

平成21年度後期 情報検定

<実施 平成22年2月14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）、ポケットベル、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38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. 合否通知の発送は平成22年3月中旬の予定です。
 - ①団体受験された方は、団体経由で合否の通知をいたします。
 - ②個人受験の方は、受験票に記載されている住所に郵送で合否の通知をいたします。
 - ③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 3	2 ページ～13 ページ
-------------------------	--------------

選択問題 A 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	20 ページ～22 ページ
・ アセンブラの問題	23 ページ～25 ページ

選択問題 B 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	26 ページ～30 ページ
・ 表計算の問題	31 ページ～34 ページ
・ アセンブラの問題	35 ページ～38 ページ

必須問題

問題 1 次のバブルソート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。

[バブルソート法の説明]

バブルソート法とは隣り合う2つの要素を比較し、並び替えの規則にしたがって要素を交換して並び替えを行うものである。

例えば、5個の要素からなる配列を昇順に並び替える場合、一巡目の動作は次の図1のようになる。

図中の矢印は比較する配列の添字を示している。また、網掛け部分はデータを入れ替えたことを示している。

比較交換は配列の最後の位置から始め、その1つ前の要素と大小比較を行う。昇順に並べるので前に位置する要素の値が大きければ要素を交換する。配列の添字を1ずつ小さくして行き配列の先頭まで調べたら一巡目の処理を終える。一巡目の処理で先頭の要素が確定する。

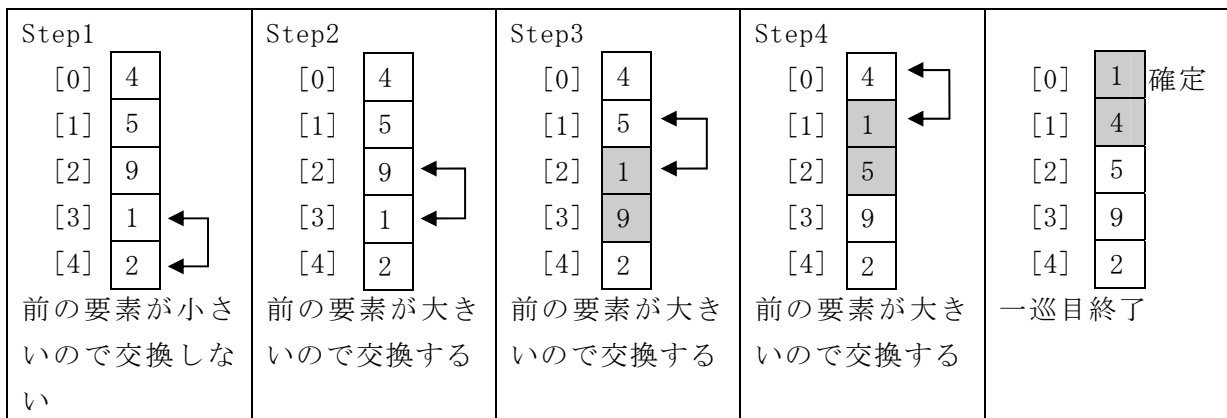


図1 一巡目の繰返し処理

さらに二巡目の繰返し処理を示す。二巡目の繰返しの場合は、先頭の要素が確定しているため、先頭から2つ目の要素まで調べる。

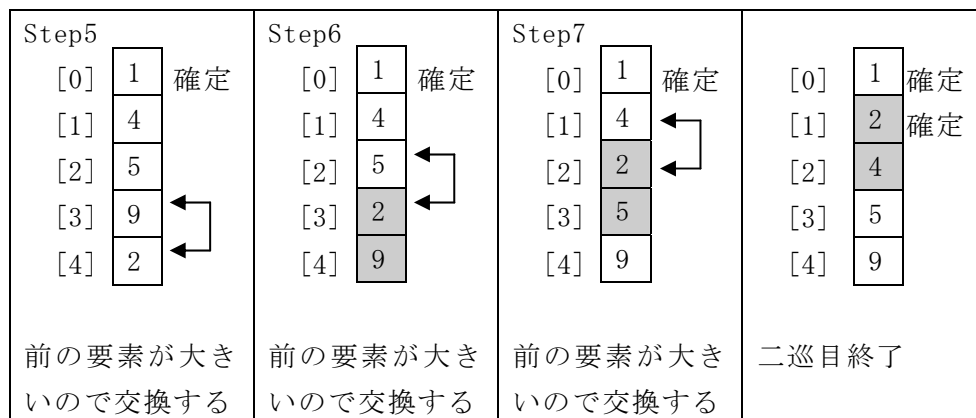


図2 二巡目の繰返し処理

二巡目の繰返し処理で、先頭から二つ目までの要素が確定する。

さらに三巡目の繰返し処理を示す。三巡目の繰返し処理の場合は、先頭から二つ目までの要素が確定しているので、先頭から三つ目の要素まで調べる。

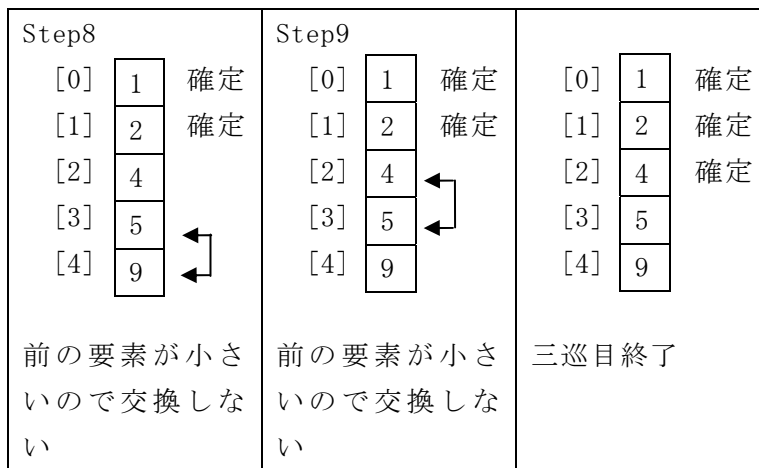


図3 三巡目の繰返し処理

[流れ図の説明]

要素数 n 個の一次元配列 TBL をバブルソート法で昇順に並び替えるものである。バブルソートの処理中にすべての要素が昇順に並んでいることが分かったとき、処理を終了する。なお、配列 TBL の添字は 0 から始まる。

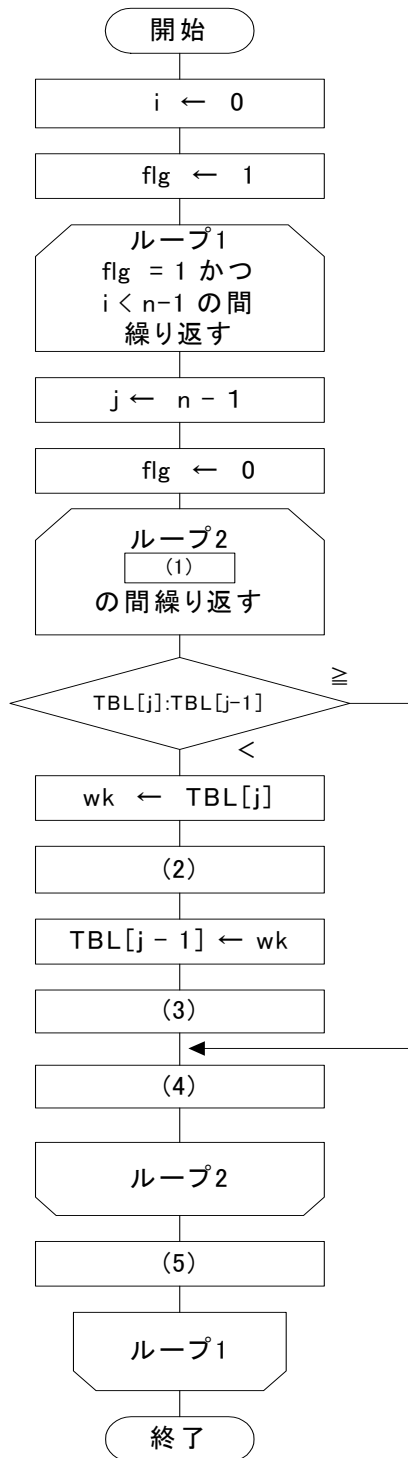


図4 流れ図

(1) の解答群

- | | |
|------------|---------------|
| ア. $i > j$ | イ. $i \geq j$ |
| ウ. $j > i$ | エ. $j \geq i$ |

(2) ~ (5) の解答群

- | | |
|---------------------------------|-------------------------|
| ア. $i \leftarrow i + 1$ | イ. $i \leftarrow i - 1$ |
| ウ. $j \leftarrow j + 1$ | エ. $j \leftarrow j - 1$ |
| オ. $flg \leftarrow 0$ | カ. $flg \leftarrow 1$ |
| キ. $TBL[j-1] \leftarrow TBL[j]$ | |
| ク. $TBL[j] \leftarrow TBL[j-1]$ | |

