FE

平成22年度　秋期
基本情報技術者試験
午後　問題

試験時間	13:00 ～ 15:30（2時間30分）

注意事項

1. 試験開始及び終了は，監督員の時計が基準です。監督員の指示に従ってください。
2. 試験開始の合図があるまで，問題冊子を開いて中を見てはいけません。
3. この注意事項は，問題冊子の**裏表紙**に続きます。必ず読んでください。
4. **答案用紙への受験番号などの記入は，試験開始の合図があってから始めてください。**
5. 問題は，次の表に従って解答してください。

問題番号	問1 ～ 問7	問8	問9 ～ 問13
選択方法	5問選択	必須	1問選択

6. 答案用紙の記入に当たっては，次の指示に従ってください。
   (1) 答案用紙は光学式読取り装置で読み取った上で採点しますので，答案用紙の**マークの記入方法**のとおりマークしてください。マークの濃度がうすいなど，**マークの記入方法**のとおりマークされていない場合は，読み取れないことがあります。
   (2) B又はHBの黒鉛筆を使用してください。シャープペンシルを使用しても構いませんが，マークの濃度に十分ご注意ください。訂正の場合は，あとが残らないように消しゴムできれいに消し，消しくずを残さないでください。
   (3) **受験番号欄**に，**受験番号**を記入及びマークしてください。正しくマークされていない場合は，採点されません。
   (4) **生年月日欄**に，受験票に印字されているとおりの**生年月日**を記入及びマークしてください。正しくマークされていない場合は，採点されないことがあります。
   (5) **選択した問題**については，右の例に従って，**選択欄**の**問題番号**の(選)をマークしてください。マークがない場合は，採点の対象になりません。問1～問7について，6問以上マークした場合は，はじめの5問を採点します。問9～問13について，2問以上マークした場合は，はじめの1問を採点します。

〔問1，問3，問4，問6，問7，問9を選択した場合の例〕

選択欄					
問1	●	問8	●	問9	●
問2	(選)			問10	(選)
問3	●			問11	(選)
問4	●			問12	(選)
問5	(選)			問13	(選)
問6	●				
問7	●				

   (6) 解答は，次の例題にならって，解答欄にマークしてください。

〔例題〕　次の［　　］に入れる正しい答えを，解答群の中から選べ。

秋の情報処理技術者試験は，［ a ］月に実施される。

解答群　ア　8　　イ　9　　ウ　10　　エ　11

正しい答えは“ウ　10”ですから，次のようにマークしてください。

例題	a	(ア) (イ) ● (エ)

裏表紙の注意事項も，必ず読んでください。

〔問題一覧〕

●問 1～問 7（7 問中 5 問選択）

問題番号	出題分野	テーマ
問 1	ハードウェア	温度モニタ
問 2	データベース	コールセンターの対応記録管理
問 3	ネットワーク	CRC（巡回冗長検査）
問 4	情報セキュリティ	認証システム
問 5	ソフトウェア設計	部品の棚卸金額計算
問 6	IT サービスマネジメント	IT サービスマネジメントにおける個人情報の保護
問 7	システム戦略	子会社の業績評価

●問 8（必須問題）

問題番号	出題分野	テーマ
問 8	データ構造及びアルゴリズム	符号付き 2 進整数の乗算

●問 9～問 13（5 問中 1 問選択）

問題番号	出題分野	テーマ
問 9	ソフトウェア開発（C）	バスの到着待ち時間
問 10	ソフトウェア開発（COBOL）	有料自動車道路のインターチェンジ別売上と利用台数の集計
問 11	ソフトウェア開発（Java）	電子会議システム
問 12	ソフトウェア開発（アセンブラ）	ビット列を逆転するプログラム
問 13	ソフトウェア開発（表計算）	シャンプーの価格弾力性分析

共通に使用される擬似言語の記述形式

擬似言語を使用した問題では，各問題文中に注記がない限り，次の記述形式が適用されているものとする。

〔宣言，注釈及び処理〕

記述形式		説明
○		手続，変数などの名前，型などを宣言する。
/* 文 */		**文**に注釈を記述する。
処理	・**変数** ← **式**	**変数**に**式**の値を代入する。
	・**手続**(**引数**, …)	**手続**を呼び出し，**引数**を受け渡す。
	▲ **条件式** ▼ **処理**	単岐選択処理を示す。 **条件式**が真のときは**処理**を実行する。
	▲ **条件式** **処理 1** —— **処理 2** ▼	双岐選択処理を示す。 **条件式**が真のときは**処理 1** を実行し，偽のときは**処理 2** を実行する。
	■ **条件式** **処理** ■	前判定繰返し処理を示す。 **条件式**が真の間，**処理**を繰り返し実行する。
	■ **処理** ■ **条件式**	後判定繰返し処理を示す。 **処理**を実行し，**条件式**が真の間，**処理**を繰り返し実行する。
	■ **変数**: **初期値**，**条件式**，**増分** **処理** ■	繰返し処理を示す。 開始時点で**変数**に**初期値**（式で与えられる）が格納され，**条件式**が真の間，**処理**を繰り返す。また，繰り返すごとに，**変数**に**増分**（式で与えられる）を加える。

〔演算子と優先順位〕

演算の種類	演算子	優先順位
単項演算	+，−，not	高 ↑
乗除演算	×，÷，%	
加減演算	+，−	
関係演算	＞，＜，≧，≦，＝，≠	
論理積	and	
論理和	or	↓ 低

注　整数同士の除算では，整数の商を結果として返す。%演算子は，剰余算を表す。

〔論理型の定数〕

true，false

次の問1から問7までの7問については，この中から5問を選択し，選択した問題については，答案用紙の選択欄の(選)をマークして解答してください。

なお，6問以上選択した場合には，はじめの5問について採点します。

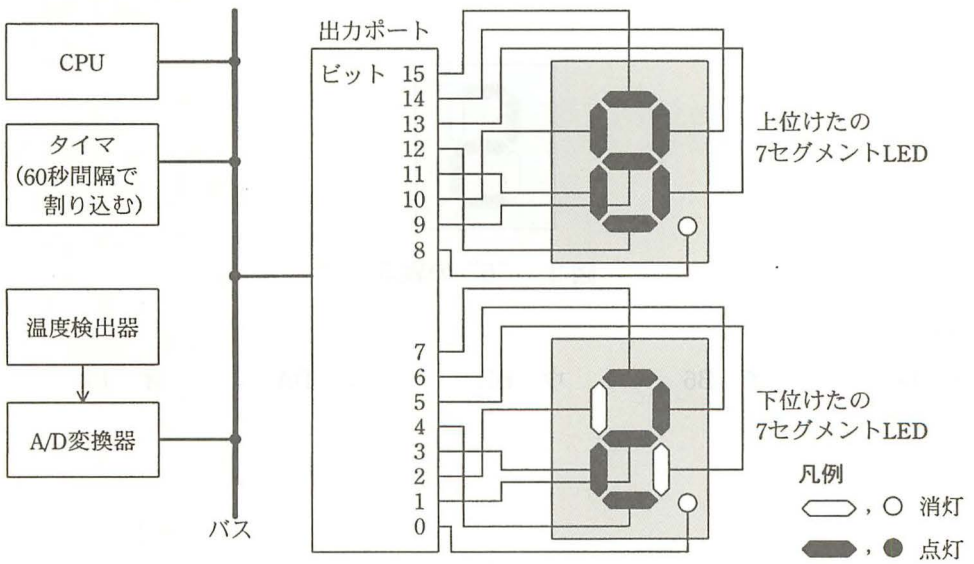
問1　温度モニタに関する次の記述を読んで，設問1～3に答えよ。

図1に温度モニタのシステム構成図の一部を示す。

温度の検出範囲は0～99℃とし，対応するA/D変換器の出力値は，0～99の16ビット符号なし2進数とする。このシステムは，タイマ割込み発生時に起動される割込みプログラムの中で温度検出器の出力値をA/D変換器を介して取り込み，2個の7セグメントLEDからなる表示器に表示する。1個の7セグメントLEDには10進数の数字1けたを表示する。検出された温度0～99℃に対して，表示器では“00”～“99”として表示する。

LEDの各セグメントは，対応する出力ポートのビットの値が1のとき点灯し，0のとき消灯する。7セグメントLEDに，“0”～“9”の数字を表示するために，対応する8ビットのデータを数字の字形に合わせて設定する。これらを形状データという。

割込みプログラムが起動するA/D変換の開始から，表示処理の完了までの時間は，タイマ割込み間隔に比べて十分短い。



注　“82”を表示した例である。ここで，出力ポートのビット0とビット8には常に0が設定され，小数点を表示するセグメントは消灯しているものとする。

図1　温度モニタのシステム構成図の一部

割込みプログラム中で，各機器に対するデータの読込み及び書込みは各機器に割り当てられたI/Oポート番号を指定して行う。7セグメントLEDが接続されている出力ポートにはI/Oポート番号1が，A/D変換器にはI/Oポート番号2と3がそれぞれ割り当てられている。表1に出力ポートとA/D変換器の動作概要を示す。

表1　出力ポートとA/D変換器の動作概要

	I/Oポート番号	動作概要
出力ポート	1	番号1のI/Oポートに形状データを書き込むと出力ポートの各ビットに値が設定され，LEDの各セグメントの点灯と消灯が行われる。
A/D変換器	2	番号2のI/Oポートに値1を書き込むとA/D変換が開始される。
		番号2のI/Oポートから読み込んだ値が0ならば，変換中を示す。
		番号2のI/Oポートから読み込んだ値が0以外ならば，A/D変換が完了し出力値が確定していることを示す。
	3	A/D変換完了後に番号3のI/Oポートから読み込むとA/D変換器の出力値（0～99）が得られる。

注　読込みデータ及び書込みデータはすべて16ビット

設問1　下位けたの7セグメントLEDに図2のように“5”を表示したい。出力ポートのビット 7～0 に設定すべき形状データとして正しい答えを，解答群の中から選べ。ここで，形状データはビット7を最上位ビットとする16進数で表記する。

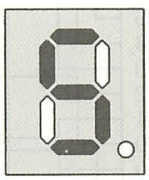


図2　“5”の表示

解答群

ア　66　　イ　B6　　ウ　BE　　エ　DA　　オ　F2

設問2　タイマ割込み発生時に起動される割込みプログラムについて，その処理の流れを図 3 に示す。図 3 の処理中の［　　］に入れる正しい答えを，解答群の中から選べ。

なお，処理中で使用している擬似命令の意味は表 2 のとおりとする。

表 2　擬似命令

命令の形式	命令の動作
INPUT　I/O ポート番号	番号で指定した I/O ポートに接続されている機器からデータを読み込み，GR に設定する。
OUTPUT　I/O ポート番号	GR に設定したデータを，番号で指定した I/O ポートに接続されている機器に書き込む。
CALL　DISPLAY	手続 DISPLAY を呼び出す。

注　GR は CPU のレジスタ

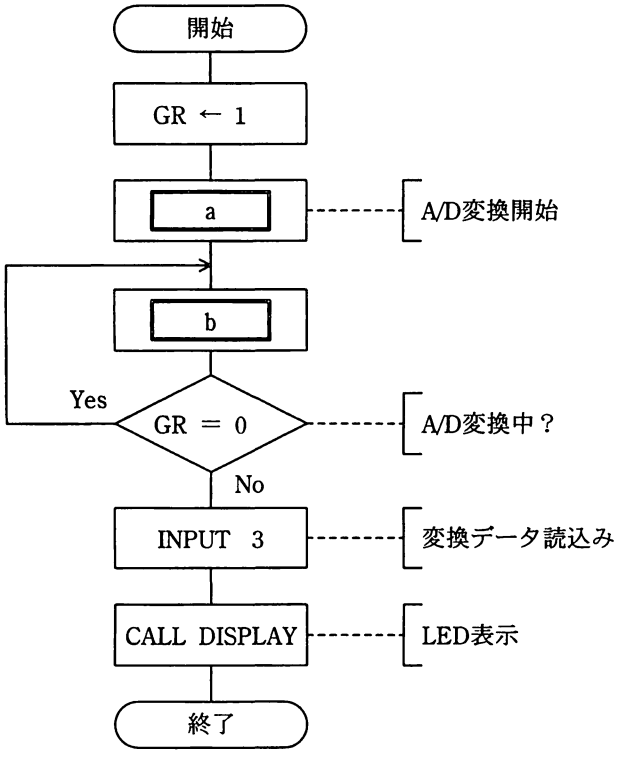


図 3　割込みプログラムの処理の流れ

解答群

ア　INPUT 1　　イ　INPUT 2　　ウ　OUTPUT 1

エ　OUTPUT 2　　オ　OUTPUT 3

設問3　割込みプログラムの中で呼び出す手続 DISPLAY の仕様と実行例は次のとおりである。実行例に関する記述中の［　　］に入れる正しい答えを，解答群の中から選べ。

〔手続 DISPLAY の仕様〕

DISPLAY は GR に設定されたデータが示す値を表示器に表示する。7 セグメント LED に“0”～“9”の数字を表示するために，それぞれ 8 ビットからなる 10 個の形状データを表としてプログラム中に保持している。

