求めるため M3 に入力する式を解答群から選べ。ただし、M3 の式は M4～M12 に複写する。

(4) の解答群

ア. =L3/L\$13 イ. =L\$3/L\$13 ウ. =L\$13/L3 エ. =L3/L13

(5) 売上構成比率累計を求めるため、M3 を N3 に複写した後、N4 に入力する式を解答群から選べ。ただし、N4 の式は N5～N12 に複写する。

(5) の解答群

ア. =M3+M4 イ. =N3+M4 ウ. =N3+M3 エ. =N\$3+M\$4

(6) 表 5 で、商品のグループを O 列に表示するためセル O3 に入力する式を解答群から選べ。ただし、O3 の式は O4～O12 に複写する。

(6) の解答群

ア. =IF(N3>0.7, "A", IF(N3>0.9, "B", "C"))
イ. =IF(N3>0.7, "A", IF(N3<0.9, "B", "C"))
ウ. =IF(N3<0.7, "A", IF(N3<0.9, "C", "B"))
エ. =IF(N3<0.7, "A", IF(N3<0.9, "B", "C"))

<設問 4> 表 5 の売上分析表から推測できることを (7) の解答群から選べ。

(7) の解答群

ア. 売上金額の順位と利益の順位は同じである。
イ. 売上金額の順位と利益率の順位は同じである。
ウ. 売上数量が多いほど利益率は高い。
エ. 売上金額の順位と利益構成比率の順位とは異なる。

選択問題 アセンブラの問題

次のアセンブラ言語CASL II プログラムの説明を読み、各設問に答えよ。

[プログラムの説明]

リスト構造から目的のデータを探索する副プログラム SUBR である。

SUBR は、図 1 のような形式でパラメタが格納された先頭番地を GR1 に設定して呼び出される。

アドレス	
(GR1) + 0	探索する値
+ 1	リスト構造の先頭番地

注 探索する値は 0~255 の値である

図 1 パラメタの構造

リスト構造は、パラメタの 2 つ目に格納された番地から始まる連続した領域に格納されており、1 語の上位 8 ビットにデータ、下位 8 ビットにポインタを格納している (図 2)。

上位 8 ビット	下位 8 ビット
データ	ポインタ

図 2 リスト構造の 1 語の形式

ポインタの値は、リスト構造が格納されている連続した領域内の相対位置である。なお、連続した領域は 255 語以内とし、リスト構造の最後のデータのポインタは 0 である。

また、リスト構造の先頭には、最初のデータが格納されているポインタを格納しており、このデータの上位 8 ビットはすべて 0 である (図 3)。

リスト構造の先頭番地 + 0	00000000	最初のデータへのポインタ
+ 1	データ	ポインタ
+ 2	データ	ポインタ
⋮	⋮	⋮

図 3 リスト構造

副プログラム SUBR は、探索する値がリスト構造のデータに存在すれば 1、存在しなければ 0 を GR0 に設定して戻る。

[プログラム]

行番号	ラベル	命令	オペランド	コメント
100	SUBR	START		
110		RPUSH		
120		LD	GR0,=0	; 見つからない場合の戻り値を設定
130		LD	GR2,1,GR1	; リスト構造の先頭番地を取り出す
140			(1)	; 最初のデータのポインタを取出す
150	LOOP	ADDL	GR3,1,GR1	; 相対番地から絶対番地へ変換
160		LD	GR4,0,GR3	; リスト構造からの取出し
170		SRL	GR4,8	; データの取出し
180		CPA	GR4,0,GR1	; データの比較
190			(2)	
200		LD	GR3,0,GR3	; リスト構造からの取出し
210			(3)	; ポインタの取出し
220		JZE	FINISH	
230		JUMP	LOOP	
240	FIN	LD	GR0,=1	; 見つかった場合の値を設定
250	FINISH	RPOP		
260		RET		
270		END		

<設問1> プログラム中の に入れるべき適切な命令を解答群から選べ。

(1) の解答群

- | | |
|-----------------|-----------------|
| ア. LD GR3,GR1 | イ. LD GR3,GR2 |
| ウ. LD GR3,1,GR1 | エ. LD GR3,0,GR2 |

(2) の解答群

- | | |
|------------|---------------|
| ア. JZE FIN | イ. JZE FINISH |
| ウ. JNZ FIN | エ. JNZ FINISH |

(3) の解答群

- | | |
|-------------------|-------------------|
| ア. SLA GR3,8 | イ. SLL GR3,8 |
| ウ. AND GR3,=#00FF | エ. XOR GR3,=#FF00 |

<設問 2 > 行番号 150 のオペランドを置き換えた場合、実行結果が等しくなるものを解答群から選べ。

(4) の解答群

ア. GR3, 0, GR2

イ. GR3, 0, GR1

ウ. GR3, 1, GR2

エ. GR3, GR2

<設問 3 > この副プログラムの改造に関する記述を読み、プログラム中の に入れるべき適切な命令を解答群から選べ。

探索するデータが存在した場合、そのデータをリスト構造から削除するように変更する。

リスト構造からデータを削除するには、ポインタを変更すればよい。

具体的には、削除しようとするデータが格納されている語のポインタを、削除する 1 つ前のデータが格納されている語のポインタに設定すればよい。

まず、1 つ前のポインタを GR5 に保持するため、次の命令を 140 行と 150 行の間および、190 行と 200 行の間に追加する。

行番号	ラベル	命令	オペランド	コメント
145		LD	GR5, 1, GR1	; 1 つ前のポインタを保持 (絶対番地)
195		LD	GR5, GR3	; 1 つ前のポインタを保持 (絶対番地)

見つけた場合に削除するように、240 行と 250 行の間に次の命令を追加する。

行番号	ラベル	命令	オペランド	コメント
241		<input type="text"/>	(5)	; 1 つ前の語を取出す
242		AND	GR4, =#FF00	; ポインタの部分を 0 クリア
243		<input type="text"/>	(6)	; 削除する語を取出す
244		AND	GR6, =#00FF	; ポインタの部分を取出す
245		OR	GR4, GR6	; 合成
246		<input type="text"/>	(7)	; 格納

(5) の解答群

ア. LD GR4, 0, GR3

イ. LD GR4, 0, GR5

ウ. LD GR4, 1, GR3

エ. LD GR4, 1, GR5

(6) の解答群

ア. LD GR6,0,GR3

ウ. LD GR6,1,GR3

イ. LD GR6,0,GR5

エ. LD GR6,1,GR5

(7) の解答群

ア. ST GR4,0,GR3

ウ. ST GR6,0,GR3

イ. ST GR4,0,GR5

エ. ST GR6,0,GR5

<メモ欄>

<メモ欄>