問題2 次の流れ図の説明を読み、流れ図中の に入れるべき適切な字句を解答群から選べ。

[流れ図の説明]

ある学校で行われたマラソン大会の結果を、学年別・クラス別に集計し印刷する。

測定結果が格納されている測定ファイルは学年コード、クラスコード、出席番号の昇順に整列済みであり、図1の形式になっている。また、印刷形式は、図2のように一人一人の結果を印刷した後に、クラス、学年ごとの平均を印刷し、最後に学校全体の平均を印刷する。

学年 コード	クラス コード	出席番号	名前	測定結果
-----------	------------	------	----	------

図1 レコードの形式

- ・ 学年コードは1から3のいずれかの値が格納されている。
- ・ クラスコードは1から4のいずれかの値が格納されている。
- ・ 出席番号は、同一クラスコード内で一意の値である。
- ・ 測定結果は秒単位での測定時間が格納されており、印刷時に自動的に時：分：秒の形式に変換されるものとする。

1年	1組	1番	相沢秀樹	タイム：2:11:31
1年	1組	2番	伊藤文明	タイム：2:03:51
1年	1組	3番	梅沢直人	タイム：2:13:11
			⋮	
			⋮	
1年	1組	31番	山岸みゆき	タイム：2:33:12
1年	1組		平均タイム	2:11:41
1年	2組	1番	有賀久美	タイム：2:28:21
			⋮	
			⋮	
1年	4組	32番	山田徹	タイム：2:01:31
1年	4組		平均タイム	2:13:16
1年			平均タイム	2:12:41
2年	1組	1番	浅野知美	タイム 2:08:51
			⋮	
			⋮	
3年	4組	35番	渡部あきら	タイム 2:01:31
3年	4組		平均タイム	2:11:03
3年			平均タイム	2:08:31
学校			平均タイム	2:10:12

図2 印刷結果

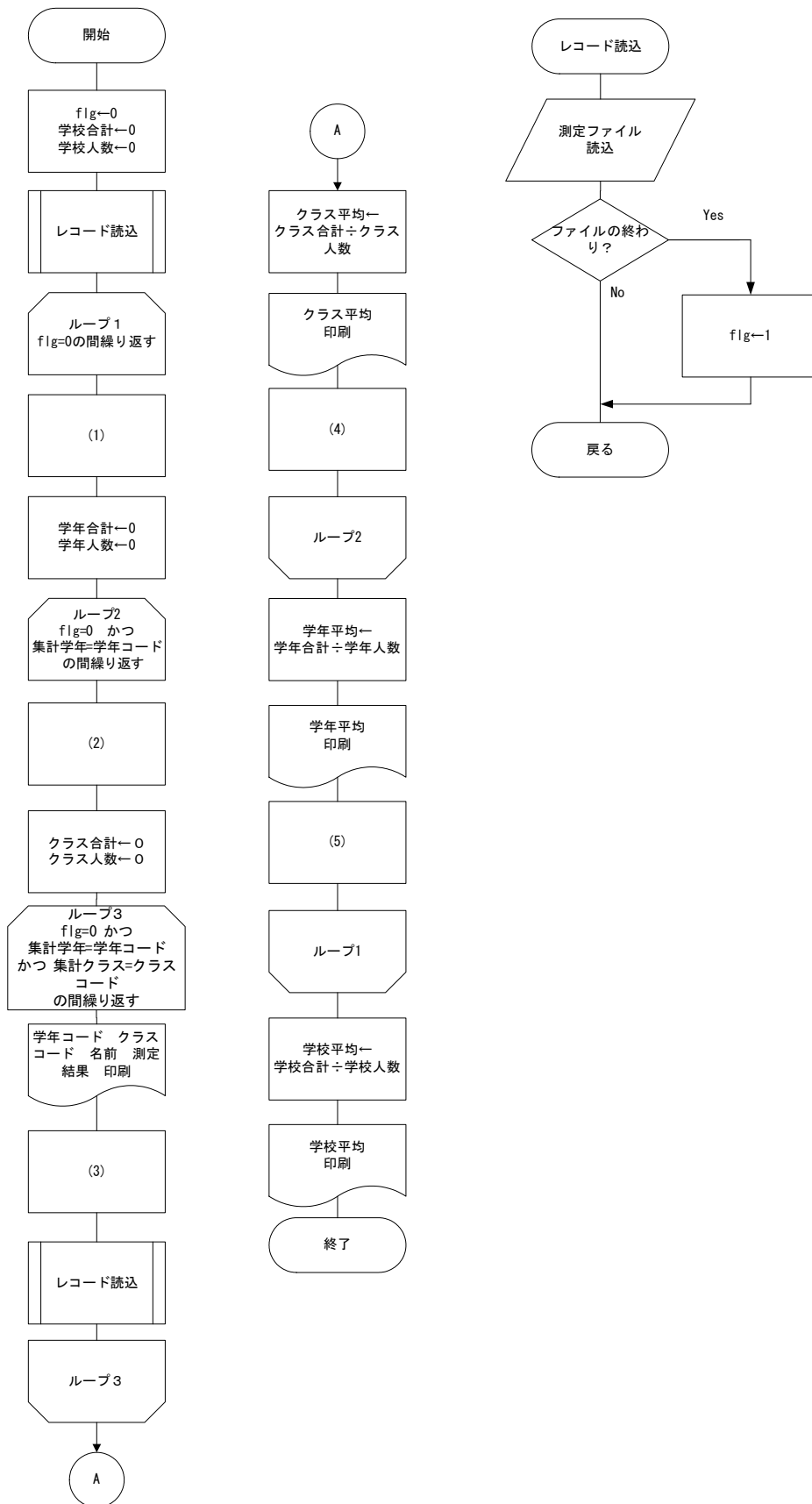


図3 流れ図

(1) , (2) の解答群

- ア. 集計クラス←クラスコード
- ウ. クラスコード←集計クラス
- オ. flg←0 カ. flg←1

- イ. 集計学年←学年コード
- エ. 学年コード←集計学年

(3) ~ (5) の解答群

- ア. クラス合計←クラス合計+測定結果
 クラス人数←クラス人数+1
- イ. クラス合計←クラス合計+測定結果
 クラス人数←クラス人数+クラス人数
- ウ. クラス合計←クラス合計+クラス合計
 クラス人数←クラス人数+1
- エ. 学年合計←学年合計+測定結果
 学年人数←学年人数+1
- オ. 学年合計←学年合計+クラス合計
 学年人数←学年人数+クラス人数
- カ. 学年合計←学年合計+学年合計
 学年人数←学年人数+1
- キ. 学校合計←学校合計+測定結果
 学校人数←学校人数+1
- ク. 学校合計←学校合計+学年合計
 学校人数←学校人数+学年人数
- ケ. 学校合計←学校合計+学校合計
 学校人数←学校人数+1

問題3 次のワイルドカード検索に関する記述を読み、設問に答えよ。

ある文字列の中から特定の文字列を検索する場合、一部だけが分かっているときにワイルドカードを使って表現することがある。ここでは、ワイルドカードを表す文字としてアスタリスク (*) を使用する。ワイルドカードを使用している部分は、0字以上の任意の文字列が含まれることを意味する。

(例)	ABC	「ABC」という文字列を探す。
	*ABC	「ABC」で終わる文字列を探す。
	ABC*	「ABC」で始まる文字列を探す。
	ABC	「ABC」が含まれる文字列を探す。

<設問1> 次の(1)～(3)のような結果を表1の文字列群から検索するための検索文字列を解答群から選べ。

表1 検索の対象となる文字列

champion
once
on
vision

- (1) champion と on と vision が検索された。
- (2) once と on が検索された。
- (3) on が検索された。

(1) ～ (3) の解答群

- ア. on
- イ. *on
- ウ. on*
- エ. *on*

<設問 2 > 次の流れ図中の に入れるべき適切な字句を解答群から選べ。

[流れ図の説明]

検索文字列を入力し、ファイルに記録してある検索対象文字列の中から一致するものを表示する流れ図である。

検索文字列にはワイルドカードが使用でき、ワイルドカードを表す文字としてアスタリスク(*)を使用する。なお、アスタリスクが2つ以上存在する場合は、1つ目と2つ目のアスタリスクの間にある文字列を有効とし、2つ目のアスタリスク以降にある文字列は無視する。

なお、文字列は配列の1要素に1文字ずつ格納され、配列の添字は0から始まる。

配列の1要素に1文字が格納されている文字列同士で、単純に検索文字列が検索対象文字列に含まれるかを調べる場合は、添字操作を行って検索文字列をスライドさせるようなイメージで一致するかを調べる。