GR に設定されたデータは，検出した温度に対応する 0～99 の 16 ビット符号なし 2 進数であり，このまま出力ポートに設定しても意図した表示にはならない。GR の内容を図 4 のとおり 2 けたの BCD（2 進化 10 進数）に変換し，表を引き，各けたに対応した形状データを図 5 のとおりに GR に設定して出力ポートに書き込む。

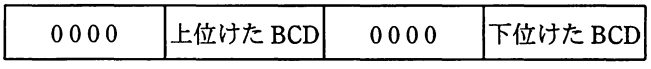


図4　BCD変換後のGRの内容

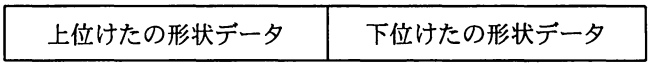


図5　出力ポートに書き込むGRの内容

〔手続 DISPLAY の実行例〕

図 5 の GR の内容が 16 進数表記で FEFC であったとき，BCD 値を格納していた図 4 の GR の内容は 16 進数表記で［　c　］であり，割込みプログラムから手続 DISPLAY に渡された GR の内容は 16 進数表記で［　d　］である。

cに関する解答群

ア　0008　　イ　0309　　ウ　0800　　エ　0903　　オ　0907

dに関する解答群

ア　0008　　イ　0027　　ウ　0050　　エ　005D　　オ　0061

問2　コールセンターの対応記録管理に関する次の記述を読んで，設問1～4に答えよ。

F社では，新しいソフトウェア製品の発売と同時に，そのソフトウェア製品に関する質問を受けるコールセンターを開設することにした。コールセンターでの対応内容は，すべてデータベースに記録する。

〔コールセンターの業務〕

(1) 製品を購入した利用者には，一意な利用者IDが発行されている。質問を受ける際は，この利用者IDを通知してもらう。

(2) 対応内容をデータベースに記録する際，その質問の原因を特定する種別を設定する。種別とは，“マニュアル不備”，“使用法誤解”などの情報である。それぞれの種別に対して一意に種別IDを割り当てる。

(3) データベースを検索し，過去に同じ種別IDをもつ類似の質問があった場合は，その受付番号を類似受付番号として記録しておく。

図1は，これらの業務を基に，データベースを構成するデータ項目を抽出したものである。下線付きの項目は主キーを表す。

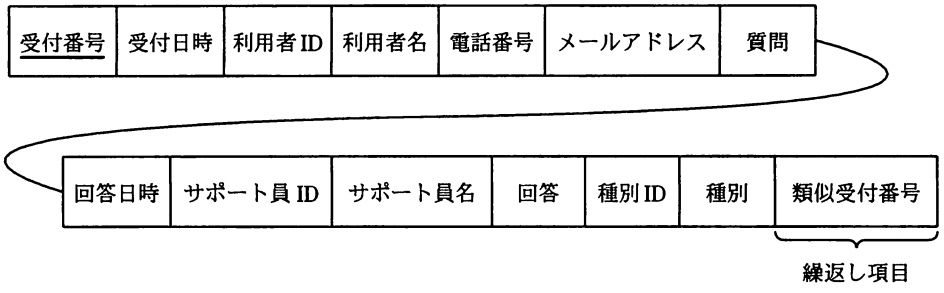


図1　データベースを構成するデータ項目

設問1　図1に示したデータ項目を正規化して図2に示す表を設計し，運用を始めた。実施した正規化に関する説明文の［　　］に入れる正しい答えを，解答群の中から選べ。

利用者表

利用者ID	利用者名	電話番号	メールアドレス

サポート員表

サポート員ID	サポート員名

種別表

種別ID	種別

類似表

受付番号	類似受付番号

対応表

受付番号	受付日時	利用者ID	質問	回答日時	サポート員ID	回答	種別ID

図2　正規化検討後の表

図1に示した状態は非正規形と呼ばれ，1事実1か所の関係が成立していないので，重複更新，事前登録，関係喪失などの問題がある。このため，第1正規化から順に第3正規化までを行うことにした。

まず，第1正規化の作業では，［　a　］。次に，第2正規化の作業では，［　b　］。そして，第3正規化の作業では，［　c　］。

解答群

ア　受付番号と類似受付番号の組合せを主キーとし，繰返し要素を排除した

イ　既に当該正規形に準じていたので，適用は不要だった

ウ　データ参照時の処理性能を考慮し，質問と回答を一つの表で管理するようにした

エ　利用者表，サポート員表及び種別表を作成し，主キー以外の項目における関数従属性を排除した

オ　類似表を作成し，主キーの一部における関数従属性を排除した

設問2　ある利用者から“オプションの指定方法”に関する質問を受けた。過去に類似の質問があったかどうかを確認するため，“オプション”というキーワードを含む質問をすべて抽出する。次の SQL 文の［　　］に入れる正しい答えを，解答群の中から選べ。

```
SELECT 対応表.受付番号，利用者表.利用者名，対応表.質問
   FROM 対応表，利用者表
   WHERE 対応表.利用者ID = 利用者表.利用者ID
         AND [                                    ]
```

解答群

ア　質問 ANY ('%オプション%')　　イ　質問 ANY ('_オプション_')

ウ　質問 IN ('%オプション%')　　エ　質問 IN ('_オプション_')

オ　質問 LIKE '%オプション%'　　カ　質問 LIKE '_オプション_'

設問3　製品のバージョンアップに当たり，コールセンターの対応記録を参考にして機能改善を検討することにした。種別が“使用法誤解”であった質問を抽出し，類似件数の多い順に表示する。次の SQL 文の［　　］に入れる正しい答えを，解答群の中から選べ。

```
SELECT 類似受付番号，COUNT(*) FROM 対応表，種別表，類似表
   WHERE [                                  ]
   ORDER BY COUNT(*) DESC
```

解答群

ア　対応表.種別ID = (SELECT 種別ID FROM 種別表 WHERE 種別 = '使用法誤解')
GROUP BY 類似表.類似受付番号

イ　対応表.種別ID = (SELECT 種別ID FROM 種別表 WHERE 種別 = '使用法誤解')
AND 対応表.受付番号 = 類似表.受付番号 GROUP BY 類似表.受付番号

ウ　対応表.種別ID = 種別表.種別ID AND 対応表.受付番号 = 類似表.受付番号
AND 種別表.種別 = '使用法誤解' GROUP BY 類似表.受付番号

エ　対応表.種別ID = 種別表.種別ID AND 対応表.受付番号 = 類似表.受付番号
AND 種別表.種別 = '使用法誤解' GROUP BY 類似表.類似受付番号

設問4　新たに提供する製品に関する質問を記録するために，現在の表に製品型番の列を追加して製品を識別できるようにする。表の拡張と同時に，これまで蓄積した情報の製品型番の列にはすべて“A001”を設定する。正しい SQL 文を，解答群の中から選べ。

解答群

ア　ALTER TABLE 対応表 ADD 製品型番 CHAR(4) DEFAULT 'A001' NOT NULL

イ　ALTER TABLE 対応表 MODIFY 製品型番 CHAR(4) DEFAULT 'A001' NOT NULL

ウ　CREATE TABLE 対応表 (製品型番 CHAR(4) DEFAULT 'A001')

エ　INSERT INTO 対応表 製品型番 VALUES 'A001'

問3　CRC（巡回冗長検査）に関する次の記述を読んで，設問1～3に答えよ。

CRC は，誤り検出方式の一つである。送信側でデータに誤り検出符号（以下，符号という）を付加して送信し，受信側で検査することによって，転送の際の誤りの有無を判断する。

設問1　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。

CRC を採用したパケット転送システムでは，送信側でパケットに符号が付加され，受信側で誤りの有無を検査する。受信側で誤りが検出されると，送信側に対して該当パケットの再送を要求する。100 個のパケットに格納されたデータの転送において，受信側が実際に受信したパケットが，再送されたパケットも含めて［　　］個であったとすると，受信したパケットの 20％から誤りが検出されたことになる。ここで，送信したパケットは必ず相手に届くものとする。また，パケットの再送要求は誤りなく届き，再送要求には必ず応じるものとする。

解答群

ア　100　　イ　102　　ウ　120　　エ　125

設問2　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。

任意の長さのビット列の符号を求める計算手順を次に示す。ここで，符号の長さは n ビットとする。

〔n ビットの符号を求める計算手順〕

(1)　左端及び右端のビットが 1 である（n+1）ビットのビットパターン（以下，マスクという）を定める。

(2)　符号計算対象のビット列の右端に n ビットの 0 を付加したビット列を作る。

(3)　(2)で作ったビット列に対して次の操作を行う。

①　ビット列の左端から調べ，最初に値が 1 であるビットの位置 p を見つける。

②　p を左端とし $p+n$ を右端とする部分ビット列に対し，マスクで排他的論理和（XOR）を取る。

③　ビット列の右端 n ビット以外がすべて 0 になるまで，①及び②を繰り返す。

(4) (3)の操作で得られたビット列の右端 n ビットが符号となる。

図に，マスクが 101 のときの符号（2 ビット）を計算する例を示す。

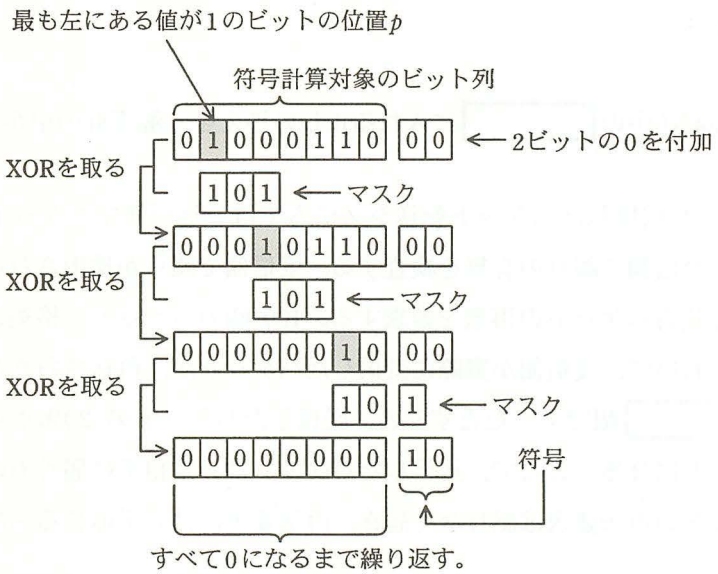


図　マスクが 101 のときの符号（2 ビット）を計算する例

マスク 101 で計算した，符号計算対象のビット列 0010 0110 の 2 ビットの符号は［　　］である。

解答群

ア　00　　イ　01　　ウ　10　　エ　11

設問 3　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。

誤りの有無の検査は，次の手順で行う。

〔誤りの有無の検査手順〕

(1) 受信したビット列に対して，送信側で符号の計算に利用したものと同じマスクを使い，〔n ビットの符号を求める計算手順〕の(3)と同じ処理を行う。

(2) 右端 n ビットの値によって，誤りの有無を判断する。

受信したビット列（符号が付加されたビット列）を，誤りの有無の検査手順に従って検査すると，誤りがなければ最後に残った右端 n ビットの値は［ a ］になる。このことは次の手順で説明できる。

〔手順〕

(1) 符号計算対象のビット列を一つの数値 D と見ると，符号 C は次の式で表せる。

式①　$(D \times 2^n) \oplus d_1 \oplus d_2 \oplus \cdots \oplus d_m = C$

ここで，⊕はXORを表し，m はXORの繰返し回数とする。d_i $(1 \leqq i \leqq m)$ はマスクに対応するビット列である。図の例では，d_1＝0101000000，d_2＝0001010000，d_3＝0000001010 である。

(2) 〔誤りの有無の検査手順〕で得られた結果の右端 n ビットの値 T は，次の式で表せる。

式②　$(D \times 2^n) \oplus C \oplus d_1 \oplus d_2 \oplus \cdots \oplus d_m = T$

(3) 式②を変形すると次の式となる。

式③　$(D \times 2^n) \oplus d_1 \oplus d_2 \oplus \cdots \oplus d_m \oplus C = T$

(4) 式①と式③によって，

［ b ］$= T$

となる。

マスク 101 で計算した符号を右端に付加したビット列 1001001101 を受信した。このビット列には［ c ］。

aに関する解答群

ア　すべてのビットが0　　イ　すべてのビットが1

ウ　符号と同じ　　エ　符号の各ビットを反転させたものと同じ

bに関する解答群

ア　$2^n - 1$　　イ　C　　ウ　$C \oplus (2^n - 1)$　　エ　$C \oplus C$

cに関する解答群

ア　誤りが含まれる　　イ　誤りは含まれない

ウ　誤りが含まれるか否かは判断できない

問4　認証システムに関する次の記述を読んで，設問に答えよ。

複数のクライアントと複数のアプリケーションサーバ（以下，APサーバという）が接続されているネットワークにおいて，単純な認証システムを利用する場合について，ここでは二つの問題点を取り上げる。