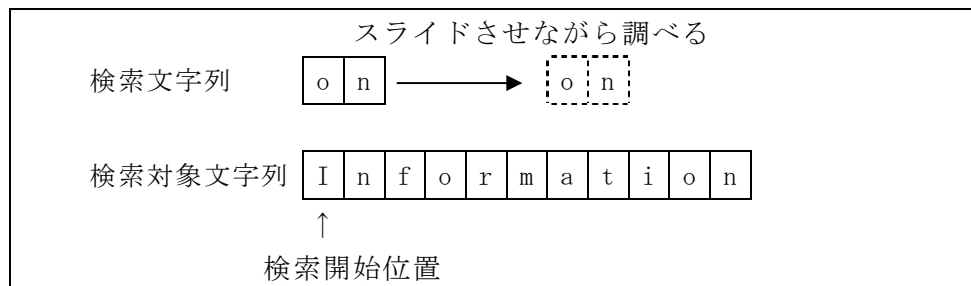


図 1 文字列検索のパターン

しかし、検索文字列にワイルドカード文字が含まれているかどうか、また、ワイルドカード文字の位置によって、検索の方法を検討しなければならない。そこで、次の4つの場合に分けて考える。

1. 文字列の前後にアスタリスクがある場合 (例: *ABC*)

図1にあるように、添字操作を繰り返し行いながら、検索文字列をスライドさせて調べる。検索対象文字列の検索開始位置は、先頭からになる。

2. 文字列の前にのみアスタリスクがある場合 (例: *ABC)

検索対象文字列の後ろを一度だけを調べる。検索対象文字列の検索開始位置は、検索対象文字列の文字数から、検索文字列の文字数を引いた値からになる。

3. 文字列の後ろにのみアスタリスクがある場合 (例: ABC*)

検索対象文字列の前を一度だけを調べる。検索対象文字列の検索開始位置は、先頭からになる。

4. アスタリスクがない場合 (例: ABC)

検索対象文字列と検索文字列が完全に一致するかを調べる。検索対象文字列の検索開始位置は、先頭からになる。

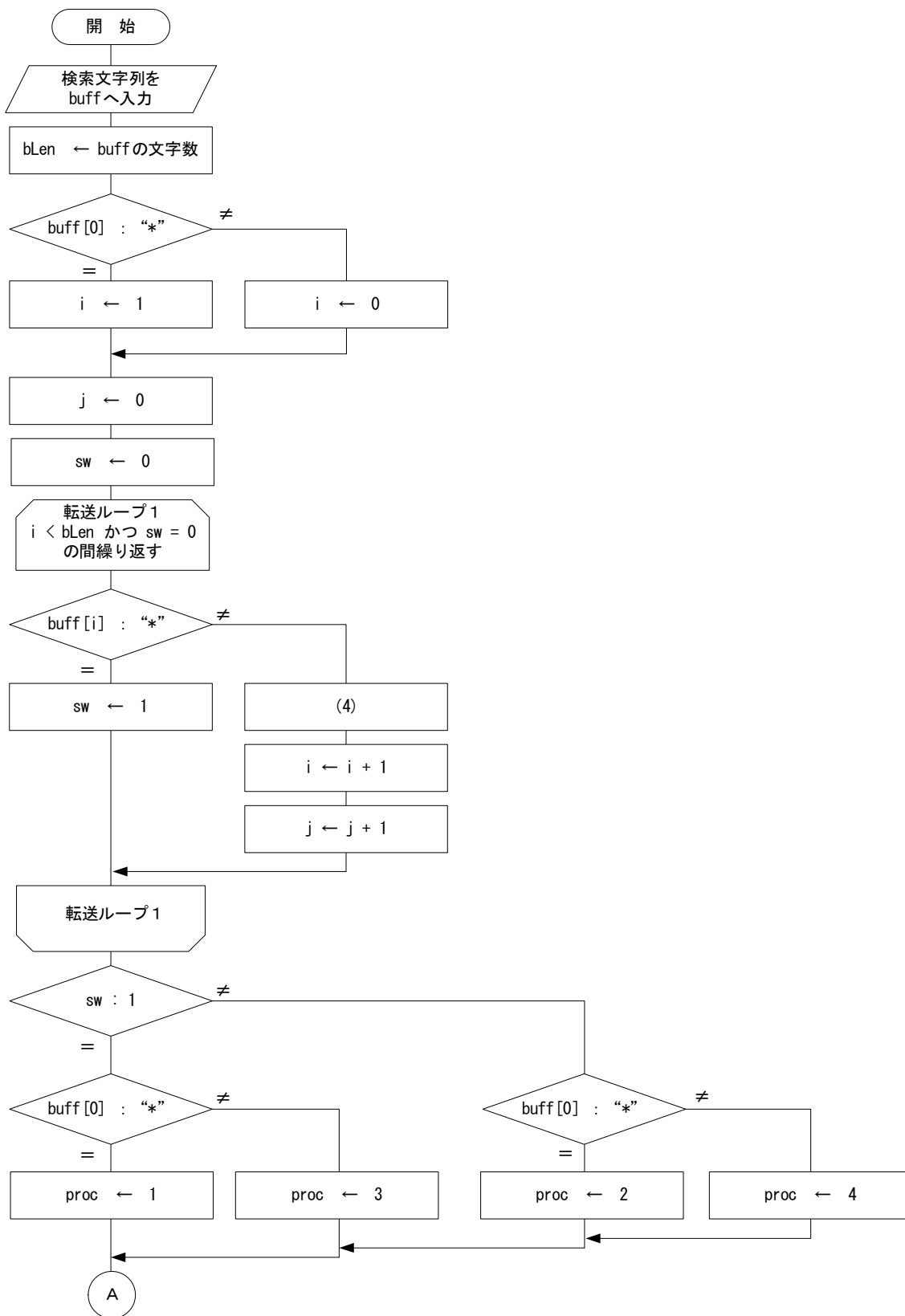


図2 流れ図1

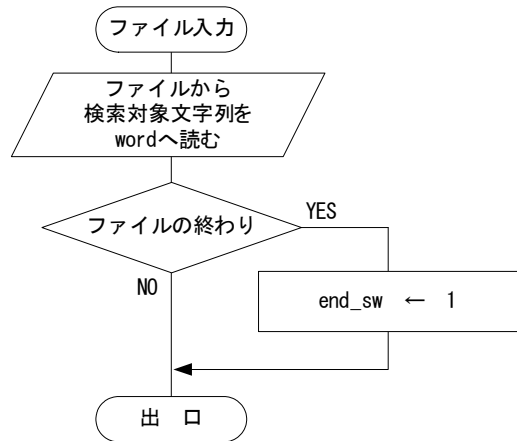
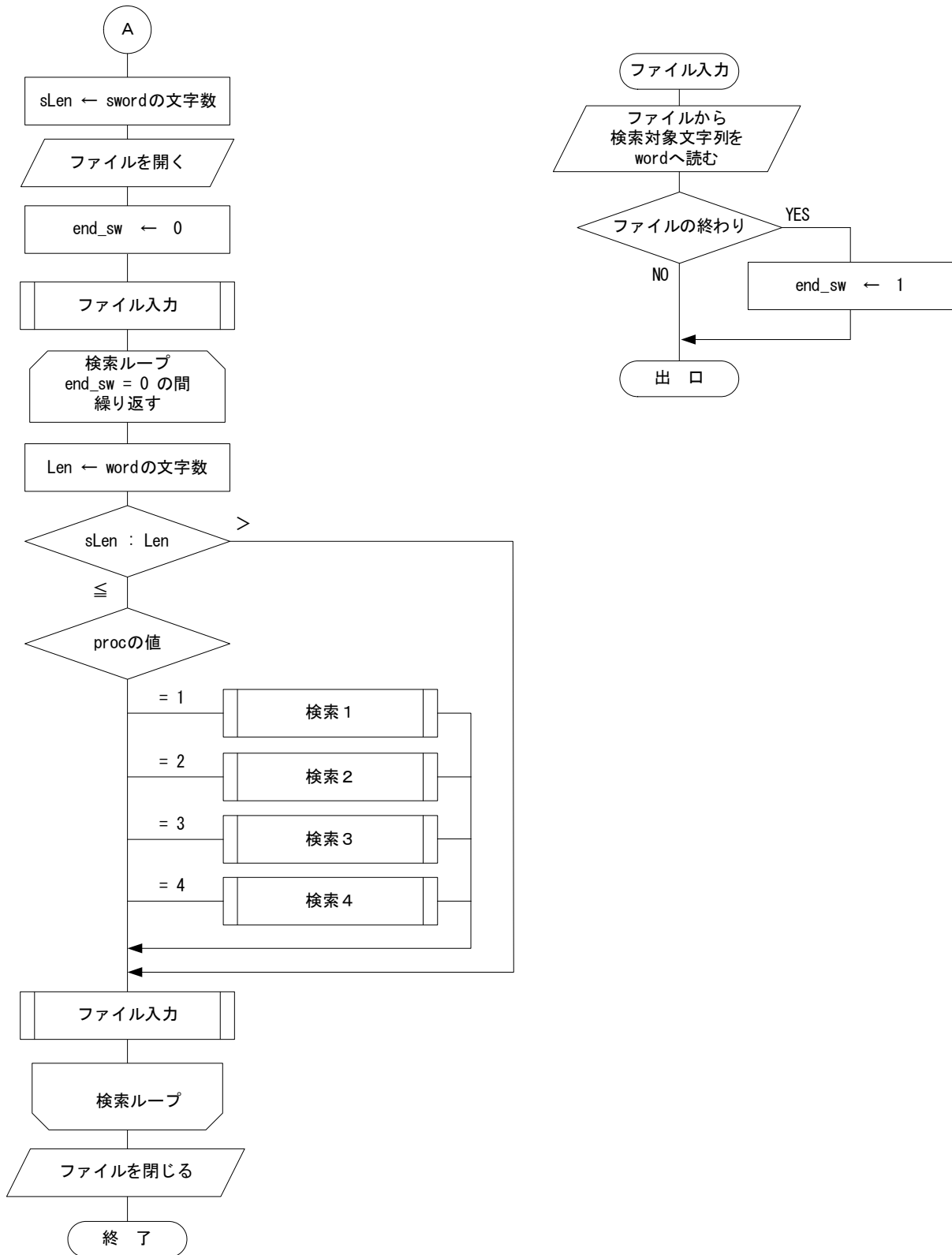


図3 流れ図2

(4) の解答群

- ア. sword[i] ← buff[i]
- ウ. sword[j] ← buff[i]

- イ. sword[i] ← buff[j]
- エ. sword[j] ← buff[j]

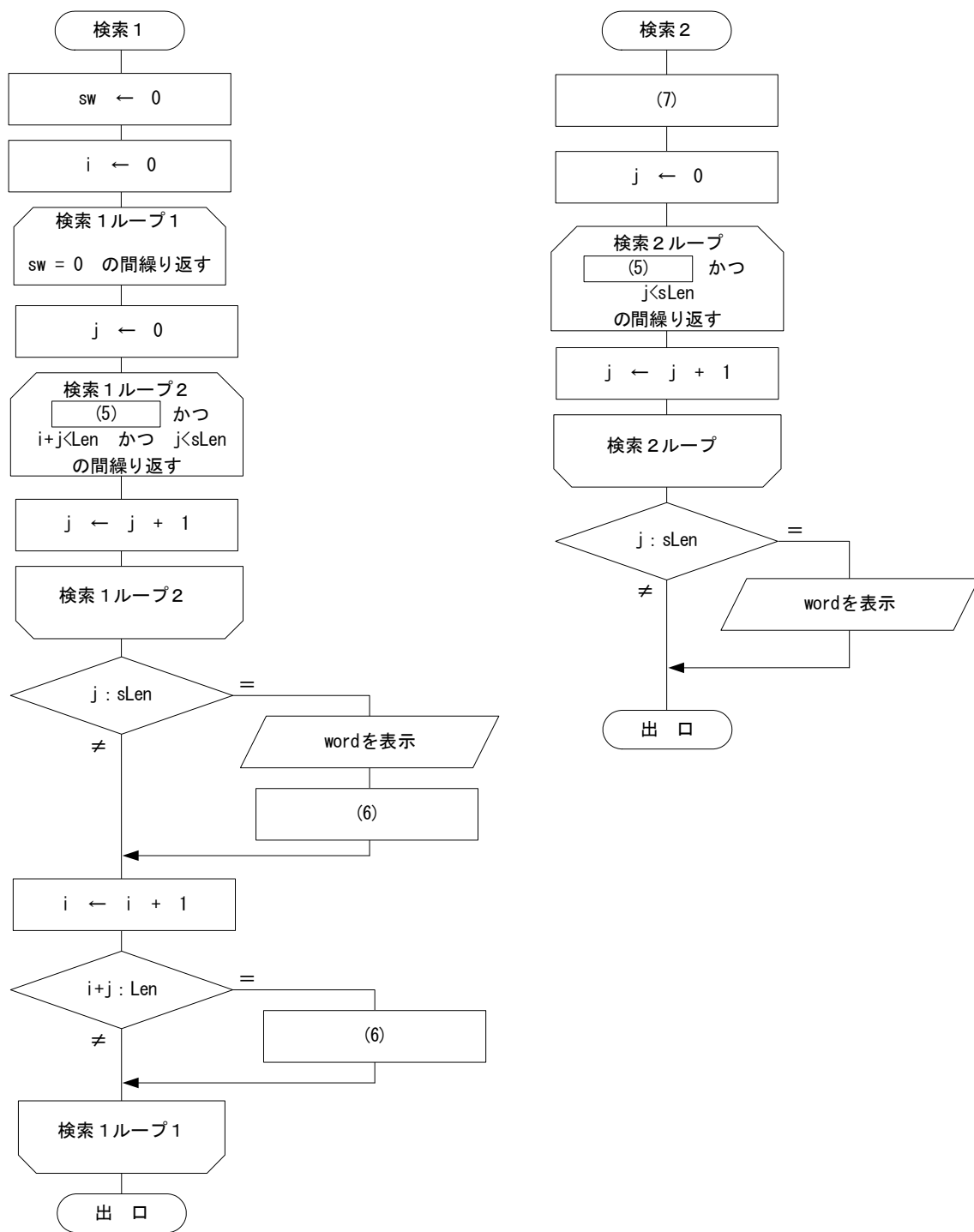


図4 流れ図3

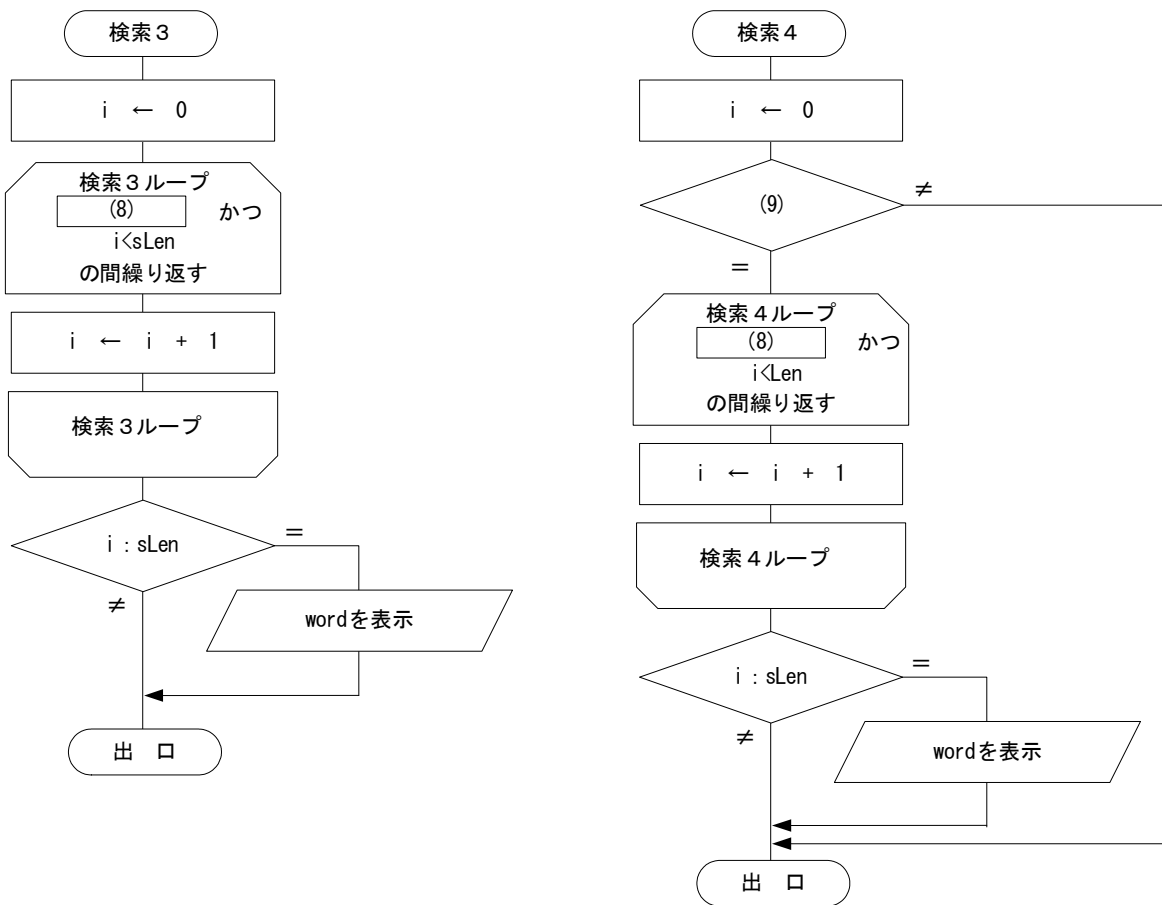


図5 流れ図4

(5) , (8) の解答群

- ア. word[i] = sword[i]
- ウ. word[j] = sword[i]
- オ. word[i] = sword[i+j]
- キ. word[i+j] = sword[i]