① 利用者は，使用するクライアントから各APサーバにログインするごとに，利用者IDとパスワードの入力操作を行わなければならない。

② クライアントとAPサーバとの間の通信データの横取りと偽造によって，APサーバのサービスが不正に利用される危険性がある。

ここで，これらの問題を改善するための認証システム（以下，新認証システムという）を考える。

なお，ここでは，これらの問題に直接関連しない仕様については，その記述を省略する。

〔新認証システムによる問題点の解消〕

問題点①に対しては，利用者が一度，利用者IDとパスワードをクライアントに入力して認証を受ければ，そのクライアントと各APサーバ間での認証は，利用者を介さないで済むように改善する。

このために，チケットと呼ぶ認証データを用いる。チケットは，クライアントに対して発行され，そのクライアントは，APサーバの認証を得るとき，発行されたチケットをAPサーバに送信する。

問題点②に対しては，APサーバに送信されたチケットが，チケットの発行を受けたクライアントから送られてきたものであることを，APサーバが確認できるよう，チケットとは別に認証子と呼ぶ認証データを用いる。

図1に新認証システムの構成を示す。

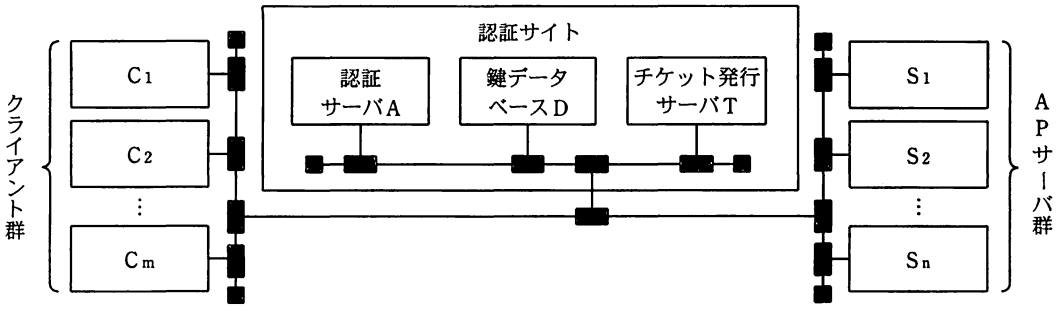


図1　新認証システムの構成

〔新認証システムの構成と方式についての説明〕

(1) 新認証システムでは，共通鍵暗号方式によって，通信データを暗号化する。以下，共通鍵を鍵という。

(2) 認証サイトは，認証サーバ，鍵データベース及びチケット発行サーバで構成する。

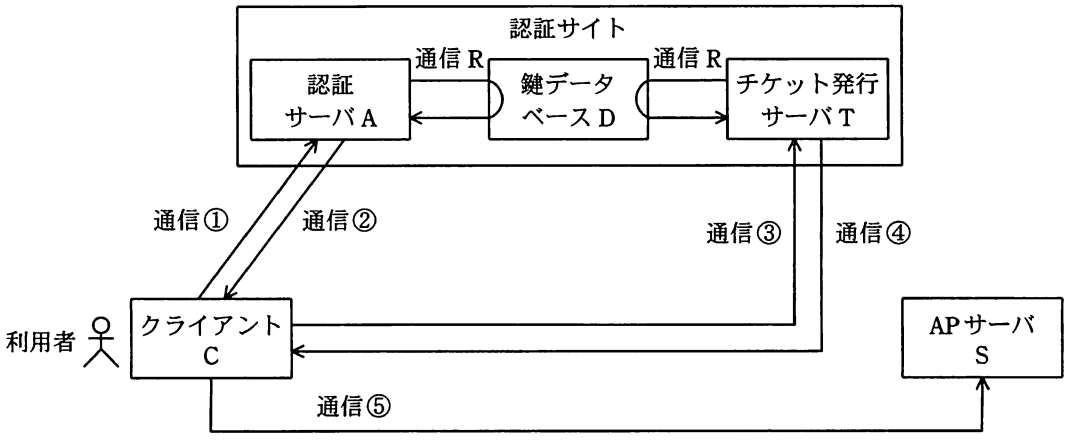
(3) チケット発行サーバの鍵は，チケット発行サーバ自体と鍵データベースに登録されている。

(4) 各 AP サーバの鍵は，それぞれの AP サーバ自体と鍵データベースに登録されている。

(5) 利用者の鍵は，利用者のパスワードから計算して決められ，鍵データベースに登録されている。クライアントには，利用者が入力したパスワードから計算した鍵が，利用者がクライアントの利用を終了するまで，一時的に保持される。

〔認証のための通信の例〕

図 2 は，利用者が，クライアント C から目的の AP サーバ S にアクセスする場合の認証の流れを示す。



注 通信 R は，鍵データベースを参照していることを表す。

図 2　認証の流れの例

認証は，次の3段階で行われる。ここで，enc(x)は，xを暗号化したものを表す。

第1段階は，クライアントCがチケット発行サーバTにチケットを要求するためのチケット（以下，チケット発行サーバT用チケットという）の認証サーバAへの要求（図2中の通信①）と，その発行（図2中の通信②）である。

通信①では，クライアントCは，次のデータを認証サーバAに送信する。

データ	データの説明
ID_C	利用者IDである。
ID_T	チケット発行サーバTのIDである。

通信②では，認証サーバAは，次のデータをクライアントCに応答する。

データ	データの暗号化に用いた鍵	データの説明
enc(KEY_{CT})	利用者の鍵 KEY_C	KEY_{CT}は，クライアントCとチケット発行サーバTとの間（以下，C-T間という）の通信データの暗号化に用いる鍵であり，C-T間のセッション鍵という。
enc($TICKET_{CT}$)	a	$TICKET_{CT}$は，チケット発行サーバT用チケットである。$TICKET_{CT}$はKEY_{CT}を含む。これによって，チケット発行サーバTにクライアントC経由でKEY_{CT}を安全に渡すことができる。

第2段階は，クライアントCがAPサーバSにアクセスするためのチケット（以下，APサーバS用チケットという）のチケット発行サーバTへの要求（図2中の通信③）と，その発行（図2中の通信④）である。

通信③では，クライアントCは，次のデータをチケット発行サーバT に送信する。

データ	データの暗号化に用いた鍵	データの説明
enc($TICKET_{CT}$)	a	enc($TICKET_{CT}$)は，クライアントCでは復号できない。クライアントCはチケット発行サーバTにそのまま送信し，チケット発行サーバTが復号する。
enc($AUTH_{C1}$)	b	認証子$AUTH_{C1}$は，クライアントCが生成する。
enc(ID_S)	b	ID_Sは，APサーバSのIDである。

チケット発行サーバTは，$TICKET_{CT}$を送信したのが間違いなくクライアントCであることを$TICKET_{CT}$と$AUTH_{C1}$から確認する。確認ができたとき，通信④では，チケット発行サーバTは，次のデータをクライアントCに応答する。

データ	データの暗号化に用いた鍵	データの説明
enc(KEY_{CS})	b	KEY_{CS}は，クライアントCとAPサーバSとの間（以下，C-S間という）の通信データの暗号化に用いる鍵であり，C-S 間のセッション鍵という。データの暗号化に用いた鍵は，チケット発行サーバTが，通信③で受け取った enc($TICKET_{CT}$)から取り出したものである。
enc($TICKET_{CS}$)	c	$TICKET_{CS}$は，APサーバS用チケットである。$TICKET_{CS}$は，KEY_{CS}を含む。これによって，APサーバSにKEY_{CS}をクライアントC経由で安全に渡すことができる。

第3段階は，APサーバS用チケットの提示である（図2中の通信⑤）。

通信⑤では，クライアントCは，次のデータをAPサーバSに送信する。

データ	データの暗号化に用いた鍵	データの説明
enc($TICKET_{CS}$)	c	enc($TICKET_{CS}$)は，クライアントCでは復号できない。クライアントCはAPサーバSにそのまま送信し，APサーバSが復号する。
enc($AUTH_{C2}$)	d	認証子$AUTH_{C2}$は，クライアントCが生成する。

APサーバSは，$TICKET_{CS}$を送信したのが間違いなくクライアントCであることを$TICKET_{CS}$と$AUTH_{C2}$から確認する。確認ができたとき，利用者は，クライアントCから，APサーバSへのアクセスが許可される。

設問　本文中の［　　］に入れる正しい答えを，解答群の中から選べ。

aに関する解答群

ア　C-T間のセッション鍵 KEY_{CT}　　イ　チケット発行サーバTのID ID_T

ウ　チケット発行サーバTの鍵 KEY_T　　エ　利用者ID ID_C

b，cに関する解答群

ア　APサーバSのID ID_S　　イ　APサーバSの鍵 KEY_S

ウ　C-S間のセッション鍵 KEY_{CS}　　エ　C-T間のセッション鍵 KEY_{CT}

オ　チケット発行サーバTの鍵 KEY_T

dに関する解答群

ア　APサーバSの鍵 KEY_S　　イ　C-S間のセッション鍵 KEY_{CS}

ウ　C-T間のセッション鍵 KEY_{CT}　　エ　チケット発行サーバTの鍵 KEY_T

問5　部品の棚卸金額計算に関する次の記述を読んで，設問1～5に答えよ。

製造業が，製品製造のために購入する部品は，同じ部品であっても，その調達時期によって購入時の単価（以下，購入単価という）が変動する場合がある。したがって，期末時点で在庫となっている部品の棚卸金額を求める場合，個々の購入単価と数量を使って計算する。

しかし，大量生産を行う製造業の T 社では，必要な部品の種類と数量が非常に多い。購入した部品を受け入れて在庫とし，在庫から部品を払い出して製品を製造するまで，部品の一つ一つを管理することは難しく，実際に在庫となっている個々の部品の購入単価から金額を計算することは困難である。

部品を受け入れた日付を受入日付，受け入れた数量を受入数量，払い出した日付を払出日付，払い出した数量を払出数量という。

T 社では，先入先出法によって，購入した部品の棚卸金額を計算している。先入先出法とは，受入日付が最も古い部品から順に払出しが行われたものとみなして，購入した部品の棚卸金額を計算する方法である。

〔棚卸金額計算の処理〕

図1に，T 社が購入した部品の棚卸金額計算処理の流れを示す。この処理で扱うファイルは，すべて順ファイルである。

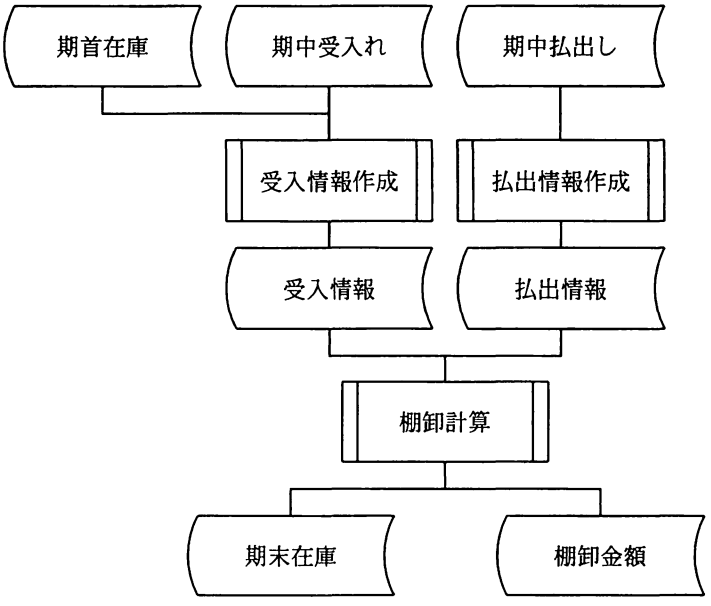


図1　T 社が購入した部品の棚卸金額計算処理の流れ

(1) 受入情報作成処理では，期首在庫ファイルと，期中の部品の受入れを記録した期中受入れファイルの二つを整列して併合し，受入情報ファイルとして出力する。

受入情報ファイルのレコード様式は，次のとおりである。部品は，部品番号で管理している。

部品番号	受入日付	購入単価	受入数量

図2 受入情報ファイルのレコード様式

なお，T社の場合，同じ部品を同じ日に複数回受け入れることはない。

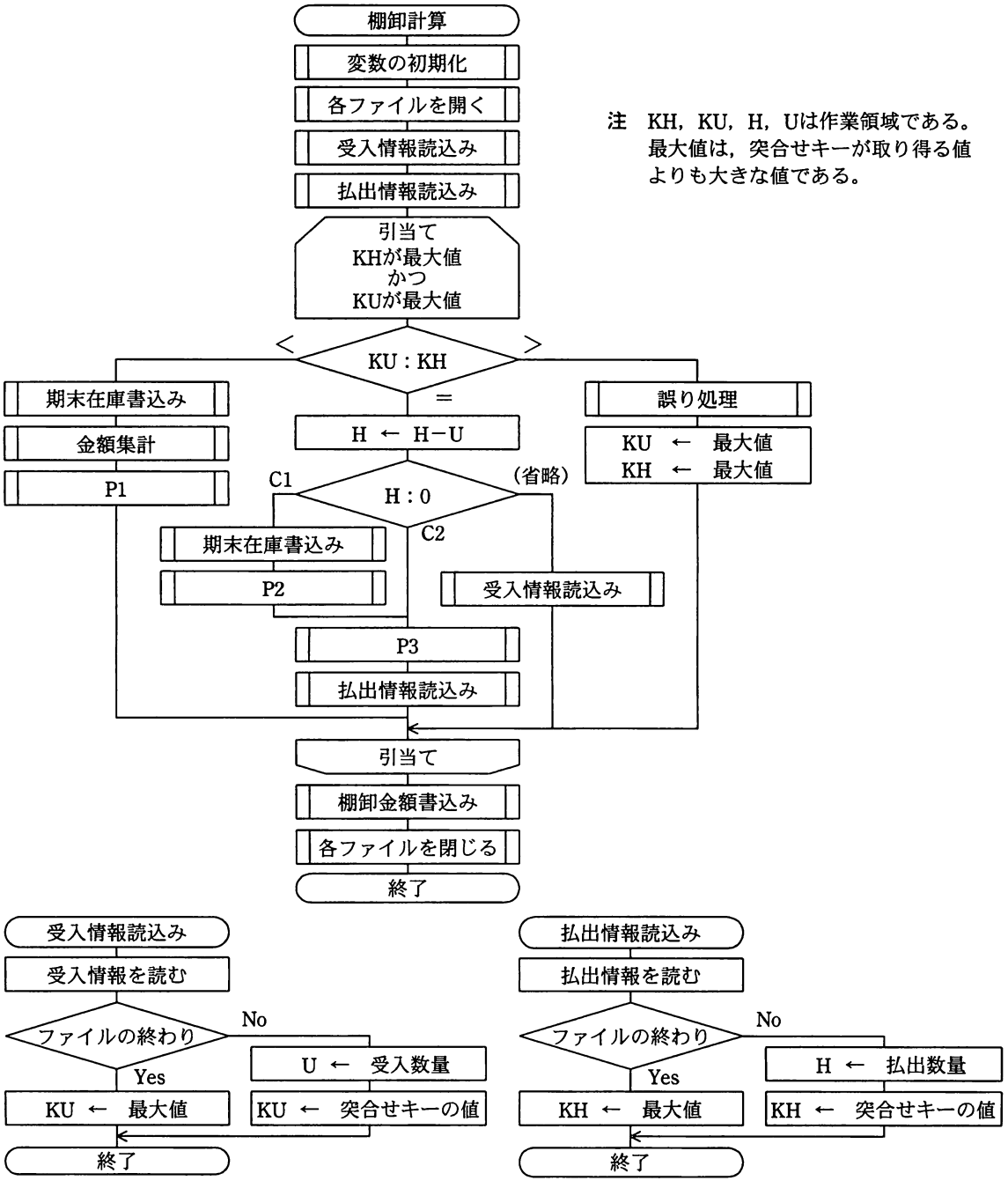
(2) 払出情報作成処理では，期中の部品の払出しを記録した期中払出しファイルを用いて払出数量の集計を行い，払出情報ファイルを出力する。

(3) 棚卸計算処理では，突合せを行い，期末在庫ファイルと棚卸金額ファイルを作成する。

図3に，棚卸計算処理の流れを示す。

受入情報ファイルと払出情報ファイルとの突合せによって，払出情報ファイルのレコードの払出数量分を，受入情報ファイルの受入日付が古いレコードから順に引き当てていく。引き当てられずに残った受入情報ファイルのレコードの数量と購入単価を使って，期末在庫ファイルと棚卸金額ファイルのレコードを作成する。

数量が0となった受入情報ファイルのレコードは，期末在庫ファイルには出力しない。また，期末在庫ファイルは，次期の期首在庫ファイルとなる。期末在庫ファイルと期首在庫ファイルは，受入情報ファイルと同じレコード様式である。



注　KH，KU，H，Uは作業領域である。
最大値は，突合せキーが取り得る値
よりも大きな値である。

図3　棚卸計算処理の流れ

設問1　図1中の受入情報作成処理では，期首在庫ファイル及び期中受入れファイルを昇順に整列して併合し，受入情報ファイルとして出力する。整列に最低限必要なキー項目とその並びとして正しい答えを，解答群の中から選べ。ただし，解答群の項目の並びは，左の項目の方が整列の優先度が高い。

解答群

ア　受入日付
イ　受入日付，受入数量
ウ　受入日付，部品番号
エ　部品番号
オ　部品番号，受入日付
カ　部品番号，購入単価

設問2　図1中の払出情報ファイルに最低限必要な項目として正しい答えを，解答群の中から選べ。

解答群

ア　受入日付，払出数量
イ　受入日付，払出数量，払出日付
ウ　受入日付，払出数量，部品番号
エ　払出数量，払出日付
オ　払出数量，払出日付，部品番号
カ　払出数量，部品番号

設問3　図1中の払出情報作成処理では，払出数量の集計を行う。この集計処理に最低限必要なキー項目として正しい答えを，解答群の中から選べ。

解答群

ア　受入日付
イ　受入日付，払出日付
ウ　受入日付，部品番号
エ　払出日付
オ　払出日付，部品番号
カ　部品番号

設問4　図3中のP1～P3のうち，“受入情報読込み”が入るすべての箇所の組合せとして正しい答えを，解答群の中から選べ。

解答群

ア　P1	イ　P2	ウ　P3
エ　P1，P2	オ　P1，P3	カ　P2，P3
キ　P1，P2，P3		

設問5　図3中の条件C1，C2に入れる組合せとして正しい答えを，解答群の中から選べ。

解答群

	C1	C2
ア	＝	＜
イ	＝	＞
ウ	＜	＝
エ	＜	＞
オ	＞	＝
カ	＞	＜

問6　IT サービスマネジメントにおける個人情報の保護に関する次の記述を読んで，設問1，2に答えよ。

X社では，約1万人の個人顧客向けに会員制の通信販売を行っている。注文はインターネットや電話で販売部が受け付け，商品と請求書を宅配便で発送する。顧客からの問合せなども，一括して販売部で受け付けている。顧客情報は，X社の基幹サーバで一括管理している。

X社の顧客管理部では，次の業務①～③を行っている。業務①は，顧客への定期的な取引状況の報告である。業務②及び③は，販売部での対応範囲を超えた問合せであり，顧客管理部に転送されてくる。

業務①　取引明細など顧客あて郵便物の印刷・発送
業務②　取引の詳細な内容や会員登録状況の照会など顧客からの問合せへの対応
業務③　住所変更や退会など顧客からの訂正・削除・利用停止などの依頼への対応

業務①の印刷・発送の作業で使っている機器が老朽化し，最近は故障が増えている。X社の役員から“機器更新などの対応も含めて業務①～③の外部委託を検討するように”との指示を受けた顧客管理部長は，自身をリーダとして，顧客管理部，システム部及びリスク管理部の管理者からなる委託検討プロジェクト（以下，プロジェクトという）を発足させた。

設問1　プロジェクトでは，まず委託元であるX社内での対応を検討することにした。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。

プロジェクトでは，“個人情報の保護に関する法律についての経済産業分野を対象とするガイドライン”（以下，ガイドラインという）に沿った管理水準で検討を進めることにした。

個人データの取扱いの委託に当たっては，法とガイドラインに基づく安全管理措置（表にその概要を示す）を委託先に遵守させるために必要な［　a　］し，業務開始後は委託元が［　b　］する必要がある。

表　安全管理措置の概要

安全管理措置	安全管理措置の内容（概要）
組織的安全管理措置	・規程等の整備と規程等に従った運用 ・安全管理措置の評価，見直し及び改善
人的安全管理措置	・非開示契約の締結 ・教育，訓練の実施
物理的安全管理措置	・入退館，入退室管理の実施 ・盗難等の防止 ・機器・装置等の物理的な保護
技術的安全管理措置	・情報システムへのアクセス制御 ・不正ソフトウェア対策 ・情報システムの監視

業務①～③について，リスクに対する費用対効果の面から総合的に検討した。その結果，業務①については機器の買換えではなく外部委託とすることにした。

一方，業務②及び③については社内処理を継続することにした。その理由として，例えば表の安全管理措置のうち技術的安全管理措置について，［　c　］などの対応を考慮する必要があったことによる。

プロジェクトとしてこのように結論を出し，役員に報告した。

a，bに関する解答群

ア　委託先の運用担当者を任命　　イ　委託先の従業者を監督

ウ　契約を締結　　エ　個人データの取扱状況を監督

cに関する解答群

ア　X社内の個人データを委託先からアクセスする際の安全性を確保するためにシステムを改修する

イ　X社の管理水準を満たす専用作業室を委託先に新たに確保する

ウ　X社の従業者が委託先の個人情報保護管理者となるために委託先で常駐する

エ　X社の従業者が委託先のシステム運用方法の指揮・命令体制を整備する

設問2　プロジェクトでは，次に委託先での安全管理措置の現状を調査することにした。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。

役員の承認を受けて業務①の外部委託が決まり，委託先は印刷・発送業のY社に内定した。また，Y社での印刷・発送に必要なデータだけを，X社の基幹サーバから抽出してY社のファイルサーバへネットワーク経由で暗号化して送信する方式とした。

個人データの安全管理措置は，現在のX社での管理水準をY社においても維持する必要がある。そこで，プロジェクトではX社における規程や手続の要件を文書にまとめてY社の運用管理者に提示した。また，サービス開始前にY社の作業現場に行って現状調査を実施することで合意した。

後日，プロジェクトのメンバがY社の作業現場を訪問し，Y社の運用管理者と一緒に現状調査を実施した。現状調査の結果，次の問題点①～④が発見された。

〔問題点〕

① 印字不良などで廃棄する，個人情報を含む印刷物は，鍵の付いた個人情報専用の廃棄箱に入れて保管し，定期的に廃棄処分をしている。しかし，この廃棄箱の鍵の保管管理が十分でなく，管理者の帰宅後はだれでも鍵を使用できる状態にある。

② ファイルサーバに対する設定変更などの特権操作は，電算室内の運用管理者席にあるシステム保守用端末だけで実行できる。しかし，この端末で行われた特権操作の内容は記録していない。

③ ファイルサーバへのログインは利用者IDとパスワードで認証しているが，認証後は，サーバ内のどのファイルも参照が可能である。

④ ファイルサーバへのログインは，すべて記録されるが，個々のファイルへのアクセスは，機能上の理由で記録していない。

プロジェクトのメンバとY社の運用管理者は，問題点について次のとおり確認した。

(1) 問題点①～④のうち，問題点［　d　］を除くほかの三つの問題点は，技

術的安全管理措置の問題である。

(2) 問題点①について，この問題を放置しておくと［ e ］というリスクがある。

(3) 問題点②及び④について，これらの問題を放置しておくと［ f ］というリスクがある。

Y社では，この現状調査の結果をガイドラインに沿ってまとめて改善計画を作成すること，及び改善計画の実施結果をX社に報告することを確約した。

dに関する解答群

ア ①　　イ ②　　ウ ③　　エ ④

e，fに関する解答群

ア 個人情報を含む媒体を持ち出せる　　イ 誤操作や不正操作の発見が困難になる

ウ システム障害の原因となる　　エ 利用者認証機能をう回できる

問7　子会社の業績評価に関する次の記述を読んで，設問1～3に答えよ。

持株会社であるA社には，国内外に製造業を営む4社の子会社がある。B社とC社は日本国内の子会社，D社はY国にある在外子会社，E社はZ国にある在外子会社である。

A社では，子会社の1年間の業績を評価して，業績の良好な子会社には経営優秀賞を授与し，業績の落ち込んでいる子会社には要調査対象として詳細な経営分析を行っている。

経営優秀賞は，当該年度の営業利益と経常利益がともに黒字であり，かつ，前年度より売上高，営業利益率及び総資産利益率 ROA（Return on Assets）のすべてが向上した子会社に授与される。要調査対象は，当該年度の売上高と営業利益がともに前年度より10％以上減少した子会社である。

このたび，2009年度の経営優秀賞と要調査対象の子会社を選出することになり，決算用の資料を基に，子会社の2008年度と2009年度の業績を表1と表2にまとめた。ここで，D社とE社の業績は日本円に換算して記されている。

表1　子会社の2008年度の業績

単位 百万円

	B社	C社	D社	E社
売上高	32,000	14,000	30,000	28,000
製造原価・販管費	28,800	11,900	28,500	29,400
営業利益	3,200	2,100	1,500	−1,400
営業外損益	−200	100	−100	280
経常利益	3,000	2,200	1,400	−1,120
総資産	30,000	8,000	28,000	25,200

注　販管費は，販売費と一般管理費の略称である。

表2　子会社の2009年度の業績

単位 百万円

	B社	C社	D社	E社
売上高	34,000	12,400	23,100	36,000
製造原価・販管費	30,700	10,912	21,910	36,300
営業利益	3,300	1,488	1,190	−300
営業外損益	−100	812	210	300
経常利益	3,200	2,300	1,400	0
総資産	31,000	7,900	21,000	30,000