- イ. word[i] = sword[j]
- エ. word[j] = sword[j]
- カ. word[j] = sword[i+j]
- ク. word[i+j] = sword[j]

(6) の解答群

- ア. i ← i + 1
- ウ. sw ← 1

- イ. j ← j + 1
- エ. end_sw ← 1

(7) の解答群

- ア. i ← 0
- ウ. i ← Len

- イ. i ← 1
- エ. i ← Len - sLen

(9) の解答群

- ア. i : Len
- ウ. Len : sLen

- イ. i : sLen
- エ. i + Len : sLen

これより

< 選 択 問 題 >

選択問題はA, Bから, それぞれ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 A

- | | |
|------------|---------------|
| ・ C言語の問題 | 15 ページ~18 ページ |
| ・ 表計算の問題 | 20 ページ~22 ページ |
| ・ アセンブラの問題 | 23 ページ~25 ページ |

選択問題 B

- | | |
|------------|---------------|
| ・ C言語の問題 | 26 ページ~30 ページ |
| ・ 表計算の問題 | 31 ページ~34 ページ |
| ・ アセンブラの問題 | 35 ページ~38 ページ |

次のプログラムの説明を読み、設問に答えよ。

[プログラムの説明]

ファイルに記録されている内容をランレングス法により圧縮し、ファイルへ出力するプログラムである。

ランレングス法とは、同じ文字が連続する個数を数えて圧縮する方法であり、連続する個数が多いほど圧縮効率は高くなる。

(例) 圧縮前 : AAAAABBBBBBCCDDDD
圧縮後 : A5B5C2D5

1. 入力および出力するファイルは、main関数のパラメタとして受け取る。第1引数が入力ファイル名（圧縮前）で、第2引数が出力ファイル名（圧縮後）である。
2. 入力ファイルは、半角の英数字のみ（8ビット文字）で記述されている。
3. 出力する内容は、文字（8ビット）と個数（8ビット）の並びである。
4. 連続する個数が255を超えた場合、いったん255個の連続数として出力し、残りの分を数えなおす。

[関数の説明]

FileOutput 関数

引数 : 出力ファイルのポインタ, 文字, 連続数

機能 : 文字と連続数をファイル出力する

戻り値 : なし

[プログラム]

```
#include <stdio.h>
#define MAXSIZE 255
void FileOutput(FILE *, unsigned char, int);
int main(int argc, char *argv[])
{
    FILE *fpi, *fpo;
    unsigned char ch, old_ch;
    int n;
    if (argc != ) {
        printf("パラメタ数が合いません¥n");
        return 1;
    }
}
```

```

fpi = fopen(argv[1], "r");
if (fpi == NULL) {
    printf("ファイル'%s'が見つかりません\n", argv[1]);
    return 1;
}
fpo = fopen(argv[2], "w");
old_ch = NULL;
(2);
while(fread(&ch, sizeof(char), 1, fpi) > 0) {
    if (ch != old_ch) { /* 異なる文字が出現した場合 */
        if (old_ch != NULL) FileOutput(fpo, old_ch, n);
        (3);
        (2);
    } else {
        if (n >= MAXSIZE) { /* 255以上であれば、いったん出力する */
            FileOutput(fpo, ch, n);
            (2);
        }
    }
    n++;
}
(4);
fclose(fpi);
fclose(fpo);
return 0;
}

void FileOutput(FILE *fp, unsigned char ch, int n)
{
    fwrite(&ch, sizeof(unsigned char), 1, fp);
    fwrite(&n, sizeof(unsigned char), 1, fp);
}

```

<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) の解答群

ア. 1 イ. 2 ウ. 3 エ. 4

(2) の解答群

ア. n = 0 イ. n = 1 ウ. n++ エ. n--

(3) の解答群

- ア. `ch = old_ch` イ. `old_ch = ch`
- ウ. `ch = n` エ. `old_ch = n`

(4) の解答群

- ア. `FileOutput(fpo, ch, NULL)`
- イ. `FileOutput(fpo, old_ch, n)`
- ウ. `FileOutput(fpo, ch, old_ch)`
- エ. `FileOutput(fpo, old_ch, ch)`

<設問 2> 次の圧縮効率に関する記述中の に入れるべき適切な数値を解答群から選べ。

このランレングス法では、連続する文字数によっては、圧縮効率が悪くなる。例えば、「ABABABAB…」というように、異なる文字が交互に出現すると、元のデータ量より大きくなってしまう。

そこで、連続する文字数が2以下の場合には圧縮せずに、そのまま出力することを考える。ただし、そのままではデータと連続する回数との区別がつかないので、前提条件として、扱う文字は、表1の文字コードにあるものとする。

表1 使用する文字と文字コード

下位 4bit 上位 4bit	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
2	間隔	!	"	#	\$	%	&	'	()	*	+	,	-	.	/
3	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
4	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
5	P	Q	R	S	T	U	V	W	X	Y	Z	[¥]	^	_
6	~	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
7	p	q	r	s	t	u	v	w	x	y	z	{		}	~	

表1から、文字データを構成する8ビットのうち、先頭ビットは必ず0になることがわかる。これを利用し、図1のように、先頭ビットをデータと回数を区別するフラグとし、先頭ビットが0であればデータ、1であれば回数というように区別する。

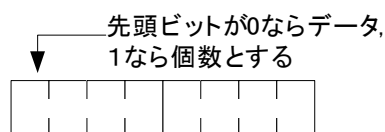


図1 ビット形式

このようにすることで、連続する個数を示す数の最大値も変更する必要があり、これまでの255から になる。

(5) の解答群

ア. 127

イ. 128

ウ. 224

エ. 248

問題を読みやすくするために、
このページは空白にしてあります。

選択問題A 表計算の問題

関数、式の仕様は次のとおりである。

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])

ワークシートを横断して集計する場合の書式

ワークシート a, b, c, d の連続するワークシートのセル A1 を横断して合計を計算する場合、=SUM('a:d'!A1)となる。

書式：関数 ('参照する先頭のワークシート名：参照する末尾のワークシート名'!セルまたはセル範囲)

J 商店では、商品の在庫を管理するために表計算ソフトを用いている。商品別および月別に表を作成しており、同一形式の表が月別の商品別に存在する。ワークシート名は商品名と月（1月～12月）を使用しており、商品名と月の昇順に並んでいる。次の表は、商品Aの1月の表である。この表のワークシート名は、「商品A1月」である。

表1 月別商品別在庫管理表

	A	B	C	D
1	2010年1月	集計データ		
2	商品名:	商品A		
3	仕入単価:	120	現在庫数	100
4	販売単価	250		
5	当月利益額	694000		
6				
7	日付	取引種別	取引量	取引金額
8	前月繰越	1	250	
9	1月4日	1	100	-12000
10	1月4日	2	50	12500
11	1月5日	1	100	-12000
12	1月5日	2	50	12500
13	1月6日	1	300	-36000
14	1月6日	2	50	12500
15	1月7日	1	400	-48000
⋮	⋮	⋮	⋮	⋮
50	1月28日	1	300	-36000
51	1月28日	2	150	37500
52	1月29日	1	100	-12000
53	1月29日	2	150	37500
54	1月30日	1	200	-24000
55	1月30日	2	200	50000
56	1月31日	2	350	87500
57	次月繰越		100	

- ・ 取引種別は 1 が仕入で 2 が販売を表す。
- ・ 取引金額は仕入の場合はマイナスの値，販売の場合はプラスの値で表示する。
- ・ 仕入単価および販売単価は同一月では変化はしないものとする。
- ・ 当月利益額は販売金額から仕入金額を引いて求めるものとする。

<設問 1> 月別商品別在庫管理表に関する次の記述の に入れるべき適切な式または字句を解答群から選べ。

現在庫数を計算するため，セル D3 には以下の式を入力する。

$$= \text{ (1) } - \text{ (2) }$$

取引金額を計算するため，セル D9 に以下の式を入力し，セル D10 からセル D56 に複写する。

$$= \text{ (3) }$$

当月利益額を計算するため，セル B5 に以下の式を入力する。

$$= \text{ (4) }$$

(1) , (2) の解答群

- | | |
|--------------------------------|--------------------------------|
| ア. SUM(B8:B56) | イ. SUM(C8:C56) |
| ウ. SUMIF(B8:B56, "=1") | エ. SUMIF(C8:C56, "=1") |
| オ. SUMIF(B8:B56, "=2") | カ. SUMIF(C8:C56, "=2") |
| キ. SUMIF(B8:B56, "=1", C8:C56) | ク. SUMIF(C8:C56, "=1", B8:B56) |
| ケ. SUMIF(B8:B56, "=2", C8:C56) | コ. SUMIF(C8:C56, "=2", B8:B56) |