設問1　経営優秀賞の選出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。

経営優秀賞の選出に当たり，“当該年度の営業利益と経常利益がともに黒字である”という条件に基づいて各社を評価すると，［　a　］が候補から除外される。

次に，子会社の売上高，営業利益率及び ROA の指標値を前年度と対比することで，子会社の業績を評価する。

営業利益率，ROA 及び経常利益率は，次の式で計算する。

営業利益率　＝　営業利益／売上高

ROA　＝　経常利益／総資産

経常利益率　＝　経常利益／売上高

ROA は，経常利益率と［　b　］の積で表すことができる。営業利益率と ROA の値を前年度と対比することで，営業利益率と経常利益率の変化によって会社の収益性を，［　b　］の変化によって会社の［　c　］を評価することができる。

aに関する解答群

ア　B社　　イ　C社　　ウ　D社　　エ　E社

bに関する解答群

ア　財務レバレッジ（総資産／自己資本）　イ　自己資本比率（自己資本／総資産）

ウ　総資産回転期間（総資産／売上高）　エ　総資産回転率（売上高／総資産）

cに関する解答群

ア　安全性　　イ　効率性　　ウ　生産性　　エ　成長性

設問2　在外子会社の業績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。ここで，Y 国の現地通貨は Y ドル，Z 国の現地通貨は Z ドルであり，各通貨を日本円に換算するときの為替レートを表3に示す。

表3　日本円と Y ドル，Z ドルの為替レート

	2008 年度	2009 年度
Y国	100 円／Y ドル	70 円／Y ドル
Z国	140 円／Z ドル	150 円／Z ドル

D 社の 2009 年度の業績を 2008 年度と比較する場合，日本円に換算して比較すると，売上高及び営業利益の両方が下がっているが，現地通貨で比較すると［ d ］いる。D 社や E 社のような在外子会社の場合，為替レートの変動によって，現地通貨と日本円換算とで［ e ］の指標値は変化する。A 社では，在外子会社の業績を評価するために，日本円換算ではなく，現地通貨を用いている。

なお，経常利益率，［ f ］は為替レートが変動しても指標値は変わらないので，現地通貨で計算しても日本円換算でも同じ値である。

dに関する解答群

ア　売上高及び営業利益が上がって

イ　売上高は上がっているが，営業利益は下がって

ウ　売上高は下がっているが，営業利益は上がって

e，fに関する解答群

ア　売上高及び ROA　　イ　売上高及び営業利益

ウ　売上高及び営業利益率　　エ　営業利益及び ROA

オ　営業利益及び営業利益率　　カ　営業利益率及び ROA

設問 3　2009 年度の経営優秀賞及び要調査対象の子会社選出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。

日本国内の子会社は日本円で，在外子会社は現地通貨で，子会社 4 社の業績を評価した結果，最終的に経営優秀賞は［ g ］となり，要調査対象子会社は［ h ］となった。

解答群

ア　B社　　イ　C社　　ウ　D社　　エ　E社　　オ　該当なし

〔 メ モ 用 紙 〕

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。

プログラム1及びプログラム2は，いずれも二つの整数M，Nを受け取り，その積M×Nの値を返す。積は，加減算とシフト演算を使って求める。M，N及び求めた積は，いずれも符号付き2進数の整数で，負数は2の補数で表現する。

〔プログラム1の説明〕

プログラム1が受け取るM，Nは，いずれも4ビットで，各値の範囲は−8〜7である。求めた積M×Nは，8ビットで返す。ただし，プログラム1中では，Mを5ビットに，Nを8ビットに拡張して処理を行う。

Rは13ビットの作業用変数であり，最下位から順にビット番号を0，1，…とし，最上位（符号ビット）のビット番号を12とする。Rは，指定した一部の範囲（例えば，ビット番号12〜8の上位5ビット）だけを符号付き2進数とみなして部分的な算術演算ができる。また，値の検査のために指定したビット番号の内容を取り出すこともできる。

なお，⑥の行の加算では，けたあふれが起きても無視する。

N=3として，M=5とM=−5の場合の処理過程を，それぞれ図1，2に示す。図1，2中の記号①〜⑧は，プログラム1の①〜⑧の行の処理と対応している。

〔プログラム1〕

```
      ○プログラム1（M，N）
      ○符号付き2進整数型： M，N，R
      ○整数型： L
① →  ・ Mを5ビットの符号付き2進整数に拡張
② →  ・ Nを8ビットの符号付き2進整数に拡張
③ →  ・ Rのビット番号7〜0にNを複写
④ →  ・ Rのビット番号12〜8を0で初期化
      ■ L: 1，L ≦ 8，1
⑤ →  ▲ Rのビット番号0のビットが1
⑥ →  ▼ ・ Rのビット番号12〜8の内容にMの値を加算
⑦ →  ・ Rの全13ビットを右に1ビット算術シフト /* 空いたビット位置には */
                                              /* 符号と同じものが入る */
⑧ →  ・ return（ Rのビット番号7〜0の内容 ）   /* 返却値（括弧内）を返す */
```

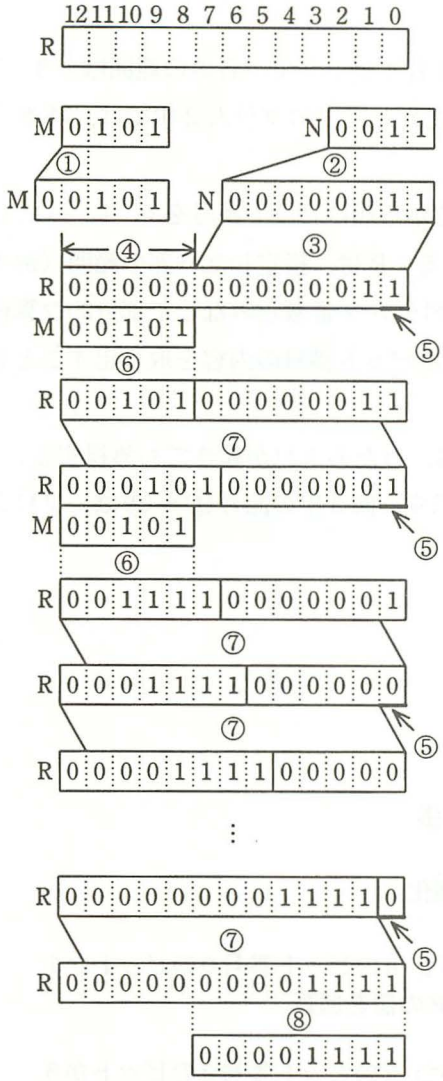
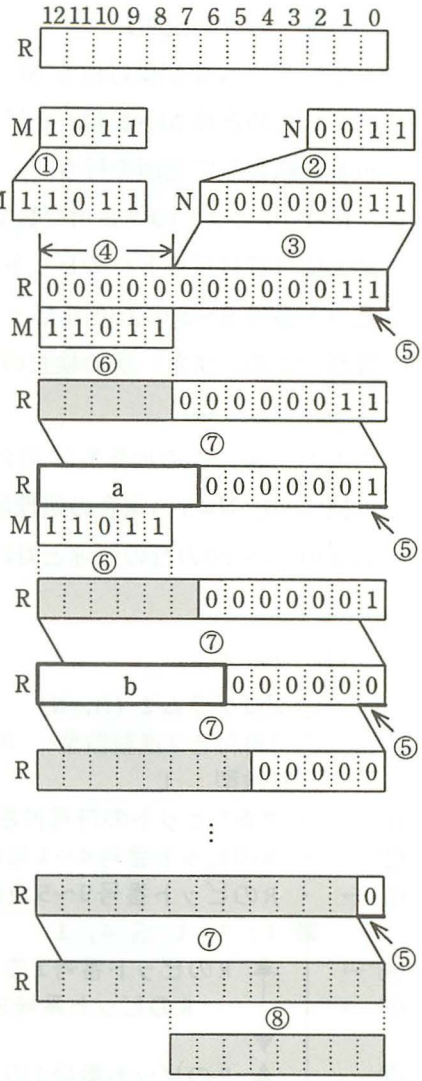


図１　プログラム１の実行例
（M＝５，N＝３）



注　網掛けの部分は，表示していない。

図２　プログラム１の実行例
（M＝－５，N＝３）

〔プログラム2の説明〕

プログラム2が受け取るM，Nは，いずれも4ビットで，各値の範囲は−8〜7である。求めた積M×Nは，8ビットで返す。ただし，プログラム2中では，Mを5ビットに拡張して処理を行う。

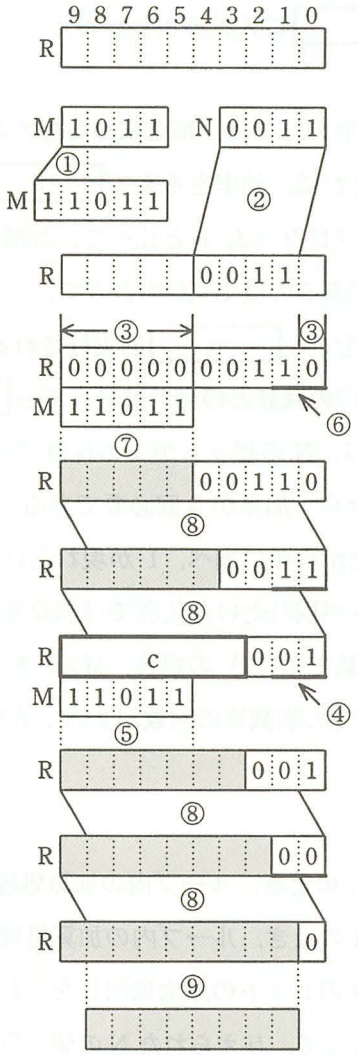
Rは10ビットの作業用変数であり，最下位から順にビット番号を0，1，…とし，最上位（符号ビット）のビット番号を9とする。Rは，指定した一部の範囲（例えば，ビット番号9〜5の上位5ビット）だけを符号付き2進数とみなして部分的な算術演算ができる。また，値の検査のために指定したビット番号の内容を取り出すこともできる。

なお，⑤の行の加算及び⑦の行の減算では，けたあふれが起きても無視する。

M=−5，N=3の場合の処理過程を図3に示す。図3中の記号①〜⑨は，プログラム2の①〜⑨の行の処理と対応している。

〔プログラム2〕

```
      ○プログラム2（M，N）
      ○符号付き2進整数型： M，N，R
      ○整数型： L
①→  ・ Mを5ビットの符号付き2進整数に拡張
②→  ・ Rのビット番号4〜1にNを複写
③→  ・ Rのビット番号9〜5及び0を0で初期化
      ■ L: 1，L ≦ 4，1
④→    ▲ Rのビット番号1のビットが0 かつ Rのビット番号0のビットが1
⑤→    ・ Rのビット番号9〜5の内容にMの値を加算
      ▼
⑥→    ▲ Rのビット番号1のビットが1 かつ Rのビット番号0のビットが0
⑦→    ・ Rのビット番号9〜5の内容からMの値を減算
      ▼
⑧→    ・ Rの全10ビットを右に1ビット算術シフト /* 空いたビット位置には */
                                              /* 符号と同じものが入る */
      ■
⑨→  ・ return（ Rのビット番号8〜1の内容 ）   /* 返却値（括弧内）を返す */
```



注　網掛けの部分は，表示していない。

図３　プログラム２の実行例
（M＝－５，N＝３）

設問１　図２及び図３中の［　　］に入れる正しい答えを，解答群の中から選べ。

aに関する解答群

ア　0 0 0 1 0 1　　イ　0 1 1 0 1 1　　ウ　1 0 0 1 0 1　　エ　1 1 1 0 1 1

b，cに関する解答群

ア　0 0 0 0 1 0 0　　イ　0 0 0 0 1 0 1　　ウ　0 0 0 1 1 1 1

エ　0 0 1 0 0 0 1　　オ　1 1 1 0 0 0 1　　カ　1 1 1 1 0 1 1

設問2　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。

(1)　プログラム1の③の行では，積を求めるためのRの下位8ビットにNの値を設定している。それは，効率を考えて［ d ］ためである。

(2)　プログラム2はプログラム1と比べて，加減算の回数が少なくて済む場合がある。特に効果があるのはN<0のときで，このとき，プログラム1では，⑥の行の加算は最低でも［ e ］回実行されるが，プログラム2では，⑤の行の加算と⑦の行の減算との合計は最高でも［ f ］回で済む。

(3)　プログラム1では，Nのビットが1の位置で加算をするので，例えばN=7（2進数で0111）なら，加算が3回必要である。一方，プログラム2では，Nの各ビットを最下位から順に調べ，1が現れたけた位置では減算をし，次に1の並びが途切れて0が現れたけた位置では加算をする，という処理を繰り返す。例えばN=7（2進数で0111）の場合，M×7をM×(［ g ］)として計算をするので，2進数での加減算の回数が2回で済む。

dに関する解答群

ア　与えられたMの値が1のとき，ループ内の加算処理の実行回数を0にできる

イ　与えられたNの値が1のとき，ループ内の加算処理の実行回数を0にできる

ウ　中間結果のシフトとNのビットの順次取出しを，1回のシフトで済ませる

エ　中間結果のシフトによって，与えられたNの値の符号検査をなくせる

e，fに関する解答群

ア　1	イ　2	ウ　3
エ　4	オ　5	カ　6

gに関する解答群

ア　-2^0+2^3　　イ　-2^0+2^4　　ウ　-2^1+2^3　　エ　-2^1+2^4

次の問9から問13までの5問については，この中から1問を選択し，選択した問題については，答案用紙の選択欄の(選)をマークして解答してください。

なお，2問以上選択した場合には，はじめの1問について採点します。

問9 次のC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。

〔プログラムの説明〕

5台のバスが走行する路線上の10か所の停留所（以下，バス停という）のそれぞれに取り付けられた表示装置に，バスの運行時間中，次のバスが到着するまでの待ち時間（以下，到着待ち時間という）を分単位で表示するシステムがある。関数update_wait_timeは，このシステムにおいて，各バス停に表示する到着待ち時間を計算するプログラムである。表示する到着待ち時間は，関数update_wait_timeが実行されたときに更新される。

(1) 関数update_wait_timeの引数は，次のとおりである。ここで，引数の値に誤りはないものとする。

bus_id	バスの車体番号
busstop	バス停の番号
route	路線表（構造体BUSSTOPの配列）
bus	路線上を走行するバスの一覧（構造体BUSの配列）

(2) 構造体BUSSTOPの構造は，次のとおりである。

```
typedef struct {
    int  std_time;   /* 前のバス停からこのバス停までの
                        標準所要時間（分） */
    int  wait_time;  /* 到着待ち時間（分） */
} BUSSTOP;
```

(3) 構造体BUSの構造は，次のとおりである。

```
typedef struct {
    int  id;         /* バスの車体番号 */
    int  cur_pos;    /* バスの走行位置 */
} BUS;
```

(4) 路線表 route は，バス停の番号（以下，バス停番号という）0～9 を添字とする配列である。route には，図 1 のようにデータを格納する。

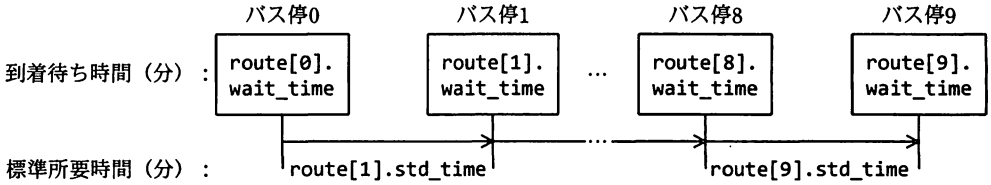


図 1　路線表 route のデータ

① route[i].std_time（i=1, 2, …, 9）は，前のバス停（i−1）からこのバス停 i までの標準所要時間である。バス停 0 における route[0].std_time の値は 0 とする。

② 各バス停 i の到着待ち時間 route[i].wait_time（i=1, 2, …, 9）は，次に到着するバスが最後に発車したバス停 a からこのバス停 i までの標準所要時間 route[k].std_time（k=a+1, …, i）の合計である。乗客の乗降にかかる時間は 0 とする。バス停 0 における route[0].wait_time の値は 0 とする。次に到着するバスがバス停 0 を発車していないとき，そのバス停 i における到着待ち時間 route[i].wait_time には 0 を格納する。

(5) バスの走行位置 bus[j].cur_pos（j=0, 1, …, 4）には，走行中の区間番号を格納する。あるバス停から次のバス停までの区間に対する区間番号には，区間の始まりとなるバス停番号を割り当てる。バス停 0 を発車していないバスの走行位置 bus[j].cur_pos には −1 を格納する。また，バス停 9 に到着したバスの走行位置 bus[j].cur_pos には 9 を格納する。

(6) プログラム中で使われる変数 preceding と succeeding の意味は次のとおりである。

preceding	busstopが示すバス停を直前に発車したバスが走行中の区間番号を格納する。busstopが示すバス停からバス停9までの区間に走行中のバスがない場合は9とする。
succeeding	busstopが示すバス停に次に到着するバスが走行中の区間番号を格納する。バス停0からbusstopが示すバス停までの区間に走行中のバスがない場合は−1とする。

バス停 5 をバス B が発車するとき，バス停 5 を直前に発車したバス A が区間 6 を走行し，次にバス停 5 に到着するバス C が区間 3 を走行しているときの例を，図 2 に示す。

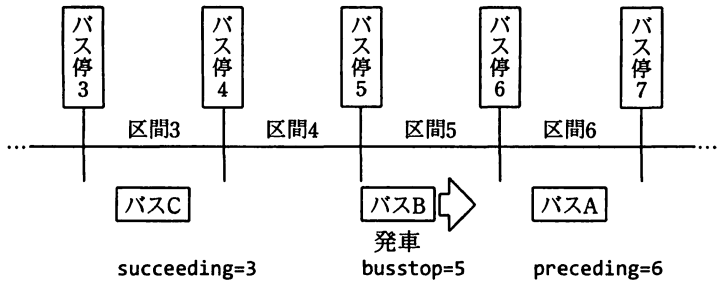


図２　変数 busstop，preceding 及び succeeding の例

(7)　関数 update_wait_time は，バスがバス停 0～8 を発車するとき又はバス停 9 に到着したときに呼び出され，bus_id で識別されるバスの走行位置を更新し，busstop が示すバス停から preceding が示す区間の始まりとなるバス停までの，各バス停に表示する到着待ち時間 route[i].wait_time (i=busstop, …, preceding) を計算する。

なお，複数台のバスがバス停 0～8 の異なるバス停を同時に発車する場合，又はあるバスがバス停 9 に到着したときに別のバスがバス停 0～8 を発車する場合には，関数 update_wait_time は，バスの一覧 bus の添字の昇順に 1 台分ずつ実行される。

(8)　一つの区間を複数台のバスが同時に走ることはない。

〔プログラム〕

(行番号)

```
 1  #define STPNUM 10        /* バス停の個数 */
 2  #define BUSNUM  5        /* バスの台数 */

 3  typedef struct {
 4      int  std_time;       /* 前のバス停からこのバス停までの
 5                              標準所要時間（分） */
 6      int  wait_time;      /* 到着待ち時間（分） */
 7  } BUSSTOP;

 8  typedef struct {
 9      int  id;             /* バスの車体番号 */
10      int  cur_pos;        /* バスの走行位置 */
11  } BUS;

12  void update_wait_time(int, int, BUSSTOP[STPNUM], BUS[BUSNUM]);

13  void update_wait_time(int bus_id, int busstop,
14                        BUSSTOP route[STPNUM], BUS bus[BUSNUM]) {
15      int preceding = STPNUM - 1, succeeding = -1, i, j;
```

```
16      /* バスの走行位置，succeeding と preceding の更新 */
17      for (j = 0; j < BUSNUM; j++) {
18          if ( [ a ] ) {
19              bus[j].cur_pos = busstop;
20          } else {
21              if (bus[j].cur_pos < busstop) {
22                  if (succeeding < bus[j].cur_pos) {
23                      succeeding = [ b ];
24                  }
25              } else {
26                  if (preceding > bus[j].cur_pos) {
27                      preceding = [ b ];
28                  }
29              }
30          }
31      }

32      /* このバス停の到着待ち時間の計算 */
33      if ( [ c ] ) {
34          route[busstop].wait_time = 0;
35      } else {
36          route[busstop].wait_time = route[busstop].std_time
37                                   + route[busstop - 1].wait_time;
38      }

39      /* 次のバス停から preceding が示すバス停までの到着待ち時間の計算 */
40      for (i = busstop + 1; i <= preceding; i++) {
41          route[i].wait_time = route[i].std_time;
42          if (i != busstop + 1) {
43              route[i].wait_time [ d ];
44          }
45      }
46  }
```

設問1　プログラム中の［　　］に入れる正しい答えを，解答群の中から選べ。

aに関する解答群

ア　bus_id == bus[j].id　　イ　bus_id != bus[j].id

ウ　bus_id == j　　エ　bus_id != j

bに関する解答群

ア　-1　　イ　busstop　　ウ　bus[j].cur_pos

エ　bus[j].cur_pos + 1　　オ　bus[j].cur_pos - 1　　カ　j

cに関する解答群

ア	`preceding == -1`	イ	`preceding == STPNUM - 1`
ウ	`preceding != -1`	エ	`succeeding == -1`
オ	`succeeding == 0`	カ	`succeeding != 0`

dに関する解答群

ア	`+= route[i - 1].std_time`	イ	`+= route[i - 1].wait_time`
ウ	`+= route[i].std_time`	エ	`= route[i - 1].std_time`
オ	`= route[i - 1].wait_time`	カ	`= route[i].std_time`

設問2　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。

サービス向上のため，路線上のそれぞれのバス停に取り付けられた表示装置に，到着待ち時間に加えて，そのバスの運行遅延時間を分単位で表示することにした。そのために，関数 update_wait_time に，各バス停に表示する運行遅延時間を計算する処理を追加する。ここで，プログラム中の［ a ］～［ d ］には，正しい答えが入っているものとする。

(1) バスの運行遅延時間は，バス停0から最後に発車したバス停までの標準所要時間の合計と実際にかかった時間の合計の差である。

(2) 関数 update_wait_time の引数に，前のバス停からの実際にかかった時間（分）act_time を追加する。ここで，引数の値に誤りはないものとする。

(3) 構造体 BUSSTOP の要素に，次に到着するバスの運行遅延時間（分）delay_time を追加する。

(4) 構造体 BUS の要素に，バスの運行遅延時間（分）cur_delay を追加する。

(5) バスが定刻よりも早くバス停0～8を発車することはない。

(6) 処理の追加に対応するために，プログラムを表のとおりに変更する。

表　プログラムの変更内容

処置	変更内容

行番号3～14を変更	typedef struct {
	　　int　std_time;　　/* 前のバス停からこのバス停までの
	　　標準所要時間（分）　*/
	　　int　wait_time;　　/* 到着待ち時間（分）　*/
	　　int　delay_time;　　/* 次に到着するバスの運行遅延時間（分）　*/
	} BUSSTOP;
	
	typedef struct {
	　　int　id;　　/* バスの車体番号 */
	　　int　cur_pos;　　/* バスの走行位置 */
	　　int　cur_delay;　　/* バスの運行遅延時間（分）　*/
	} BUS;
	
	void update_wait_time(int, int, int, BUSSTOP[STPNUM], BUS[BUSNUM]);
	
	void update_wait_time(int bus_id, int busstop, int act_time,
	　　BUSSTOP route[STPNUM], BUS bus[BUSNUM]) {

行番号15と16の間に追加	int delay_preceding, delay_succeeding = 0;
行番号19と20の間に追加	bus[j].cur_delay [e]; 　　delay_preceding [f];
行番号23と24の間に追加	delay_succeeding [f];
行番号38と39の間に追加	route[busstop].delay_time = delay_succeeding;
行番号44と45の間に追加	route[i].delay_time = delay_preceding;

eに関する解答群

ア　= act_time - route[busstop].std_time

イ　= bus[j - 1].cur_delay

ウ　= route[busstop].delay_time

エ　+= act_time - route[busstop].std_time

オ　+= bus[j - 1].cur_delay

カ　+= route[busstop].delay_time

fに関する解答群

ア = act_time

イ = bus[j].cur_delay

ウ = route[busstop].delay_time

エ += act_time

オ += bus[j].cur_delay

カ += route[busstop].delay_time

問 10 次の COBOL 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。

〔プログラムの説明〕

ある有料自動車道路には A～J の順に 10 か所のインターチェンジ（以下，IC という）があり，すべての IC に自動発券精算システムを導入することにした。すべての IC には一般道路からの出入口があり，利用者は，入った IC（以下，入車 IC という）で発券された通行券を，出た IC（以下，出車 IC という）の精算機に投入し，利用区間に応じた通行料を支払う。システムは計算センタとオンラインで結ばれており，通行料は出車 IC の売上として売上ファイルに記録される。

このプログラムは，売上ファイルを読み込み，各 IC の売上，及び利用区間とその利用台数を台数の多い順に 10 区間まで，図 1 に示す形式で表示する。


```
A:          187,200
B:          514,700
          ⋮
J:           95,400
TOTAL:    3,086,200

B-E:    1,125
E-B:    1,079
C-E:      996
        ⋮
```