(3) の解答群

- ア. IF(B9=1, C9*B\$3, C9*B\$4)
- イ. IF(B9=1, C9*B\$3*(-1), C9*B\$4)
- ウ. IF(B9=1, C9*B\$3, C9*B\$4*(-1))
- エ. IF(B9=1, C9*B\$3*(-1), C9*B\$4*(-1))

(4) の解答群

- | | |
|------------------------|--|
| ア. SUM(D9:D56) | イ. SUMIF(D9:D56, ">1") |
| ウ. SUMIF(D9:D56, "<1") | エ. SUMIF(D9:D56, ">1")-SUMIF(D9:D56, "<1") |

商品Aの2009年1月から12月の月別商品別在庫管理表を集計した表を作成した。

表2 商品別年間集計表

	A	B	C
1	2009年 年間	集計データ	
2	商品名:	商品A	
3	年間利益額	¥6,457,400	
4			
5		仕入量	販売量
6	1月	5300	5200
7	2月	5190	5160
8	3月	3260	2490
9	4月	3650	3420
10	5月	3520	3510
11	6月	4325	4305
12	7月	5200	5200
13	8月	4100	5200
14	9月	4620	4600
15	10月	5920	5600
16	11月	4250	4050
17	12月	5630	5400
18	計	54965	54135

<設問2> 商品別年間集計表に関する次の記述中の に入れるべき適切な式または字句を解答群から選べ。

年間純利益を計算するために、セルB3に次の式を入力する。なお、年間純利益は、ワークシート商品A1月から商品A12月の合計で、ワークシートを横断して集計する。

=

(5) の解答群

ア. SUM(B5)

イ. SUM('商品A1月'!B5)

ウ. SUM('商品A12月'!B5)

エ. SUM('商品A1月:商品A12月'!B5)

選択問題A アセンブラの問題

次のCASL IIプログラムに関する説明を読み、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。

[プログラムの説明]

アパートの物件検索を行う副プログラムである。検索条件は最寄り駅からの距離と家賃の2つの条件とする。それぞれの条件は物件ごとに図1のように格納されている。

物件番号	最寄りの駅からの距離	家賃	その他
8ビット	2ビット	2ビット	4ビット

図1 物件情報

最寄りの駅からの距離に関する情報は2ビットで表1のように表す。

表1 最寄り駅からの距離情報

2ビット情報	内容
00	最寄り駅から1km未満
01	最寄り駅から1km～3km未満
10	最寄り駅から3km以上

家賃に関する情報は2ビットで表2のように表す。

表2 家賃情報

2ビット情報	内容
00	5万円未満
01	5万円～7万円未満
10	7万円～10万円未満
11	10万円以上

物件情報はGR1を先頭アドレスとするn語の連続した領域に次のように格納されている。

(GR1) +0	物件数(n)
+1	1件目の物件情報
	⋮
+(n)	n件目の物件情報

図2 物件情報の引数

検索条件はGR2を先頭アドレスとする2語の連続した領域に0~3の数値で格納されている。

(GR2) +0	最寄駅からの距離情報
+1	家賃情報

図3 検索条件の引数

検索した結果は物件情報の物件番号(8ビット)を取り出し、GR3を先頭アドレスとする連続した領域に格納し主プログラムに戻る。

(GR3) +0	該当する物件番号(1件目)
+1	該当する物件番号(2件目)
	⋮

図4 検索結果

[プログラム]

```
SEARCH START
    LD    GR0,0,GR2    ; 最寄り駅からの距離情報の取り出し
    SLL  GR0,6
    ST    GR0,WK
    [ ] (1)           ; 家賃情報の取り出し
    SLL  GR0,4
    [ ] (2)           ; 検索条件を結合
    ST    GR0,WK      ; 検索条件を格納
    LD    GR0,0,GR1   ; 物件数の取り出し
    ADDL GR0,GR1
    ST    GR0,N       ; 検索終了アドレスを格納
    LAD  GR1,1,GR1
LOOP   LD    GR0,0,GR1 ; 物件情報の取り出し
    [ ] (3)           ; 条件の取り出し
    XOR  GR0,WK      ; 条件を検証
    JNZ  L1
    LD    GR0,0,GR1
    SRL  GR0,8       ; 物件番号のみ取り出し
    ST    GR0,0,GR3
    LAD  GR3,1,GR3
L1     LAD  GR1,1,GR1
    CPL  GR1,N
    [ ] (4)           ; 繰り返し判定
    JZE  LOOP
    RET
WK     DS    1
N      DS    1
MASK   DC    [ ] (5)
END
```

(1) ~ (4) の解答群

ア. LD GR0, 0, GR2 イ. LD GR0, 1, GR2 ウ. LD GR1, GR0
エ. AND GR0, WK オ. AND GR0, MASK カ. OR GR0, WK
キ. JPZ LOOP ク. JMI LOOP ケ. JNZ LOOP

(5) の解答群

ア. #000F イ. #00F0 ウ. #0F00 エ. #FF00

選択問題B C言語の問題

次の高速バスの予約に関する記述を読み、設問に答えよ。

J交通は主要都市を結ぶ高速バスを運行している。高速バスの需要は年々伸びており、他社との差別化を図るため、J交通で運行しているバスは、ゆったりと乗れることをコンセプトにしたものになっている。料金は従来型のバスに比べると少し高めに設定しているが、それでも少しでもゆとりを持った旅をしたいユーザには好評である。なお、このバスは完全予約制で、出発時刻の24時間前までに申し込みが必要である。

J交通で運行しているバスは、3列シート（A列、B列、C列）であり、B列とC列の間に通路があるタイプである。シート番号は1から9まであり、全部で27人が乗車可能である。

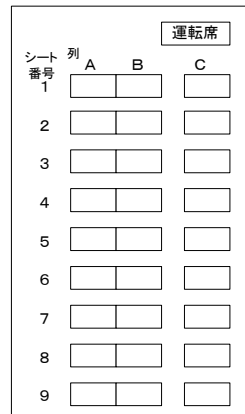


図1 バスの座席

座席を割り当てる方法は、次のようになる。

- 1人ずつ席を割り当てる場合は、A列、C列、B列のシート番号の小さい順に割り当てる。
- 2人以上のグループで予約をした場合は、隣り合うA列とB列を割り当てる。ただし、隣り合う座席が確保できない場合は、1人ずつ席を割り当てる。

例えば、次の図2の状況で5人の予約を入れると、図3のように割り当てる。

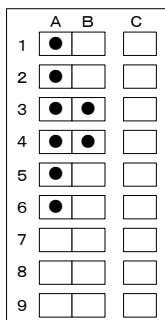


図2 予約前（●が予約済み）

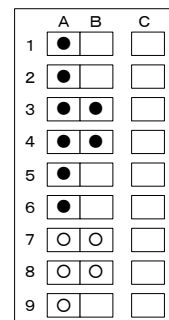


図3 予約後（○が新しい予約）

[プログラムの説明]

座席の予約をする関数SheetReservである。

1. 座席の予約状況は大域変数Sheet_Areaに格納されており、空いている座席は0、予約済みの座席は1が格納されている。
2. SheetReservは、予約したい人数を受け取り、全員の予約ができれば、その座席（列名とシート番号）をリスト構造で格納したデータの先頭のポインタを返し、全員の予約ができなければ、NULLを返す。なお、列名は‘A’，‘B’，‘C’という文字である

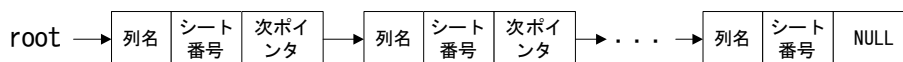


図4 SheetReservが返すリスト

[関数の説明]

SheetReserv 関数

引 数：予約人数。

機 能：座席の予約を行い、予約ができた座席をリスト構造で格納する。

戻り値：予約した座席をリスト構造で格納した先頭のポインタ。ただし、予約ができなかった場合は NULL。

ListInsert 関数

引 数：座席の列名，シート番号。

機 能：リスト構造に座席の列名とシート番号を格納する。

戻り値：なし。

[プログラム]

```

#include <stdio.h>
#include <stdlib.h>

#define COLS 3          /* 列数 */
#define ROWS 9         /* 行数 */
typedef struct s_rec { /* 構造体宣言 */
    char col;          /* 座席の列名 */
    int no;            /* 座席の番号 */
    struct s_rec *next; /* 次のポインタ */
} SHEET_REC;

SHEET_REC *root, *old_p; /* 返却リストの先頭と前のポインタ */
int Sheet_Area[COLS][ROWS]; /* 座席の予約状況管理用配列 */
SHEET_REC *SheetReserv(int); /* プロトタイプ宣言 */
void ListInsert(char, int); /* プロトタイプ宣言 */
  