図 1　プログラム実行時の表示例

(1) 売上ファイルは，図 2 に示すレコード様式の順ファイルである。

入車IC 2けた	出車IC 2けた	通行料 4けた

図 2　売上ファイルのレコード様式

① 車 1 台の 1 回の利用につき 1 レコードが生成され，一つのファイルには 1 日分の売上が記録されている。

② 入車 IC 及び出車 IC には，各 IC に対応する 01～10 の番号が格納される。A は 01，B は 02，…，J は 10 とする。

(2) 各 IC の 1 日の売上は 999,999 円以下とする。

(3) 1 日の利用台数は 99,999 台以下とする。

(4) 利用台数の多い順に 10 区間までを表示する際，10 区間目と台数が同じ利用区間

が複数ある場合は，それらの利用区間もすべて表示する。ただし，利用台数がゼロになった場合は，表示が10区間に満たなくても処理を終了する。

(5) Uターン及び同じICからの入出車はないものとする。

〔プログラム〕

(行番号)

```
 1  DATA DIVISION.
 2  FILE SECTION.
 3  SD  SORT-FILE.
 4  01  SORT-REC.
 5      02  SORT-IN              PIC 9(2).
 6      02  SORT-OUT             PIC 9(2).
 7      02  SORT-DATA            PIC 9(6).
 8  FD  SALES-FILE.
 9  01  SALES-REC.
10      02  SAL-IN               PIC 9(2).
11      02  SAL-OUT              PIC 9(2).
12      02  SAL-TOLL             PIC 9(4).
13  WORKING-STORAGE SECTION.
14  77  LOOP-FLAG                PIC X(1) VALUE SPACE.
15      88  INIT                 VALUE SPACE.
16      88  LOOP-END             VALUE "E".
17  77  CNT1                     PIC 9(2).
18  77  CNT2                     PIC 9(2).
19  01  .
20      02  IC-TOTAL             PIC 9(6) VALUE ZERO OCCURS 10.
21  77  TOTAL                    PIC 9(8) VALUE ZERO.
22  01  SECT-DATA.
23      02                        OCCURS 10.
24          03  SECT-NUM         PIC 9(5) VALUE ZERO OCCURS 10.
25  77  PREV-NUM                 PIC 9(6) VALUE ZERO.
26  01  PRT-IC.
27      02  IC-HEADER            PIC X(1).
28      02                        PIC X(9) VALUE ":".
29      02  IC-SALES             PIC ZZZ,ZZ9.
30  01  PRT-TOTAL.
31      02                        PIC X(7) VALUE "TOTAL:".
32      02  TOTAL-SALES          PIC ZZ,ZZZ,ZZ9.
33  01  PRT-SECT.
34      02  PRT-IN               PIC X(1).
35      02                        PIC X(1) VALUE "-".
36      02  PRT-OUT              PIC X(1).
37      02                        PIC X(2) VALUE ":".
38      02  PRT-NUM              PIC ZZ,ZZ9.
39  01  IC-NAME                  PIC X(10) VALUE "ABCDEFGHIJ".
40  PROCEDURE DIVISION.
41  MAIN-PROC SECTION.
42      OPEN INPUT SALES-FILE.
43      PERFORM UNTIL LOOP-END
```

```
44          READ SALES-FILE AT END     SET LOOP-END TO TRUE
45                          NOT AT END PERFORM CNT-PROC
46          END-READ
47       END-PERFORM.
48       CLOSE SALES-FILE.
49       PERFORM PRINT-PROC.
50       STOP RUN.
51   CNT-PROC SECTION.
52       ADD SAL-TOLL TO IC-TOTAL(SAL-OUT).
53       [          a          ].
54   PRINT-PROC SECTION.
55       PERFORM VARYING CNT1 FROM 1 BY 1 UNTIL CNT1 > 10
56          MOVE IC-NAME(CNT1:1) TO IC-HEADER
57          MOVE IC-TOTAL(CNT1) TO IC-SALES
58          DISPLAY PRT-IC
59          [          b          ]
60       END-PERFORM.
61       MOVE TOTAL TO TOTAL-SALES.
62       DISPLAY PRT-TOTAL.
63       DISPLAY " ".
64       SORT SORT-FILE ON DESCENDING KEY SORT-DATA
65                      INPUT  PROCEDURE IS IN-PROC
66                      OUTPUT PROCEDURE IS OUT-PROC.
67   IN-PROC  SECTION.
68       PERFORM VARYING CNT1 FROM 1 BY 1 UNTIL CNT1 > 10
69          PERFORM VARYING CNT2 FROM 1 BY 1 UNTIL CNT2 > 10
70             MOVE CNT1 TO SORT-IN
71             MOVE CNT2 TO SORT-OUT
72             [          c          ]
73             RELEASE SORT-REC
74          END-PERFORM
75       END-PERFORM.
76   OUT-PROC SECTION.
77       SET INIT TO TRUE.
78       PERFORM VARYING CNT1 FROM 1 BY 1 UNTIL LOOP-END
79          RETURN SORT-FILE
80             AT END
81                SET LOOP-END TO TRUE
82             NOT AT END
83                IF SORT-DATA = ZERO OR
84                   ([          d          ]) THEN
85                   SET LOOP-END TO TRUE
86                ELSE
87                   MOVE IC-NAME(SORT-IN:1) TO PRT-IN
88                   MOVE IC-NAME(SORT-OUT:1) TO PRT-OUT
89                   MOVE SORT-DATA TO PRT-NUM
90                   DISPLAY PRT-SECT
91                   MOVE SORT-DATA TO PREV-NUM
92                END-IF
93          END-RETURN
94       END-PERFORM.
```

設問1　プログラム中の［　　］に入れる正しい答えを，解答群の中から選べ。

a～cに関する解答群

ア　ADD SECT-NUM(SAL-IN SAL-OUT) TO TOTAL

イ　ADD 1 TO IC-TOTAL(CNT1)

ウ　ADD 1 TO SECT-NUM(SAL-IN SAL-OUT)

エ　ADD IC-TOTAL(CNT1) TO SECT-NUM(SAL-IN SAL-OUT)

オ　ADD IC-TOTAL(CNT1) TO TOTAL

カ　MOVE IC-TOTAL(CNT1) TO SORT-DATA

キ　MOVE SECT-NUM(CNT1 CNT2) TO SORT-DATA

dに関する解答群

ア　CNT1 > 10 AND PREV-NUM <= 10

イ　CNT1 > 10 AND SORT-DATA < PREV-NUM

ウ　CNT1 > 10 AND SORT-DATA <= PREV-NUM

エ　CNT1 > 10 AND SORT-DATA >= PREV-NUM

オ　SORT-IN > 10 AND SORT-OUT > 10

設問2　この道路の運営会社では，利用区間に応じた料金の割引きを検討している。売上への影響を検証した上で，次の二つのプランのいずれかを採用することにした。なるべく新しいデータで検証するため，ある期間だけプログラムを変更して，それぞれのプランを適用した場合の売上を表示することにした。図3に変更後の表示形式を示す。次の表中の［　　］に入れる正しい答えを，解答群の中から選べ。

プランX：5区間以上利用した場合は20％，通行料を割り引く。

プランY：3～6区間利用した場合は10％，7区間以上利用した場合は20％，通行料を割り引く。

```
A:         187,200
B:         514,700
          ⋮
J:          95,400
TOTAL:   3,086,200   PLAN X:  2,899,400   PLAN Y:  2,814,990

B-E:  1,125
E-B:  1,079
C-E:    996
      ⋮
```

図３　割引プラン適用時の売上を追加表示する場合の表示例

表　プログラムの変更内容

処置	変更内容
行番号25と26の間に追加	77 TOTAL-PLANX PIC 9(8) VALUE ZERO. 77 TOTAL-PLANY PIC 9(8) VALUE ZERO. 77 DISCOUNT-X PIC 9V9. 77 DISCOUNT-Y PIC 9V9.
行番号32と33の間に追加	02 PIC X(11) VALUE " PLAN X:". 02 TOTAL-X PIC ZZ,ZZZ,ZZ9. 02 PIC X(11) VALUE " PLAN Y:". 02 TOTAL-Y PIC ZZ,ZZZ,ZZ9.

[e]の間に追加	IF SAL-IN < SAL-OUT THEN
	COMPUTE CNT1 = SAL-OUT - SAL-IN
	ELSE
	COMPUTE CNT1 = SAL-IN - SAL-OUT
	END-IF.
	EVALUATE CNT1
	WHEN [f] MOVE 1.0 TO DISCOUNT-X
	MOVE 0.9 TO DISCOUNT-Y
	WHEN [g] MOVE 0.8 TO DISCOUNT-X
	MOVE 0.9 TO DISCOUNT-Y
	WHEN [h] MOVE 0.8 TO DISCOUNT-X DISCOUNT-Y
	WHEN OTHER MOVE 1.0 TO DISCOUNT-X DISCOUNT-Y
	END-EVALUATE.
	COMPUTE TOTAL-PLANX = TOTAL-PLANX + SAL-TOLL * DISCOUNT-X.
	COMPUTE TOTAL-PLANY = TOTAL-PLANY + SAL-TOLL * DISCOUNT-Y.

行番号61と62の間に追加	MOVE TOTAL-PLANX TO TOTAL-X. MOVE TOTAL-PLANY TO TOTAL-Y.

eに関する解答群

ア　行番号 47 と 48　　イ　行番号 53 と 54

ウ　行番号 54 と 55　　エ　行番号 60 と 61

f～hに関する解答群

ア　1 THRU 2	イ　3	ウ　3 THRU 4
エ　3 THRU 5	オ　5	カ　5 THRU 6
キ　5 THRU 7	ク　7	ケ　7 THRU 9

問11　次のJava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。

（Javaプログラムで使用するAPIの説明は，この冊子の末尾を参照してください。）

〔プログラムの説明〕

電子会議システムのプログラムである。

電子会議は，サーバに接続しているクライアント間で行われ，一つのクライアントは，会議の参加者1人に対応する。会議に参加するときは，電子会議システムのサーバにログインする。参加者の発言はクライアントからのメッセージとしてサーバに送られる。サーバは受信したメッセージを，発信元クライアントを含む全クライアントに配信する。会議から退出するときは，ログアウトする。

この電子会議システムのサーバを実装するために，次のクラスを定義する。

(1) クラスMessageQueueは，クライアントからのメッセージを格納するための待ち行列（メッセージキュー）である。メッセージは，先入れ先出しで管理される。メソッドputは，引数messageで与えられたメッセージをメッセージキューに追加する。メッセージキューに格納できるメッセージ数には制限があり，満杯のときは，空きができるまで待つ。メソッドtakeは，メッセージキューの先頭からメッセージを取り出す。メッセージキューが空のときは，メッセージが追加されるまで待つ。

(2) クラスConfServerは，サーバを定義する。クライアントとのログイン状態を管理し，メッセージの受信と配信を行う。ここでクライアントとは，(3)で説明する抽象クラスの型のインスタンスである。クラスConfServerは，クラスが初期化されるときにインスタンスが作成され，単独のスレッドとして動作する。メソッドrunは，メッセージキューからメッセージを取り出し，ログインしている全クライアントにそのメッセージを配信する。この操作を繰り返す。メソッドloginは，入れ子クラスConfServer.Sessionのインスタンスを生成し，それをキーとしてクライアントを管理テーブルに登録し，そのインスタンスを返す。クライアントが既に登録されている場合は，IllegalArgumentExceptionを投げる。クライアントが管理テーブルに登録されているとき，そのクライアントはサーバにログインしている状態であるとする。

(3) 抽象クラスConfClientは，サーバが必要とするクライアントの機能を定義する。サーバは，メソッドdisplayMessageを呼び出してクライアントにメッセージを配

信する。クライアントは，このクラスを実装しなければならない。クライアントがサーバにメッセージを送信するときは，ログイン時に返されたクラス ConfServer.Session のインスタンスのメソッド writeMessage を呼び出す。クライアントがログアウトするときは，同じインスタンスのメソッド logout を呼び出す。

(4) サーバをテストするために，クラス TestClient を定義する。このクラスは，ConfClient で定義されたメソッドをテスト用に実装する。TestClient のインスタンスは，サーバに対してクライアントの役割をする。メソッド displayMessage は，メッセージを次の形式で出力する。

発信クライアント名: メッセージ　>受信クライアント名

〔プログラム 1〕

```
import java.util.LinkedList;

public class MessageQueue {
   // 格納できる最大のメッセージ数
   private final static int MAX_SIZE = 3;
   private final LinkedList<String> queue = new LinkedList<String>();

   public synchronized void put(String message) {
      while (queue.size() >= MAX_SIZE) {
         try {
            wait();
         } catch (InterruptedException e) { }
      }
      queue.add(message);
      notifyAll();
   }

   public synchronized String take() {
      while ( [ a ] ) {
         try {
            wait();
         } catch (InterruptedException e) { }
      }
      String message = queue.removeFirst();
      notifyAll();
      return message;
   }
}
```