```

```

SHEET_REC *SheetReserv(int ninzu)
{
    int priority[3] = { 0, 2, 1 }; (a)
    int i, j, c, n;
    /* 人数分確保できるか調べる */
    n = 0;
    for(i=0; i<COLS; i++) {
        for(j=0; j<ROWS; j++) {
            if (Sheet_Area[i][j] == 0) n++;
        }
    }
    if ( (1) ) return NULL;
    /* 返すリストの先頭を初期化 */
    root = NULL; (b)
    for(i=0; i<COLS && ninzu>0; i++) {
        c = priority[i]; (c)
        for(j=0; j<ROWS && ninzu>0; j++) {
            if (Sheet_Area[c][j] == 0) { /* 空席を見つけた場合 */
                if (c == 0 && ninzu > 1) { /* 隣同士を予約の場合 */
                    if (Sheet_Area[1][j] == 0) {
                        Sheet_Area[0][j] = 1;
                        Sheet_Area[1][j] = 1;
                        ListInsert('A', j+1);
                        ListInsert('B', j+1);
                        (2);
                    }
                } else {
                    Sheet_Area[c][j] = 1; /* 1つの席を予約 */
                    ListInsert('A'+c, j+1);
                    (3);
                }
            }
        }
    }
    return root;
}

```

```

void ListInsert(char ch, int n)
{
    SHEET_REC *p;
    /* 格納する領域を確保 */
    p = (SHEET_REC *)malloc(sizeof(SHEET_REC));
    if (root == NULL)
        root = p;
    else
        (4);
    p->row = ch;
    p->no = n;
    p->next = NULL;
    (5);
}

```

<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) の解答群

- | | |
|------------------------------|------------------------------|
| ア. <code>n < 0</code> | イ. <code>n > 0</code> |
| ウ. <code>n > ninzu</code> | エ. <code>n < ninzu</code> |

(2) , (3) の解答群

- | | |
|----------------------------|----------------------------|
| ア. <code>ninzu--</code> | イ. <code>ninzu++</code> |
| ウ. <code>ninzu -= 2</code> | エ. <code>ninzu += 2</code> |
| オ. <code>ninzu += j</code> | カ. <code>ninzu -= j</code> |
| キ. <code>ninzu -= n</code> | ク. <code>ninzu += n</code> |

(4) の解答群

- | | |
|---------------------------------------|------------------------------------|
| ア. <code>old_p->next = root</code> | イ. <code>old_p->next = p</code> |
| ウ. <code>p->next = old_p</code> | エ. <code>root->next = p</code> |

(5) の解答群

- | | |
|---------------------------|---|
| ア. <code>old_p = p</code> | イ. <code>old_p->next = p->next</code> |
| ウ. <code>root = p</code> | エ. <code>root->next = p->next</code> |

<設問2> プログラムの改善に関する次の記述中の に入れるべき適切な字句を解答群から選べ。

このプログラムは、全予約人数によって、バスの左側にかたよって座席を割り当てる傾向がある。

そこで、予約人数が複数人の場合はA列、C列、B列の順に、予約人数が1人の場合はC列、A列、B列の順に割り当てるようにするため、プログラムの(a)、(b)、(c)の部分に変更と追加を行う。

(a)の変更

座席の優先順位を人数によって変更するため、priorityを2次元配列とし、次の値を設定する。

```
int priority[2][3] = { { 0, 2, 1 },      /* 複数人 */
                      { 2, 1, 0 } };    /* 1人 */
```

(b)の次に追加する処理と(c)の変更

2次元配列priorityからデータを取り出すため、(c)を次のように変更する。

```
c = priority[n][i];
```

また、priorityの添え字を決定するため、(b)の次に以下の処理を追加する。

```
if (ninzu == 1)
     (6);
else
     (7);
```

(6)、(7)の解答群

ア. n = 0

イ. n = 1

ウ. n = 2

エ. n = 3

選択問題B 表計算の問題

関数、式の仕様は次のとおりである。

COUNT 関数

範囲内にある数値の入ったセルの個数を返す。

書式：COUNT(範囲)

RANK 関数

範囲内にある数値の順位を返す。順序が 0 の場合、降順の順位を返し、0 以外の場合、昇順の順位を返す。

書式：RANK(数値, 範囲, 順序)

ROUND 関数

指定した桁で値を四捨五入する。正の数であれば小数点以下、負の数であれば小数点以上の桁になる。例えば、1 にすると小数点以下第 2 位を四捨五入して小数点以下第 1 位までを表示し、-1 にすると 1 の位で四捨五入する。

書式：ROUND(式または値, 桁数)

STDEV 関数

範囲内の数値の標準偏差を返す。

書式：STDEV(範囲)

J 学院の A 先生は、自分が担任をしているクラスの成績を管理するために表計算を用いている。今回行われた試験の結果を表 1 にまとめた。

表 1 試験結果表

	A	B	C	D	E	F	G	H	I	J	K
1	番号	名前	国語	数学	理科	社会	英語	合計点数	平均点数	順位	偏差値
2	1	相沢博	87	87	76	75	60	385	77	13	54.37
3	2	伊藤司	90	98	99	100	92	479	95.8	2	62.93
4	3	上野美由紀	87	83	80	70	65	385	77	13	54.37
5	4	遠藤幸一	56	55	52	52	43	258	51.6	32	42.80
6	5	小野寺真一	65	70	75	65	70	345	69	22	50.72
7	6	小野寺修	60	72	73	73	65	343	68.6	23	50.54
8	7	加藤奈津美						0	0	39	19.29
9	8	加藤真理	79	71	87	63	71	371	74.2	17	53.09
10	9	栗本愛子			72	73	77	222	74	35	39.52
11	10	小泉元子	65	65	70	60	75	335	67	24	49.81
∴	∴	∴	∴	∴	∴	∴	∴	∴	∴	∴	∴
37	36	森口文明	75	72	92	85	75	399	79.8	10	55.64
38	37	山口幸一	72	60				132	66	38	31.32
39	38	山田修三	0	0	0	0	0	0	0	39	19.29
40	39	弓野東一	98	92	98	100	98	486	97.2	1	63.57
41	40	渡辺浩一郎	98	90	85	88	92	453	90.6	4	60.56
42		クラス平均	71.34	69.47	68.08	70.05	67	337.08			109.77

各科目に点数を入力した。ただし欠席の科目は、後日再試験を行うため空欄になっている。そのため、表内の平均点数およびクラス平均は、空欄のセルは含めない。

<設問 1 > 試験結果表に関する次の記述中の に入れるべき適切な式または字句を解答群から選べ。

個人別の合計点数を計算するためにセル H2 に以下の式を入力し、セル H3 から H41 に複写した。

$$= \text{SUM}(C2:G2)$$

個人別の平均点数を計算するためにセル I2 に $\text{AVERAGE}(C2:G2)$ を入力しようとしたが、この式では (1) という問題が生じる可能性がある。そこで、これを解決するためにセル I2 を以下の式に変更し、セル I3 から I41 へ複写した。

$$= \text{ (2) }$$

各個人の合計点数に対する順位を計算するためにセル J2 に以下の式を入力し、セル J3 から J41 に複写した。

$$= \text{ (3) }$$

各科目別および合計点数でクラス全体の平均点数（小数点以下第 3 位四捨五入）を計算するためにセル C42 に以下の式を入力し、セル D42 から H42 に複写した。ただし、各クラスの受験者は必ず複数人いるものとする。

$$= \text{ (4) }$$

しかし、この式ではセル H42 において、 (5) という問題点が発生した。そこで、問題を解決するために (6) という変更を行った。これにより、表内の他の計算結果に影響がないように修正することができた。

合計点数に対する偏差値を計算する。偏差値を求める式は以下のようになる。

偏差値を求める式

$$(\text{点数} - \text{平均点}) * 10 / \text{標準偏差} + 50$$

まず、標準偏差を計算するために、セル K42 に以下の式を入力する。

$$= \text{STDEV}(H2:H41)$$

偏差値を求めるために、セル K2 に以下の式を入力し、セル K3 から K40 まで複写した。

セル K2 に入力する式

$$= \text{ (7) }$$

(1) の解答群

- ア. 0 による除算が発生したため、エラーが発生する
- イ. 無限小数が発生したため、エラーが発生する
- ウ. 空欄が数字と判別され、正しい計算結果が表示されない
- エ. 循環参照をしてしまい、正しい計算結果が表示されない

(2) の解答群

- ア. $\text{IF}(\text{COUNT}(C2:G2)=0, 0, \text{AVERAGE}(C2:G2))$
- イ. $\text{IF}(\text{COUNT}(C2:G2)=0, \text{AVERAGE}(C2:G2), 0)$
- ウ. $H2/\text{COUNT}(C2:G2)$
- エ. $\text{COUNT}(C2:G2)/H2$

(3) の解答群

- ア. $\text{RANK}(H2, H2:H41, 0)$
- イ. $\text{RANK}(H2, H\$2:H\$41, 0)$
- ウ. $\text{RANK}(H2, \$H2:\$H41, 0)$
- エ. $\text{RANK}(H2, \$H2:H\$41, 0)$

(4) の解答群

- ア. $\text{AVERAGE}(C2:C41)$
- イ. $\text{AVERAGE}(C2:C41)*0.05$
- ウ. $\text{ROUND}(\text{AVERAGE}(C2:C41), 2)$
- エ. $\text{ROUND}(\text{AVERAGE}(C2:C41), -2)$

(5) の解答群

- ア. ある科目で欠席のあった学生もクラス平均の分母に入ってしまうため、正しい結果が得られない
- イ. 個人別の合計点数の値に誤差があるため、正しい結果が得られない
- ウ. 個人別の平均点数の値に誤差があるため、正しい結果が得られない
- エ. 全科目欠席の学生がクラス平均の分母に入らないため、正しい結果が得られない

(6) の解答群

- ア. 合計点数が 0 の場合は、クラス平均の分母にカウントしないようにする
- イ. 個人別に各科目の点数で空欄でないセル数を集計する列を追加し、5 未満の場合はクラス平均の分母にカウントしないようにする
- ウ. 全科目欠席の学生の合計点数を空欄で表示するようにする
- エ. セル C2 から G41 までの空欄のセルをカウントした値からクラス人数を引いた値を分母とする

(7) の解答群

- ア. $H2-H\$42*10/K\$42+50$
- イ. $(H2-H\$42)*10/K\$42+50$
- ウ. $(H2-H\$42*10)/K\$42+50$
- エ. $(H2-H\$42*10/K\$42)+50$

試験結果表の順位を複写することで、前回の試験の順位と比較できる表を作成した。

表 2 順位比較表

	A	B	C	D	E
1			前回順位	変化	今回順位
2	1	相沢博	16	↑(3)	13
3	2	伊藤司	1	↓(1)	2
4	3	上野美由紀	11	↓(2)	13
5	4	遠藤幸一	31	↓(1)	32
6	5	小野寺真一	24	↑(2)	22
7	6	小野寺修	26	↑(3)	23
8	7	加藤奈津美	20	↓(19)	39
9	8	加藤真理	12	↓(5)	17
10	9	栗本愛子	17	↓(18)	35
⋮	⋮	⋮	⋮	⋮	⋮
36	35	村西薫	15	→(0)	15
37	36	森口文明	10	→(0)	10
38	37	山口幸一	36	↓(2)	38
39	38	山田修三	40	↑(1)	39
40	39	弓野東一	3	↑(2)	1
41	40	渡辺浩一郎	4	→(0)	4

<設問 2> 順位比較表に関する次の記述の に入れるべき適切な式を解答群から選べ。

セル C2 から C41 に前回行われた試験の順位を入力し、セル E2 から E41 に今回行われた試験の順位を入力した。セル D2 からセル D41 には、この間の順位の変化を表 2 にあるように表示させたい。そのために、セル D2 に以下の式を入力し、セル D3 から D41 に複写した。

$$= \text{ (8) } \& \text{"("} \& \text{ (9) } \& \text{"")"}$$

(8) , (9) の解答群

- ア. E2-C2 イ. C2-E2 ウ. IF(E2>C2,"↓","↑") エ. "("
 オ. IF(E2>C2,E2-C2,C2-E2) カ. IF(E2>C2,"↑","↓") キ. ")"
 ク. IF(E2>C2,C2-E2,E2-C2) ケ. IF(E2>C2,"↓",IF(E2=C2,"→","↑"))
 コ. IF(E2>C2,"↓",IF(E2=C2,"↑","→"))

選択問題B アセンブラの問題

次のCASL II プログラムに関する説明を読み、設問に答えよ。

[プログラムの説明]

文字列（256文字以下）を1行（256文字）内で右揃え、左揃え、中央揃えの3種類の文字揃えを行う副プログラムである。

主プログラムからの引数は表1のとおりとする。

(GR1)+0	文字列が格納されている領域の先頭アドレス
+1	文字揃え後の文字列を格納する領域の先頭アドレス
+2	文字数(256以下)
+3	文字揃え指定

図1 プログラムの引数

ここで、文字揃え指定は次のとおりとする。

L: 左揃え

R: 右揃え

M: 中央揃え

(文字揃えの例)

文字列

J	-	K	E	N
---	---	---	---	---

文字揃え指定が' L ' のとき

J	-	K	E	N		...	
---	---	---	---	---	--	-----	--

文字揃え指定が' R ' のとき

	...		J	-	K	E	N
--	-----	--	---	---	---	---	---

文字揃え指定が' M ' のとき

	...		J	-	K	E	N	...	
--	-----	--	---	---	---	---	---	-----	--

なお、文字揃え後文字列を格納する領域にはあらかじめ、256個の空白が格納されている。

[プログラム]

```
MOJI    START
        LD      GR2, 0, GR1      ; 文字列の先頭アドレス
        LD      GR3, 1, GR1      ; 文字列を格納する領域の先頭アドレス
        LD      GR4, 2, GR1      ; 文字数
        LD      GR5, 3, GR1      ; 文字揃え指定
        CPL     GR5, ='L'
        [ ] (1)
        LD      GR6, GR3          ; 文字揃え後の文字列の先頭アドレス
        JUMP    L3
L1      CPL     GR5, ='R'
        JNZ     L2
        LAD     GR6, 256, GR3
        [ ] (2)                  ; 文字揃え後の文字列の先頭アドレス
        JUMP    L3
L2      CPL     GR5, ='M'
        JNZ     L4
        LAD     GR6, 128, GR3
        [ ] (3)
        SUBL    GR6, GR4          ; 文字揃え後の文字列の先頭アドレス
L3      CALL    CHG
        JUMP    L5
L4      OUT     OUTBUF, OUTLEN
L5      RET
OUTBUF  DC     '[ ] (a) '
OUTLEN  DC     20
        END
```

; 文字列を文字揃え後の領域に格納する

```
CHG     START
        LD      GR7, GR6
        [ ] (4)                  ; 文字列の最後の文字の格納アドレス + 1
LOOP    LD      GR0, 0, GR2
        [ ] (5)
        LAD     GR2, 1, GR2
        LAD     GR6, 1, GR6
        CPL     GR6, GR7
        JMI     LOOP
        RET
        END
```

<設問 1> プログラム中の (1) ~ (5) の に入れるべき適切な命令を解答群から選べ。

(1) ~ (5)の解答群

ア. **ADDL GR6, GR4**

ウ. **SUBL GR6, GR4**

オ. **SLA GR4, 1**

キ. **ST GR0, 0, GR6**

ケ. **JNZ L3**

イ. **ADDL GR7, GR4**

エ. **ADDL GR7, 2, GR1**

カ. **SRA GR4, 1**

ク. **JNZ L1**

コ. **JNZ L5**

<設問 2> プログラム中の (a) に入れるべきメッセージの内容として適切な字句を解答群から選べ。

(6) の解答群

ア. 文字列が不足している。

イ. 文字列が超過している。

ウ. 文字揃え指定に誤りがある。

エ. 文字揃え後の文字列の先頭アドレスに誤りがある。

<設問 3> インデント処理ができるようにプログラムを変更したい。文字揃え指定にインデントする数値を格納し、この副プログラムを呼び出す。このとき、ラベル L4 の命令を下記の命令群と置き換え、エラー表示用のデータを追加した。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。

(インデントとして5を指定した場合の例)

					J	-	K	E	N		...	
--	--	--	--	--	---	---	---	---	---	--	-----	--

(インデント処理を追加したプログラム)

```
L4      LD      GR6,GR3
        ADDL   GR6,GR5
        

|     |
|-----|
| (7) |
|-----|


        CPA    GR5,=256
        JPL    ERR
        CALL   CHG
        JUMP   L5
ERR     OUT    ERRMSG,ERRLEN
```

(エラー表示用に追加するデータ)

```
ERRMSG  DC    'LENGTH ERROR'
ERRLEN  DC    12
```

(7) の解答群

ア. ADDA GR5, 0, GR1
ウ. SUBA GR5, 2, GR1

イ. ADDA GR5, GR4
エ. SUBA GR5, GR4

<メモ欄>

<メモ欄>

<メモ欄>