〔プログラム 2〕

```
import java.util.HashMap;
import java.util.Map;

public class ConfServer implements Runnable {
   // ConfServer のインスタンス
   private static final ConfServer server;
   static {
      server = new ConfServer();
      new Thread(server).start();
   }

   // クライアントからのメッセージを格納するメッセージキュー
   private final MessageQueue queue = new MessageQueue();

   // セッション管理テーブル
   private final Map<Session, ConfClient> sessionsTable
         = new HashMap<Session, ConfClient>();

   public static Session login(ConfClient client) {
      if (client == null)
         throw new NullPointerException();
      return server.loginImpl(client);
   }

   private ConfServer() { }

   public void run() {
      while (true) {
         String message = queue.take();
         deliverMessage(message);
      }
   }

   private void writeMessage(Session session,
                             String message) {
      ConfClient client = getClient(session);
      String s = client.getName() + ": " + message;
      queue.put(s);
   }

   private synchronized void deliverMessage(String message) {
      for (Session session : sessionsTable.keySet())
         [ b ].displayMessage(message);
   }

   private synchronized ConfClient getClient(Session session) {
      ConfClient client = sessionsTable.get(session);
      if (client == null)
         throw new IllegalStateException("無効なセッション");
      return client;
   }
```

```
    private synchronized Session loginImpl(ConfClient client) {
        if (sessionsTable.containsValue(client))
            throw new IllegalArgumentException(
                    client.getName() + "はログイン済み");
        Session session = new Session();
        sessionsTable.put(session, client);
        return session;
    }

    private synchronized void logoutImpl(Session session) {
        sessionsTable.remove(session);
    }

    public static class Session {
        private Session() { }

        public void writeMessage(String msg) {
            server.writeMessage(this, msg);
        }

        public void logout() {
            server.logoutImpl( c );
        }
    }
}
```

〔プログラム3〕

```
public abstract class ConfClient {
    private final String name;

    public ConfClient(String name) {
        if (name == null)
            throw new NullPointerException();
        this.name = name;
    }

    public final String getName() { return name; }

    public abstract void displayMessage(String message);

    public boolean equals(Object obj) {
        if (!(obj instanceof ConfClient))
            return false;
        return name.equals(((ConfClient)obj).name);
    }

    public int hashCode() {
        return name.hashCode();
    }
}
```

〔プログラム4〕

```
public class TestClient [ d ] ConfClient {
   TestClient(String name) { [ e ]; }

   public void displayMessage(String message) {
      System.out.println(message + " >" + getName());
   }

   public static void main(String[] arg) {
      ConfServer.Session yamada
              = ConfServer.login(new TestClient("山田"));
      ConfServer.Session sato
              = ConfServer.login(new TestClient("佐藤"));

      yamada.writeMessage("こんにちは。山田です。");
      sato.writeMessage("こんにちは。");
      yamada.writeMessage("開発プランの資料は届いていますか。");
      sato.writeMessage("はい、手元にあります。");
      yamada.writeMessage("では、資料に沿ってご説明します。");

      // 全メッセージを配信し終わるように，1秒間待つ。
      try {
         Thread.sleep(1000);
      } catch (InterruptedException e) { }
      yamada.logout();
      sato.logout();
   }
}
```

Java

設問1　プログラム中の［　　］に入れる正しい答えを，解答群の中から選べ。

aに関する解答群

ア　!queue.isEmpty()　　イ　queue.isEmpty()

ウ　queue.size() != MAX_SIZE　　エ　queue.size() < MAX_SIZE

オ　queue.size() == MAX_SIZE

bに関する解答群

ア　session.get(message)　　イ　session.get(server)

ウ　session.get(sessionsTable)　　エ　sessionsTable.get(message)

オ　sessionsTable.get(server)　　カ　sessionsTable.get(session)

cに関する解答群

ア　queue　　イ　server　　ウ　session

エ　sessionsTable　　オ　this

dに関する解答群

ア　abstract　　イ　extends　　ウ　implements

エ　static　　オ　throws

eに関する解答群

ア　super()　　イ　super(name)　　ウ　super(this)

エ　this()　　オ　this(name)

設問2　クラスMessageQueueのメソッドput及びtakeには，synchronized修飾子が付けられている。これには理由が二つある。一つは，キューの状態（空又は満杯）によってスレッド間でメソッドwait及びnotifyAllを呼び出して同期を取るためである。もう一つの理由として適切な答えを，解答群の中から選べ。

解答群

ア　クラスConfClientのインスタンスがそれぞれ別スレッドとして同時に動くことを想定すると，一度にフィールドqueueで参照されるLinkedListのインスタンスにアクセスが集中することが考えられる。そこで，一時点でqueueにアクセスできるスレッドを一つにして，システムに負荷がかかりすぎるのを防ぐ。

イ　クラスConfClientのインスタンスがそれぞれ別スレッドとして同時に動くことを想定すると，タイミングによってはフィールドqueueで参照されるLinkedListのインスタンスに格納されている他のスレッドが書き込んだメッセージを読み出すことができ，セキュリティ上問題となる。そこで，一時点でqueueにアクセスできるスレッドを一つにして，queueに既に書き込まれているメッセージを読み出せないようにする。

ウ　クラスConfClientのインスタンスとクラスConfServerのインスタンスは別スレッドとして動くので，フィールドqueueで参照されるLinkedListのインスタンスに対しての操作が同時に実行されることがある。そこで，一時点でqueueにアクセスできるスレッドを一つにして，データの内部状態に矛盾が起きないようにする。

エ　クラスConfClientのインスタンスの個数が多いとフィールドqueueで参照されるLinkedListのインスタンスに格納するメッセージ数の増大によってメモリ不足によるエラーが発生し，ConfServerを実行しているスレッドが停止する可能性がある。そこで，ConfClientのインスタンスの個数を制限して，エラーを起こさないようにする。

問 12 次のアセンブラプログラムの説明及びプログラムを読んで，設問 1～3 に答えよ。

〔プログラムの説明〕

16 ビット（1 語）からなるビット列のビットの並びを，図 1 に示すように逆転する副プログラム REVRS である。

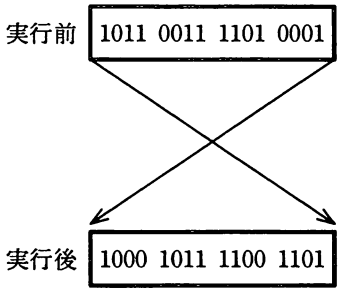


図 1　ビットの並びを逆転する例

(1) ビット列を格納している語のアドレスは GR1 に設定されて，主プログラムから渡される。

(2) 結果は元の領域に格納して，主プログラムに返す。

(3) 副プログラムから戻るとき，汎用レジスタ GR1～GR7 の内容は元に戻す。

〔プログラム 1〕

```
REVRS   START
        RPUSH
        LD      GR4,=0          ; 結果のビット列を初期化
        LAD     GR2,15          ; ループカウンタ
        LD      GR3,0,GR1       ; GR3 ← ビット列
LOOP    SLL     GR4,1           ; 結果のビット列を左シフト
        SRL     GR3,1           ; 元のビット列を右シフト
        [      a      ]
        JZE     FIN1            ; 元のビット列の残りのビットはすべてゼロ
        JUMP    OFF
ON      OR      GR4,=#0001
OFF     SUBA    GR2,=1
        JMI     FIN2            ; 16 ビット処理済み
        JUMP    LOOP
FIN1    [      b      ]         ; 結果のビット列を残りのビット数だけシフト
FIN2    ST      GR4,0,GR1
        RPOP
        RET
        END
```

アセンブラ

設問1　プログラム中の［　　］に入れる正しい答えを，解答群の中から選べ。

aに関する解答群

ア　JMI　ON　　イ　JNZ　ON　　ウ　JOV　ON　　エ　JPL　ON

bに関する解答群

ア　SLL　GR4,-1,GR2　　イ　SLL　GR4,0,GR2

ウ　SRL　GR4,-1,GR2　　エ　SRL　GR4,0,GR2

設問2　連続した n 語（n≧1）を 16×n ビットのビット列とみなす。副プログラム REVRS を利用して，このビット列のビットの並びを，図2に示すように逆転する副プログラム LREVRS を作成した。

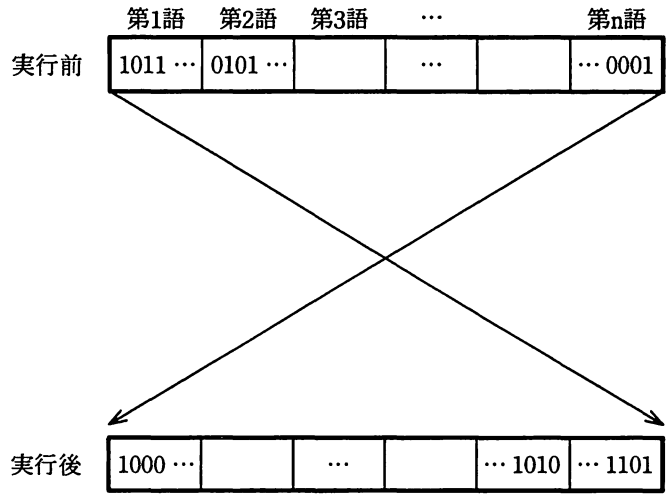


図2　n語のビット列のビットの並びを逆転

主プログラムから渡されるレジスタの内容は，次のとおりとする。

GR1：　ビット列が格納されている領域の先頭アドレス

GR2：　n

副プログラム LREVRS 中の［　　］に入れる正しい答えを，解答群の中から選べ。

〔プログラム 2〕

```
LREVRS  START
        RPUSH
        LD      GR3,GR1
        LD      GR4,GR2
        SUBA    GR4,=1
        JZE     LOOP2
        ADDA    GR4,GR1
LOOP1   LD      GR5,0,GR3       ;
        LD      GR6,0,GR4       ;   GR3 が指す語と GR4 が指す語の内容を
        ST      GR5,0,GR4       ;   入れ替える
        ST      GR6,0,GR3       ;
        LAD     GR3,1,GR3       ; GR3 を次の語に位置付ける
        LAD     GR4,-1,GR4      ; GR4 を一つ前の語に位置付ける
        CPA     GR3,GR4
        [      c      ]
LOOP2   CALL    REVRS
        LAD     GR1,1,GR1
        SUBA    GR2,=1
        [      d      ]
FIN3    RPOP
        RET
        END
```

解答群

ア	JMI	LOOP1	イ	JNZ	LOOP1	ウ	JNZ	LOOP2
エ	JPL	LOOP1	オ	JZE	FIN3	カ	JZE	LOOP2

設問 3　副プログラム REVRS を使用して，16 ビット（1 語）からなるビット列中の部分ビット列 α のビットの並びを，図 3 に示すように逆転する副プログラム PREVRS を作成した。p，q は，それぞれ部分ビット列の直前までのビット数及び部分ビット列のビット数を表す。

ここで，$p \geqq 0$，$q > 1$，$p + q \leqq 16$ とする。

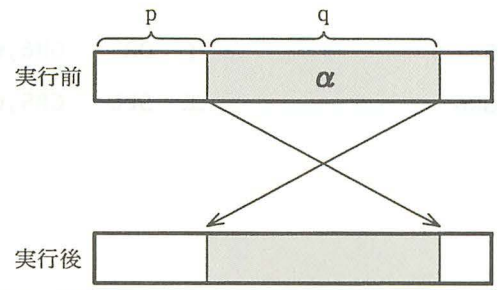


図 3　部分ビット列 α のビットの並びを逆転

主プログラムから渡されるレジスタの内容は，次のとおりとする。

GR1： ビット列が格納されている語のアドレス

GR2： p

GR3： q

副プログラム PREVRS 中の［　　］に入れる正しい答えを，解答群の中から選べ。

〔プログラム3〕

```
PREVRS  START
        RPUSH
        LD      GR4,0,GR1       ; ビット列を保存
        CALL    REVRS           ; ビット列のビットの並びを逆転
        LD      GR5,0,GR1       ; GR5 ← 逆転したビット列
        LD      GR6,=16
        SUBA    GR6,GR3         ; GR6 ← 16-q
        SRL     GR5,0,GR2       ; 逆転した部分ビット列αを右端に移動
        [   e   ]               ; 逆転した部分ビット列αを左端に移動
        SRL     GR5,0,GR2       ; 逆転した部分ビット列αをpビット右に移動
        LD      GR6,=#8000
        SRA     GR6,-1,GR3      ; qビット連続した1の並びを作成
        SRL     GR6,0,GR2
        XOR     GR6,=#FFFF
        AND     GR6,GR4         ; 元のビット列中の部分ビット列αにゼロを設定
        [   f   ]
        ST      GR6,0,GR1
        RPOP
        RET
        END
```

解答群

ア　OR　　GR6,GR5　　イ　OR　　GR6,0,GR1

ウ　SLL　　GR5,0,GR6　　エ　SLL　　GR5,0,GR3

問 13　次の表計算及びワークシートの説明を読んで，設問 1～3 に答えよ。

〔表計算の説明〕

日用品メーカ N 社は，普及品及び高級品の二つのブランドのシャンプーを製造販売している。原料価格の高騰によって，値上げを検討することになった。値上げによる売上数量の減少が懸念されるので，消費者調査を行い，ブランドごとの購入意向率の価格弾力性（以下，価格弾力性という）の分析及び価格設定に関する検討を，表計算ソフトを利用して行うことにした。

〔購入意向率及び価格弾力性の説明〕

(1)　購入意向率とは，ある商品の価格について，“その価格で購入する意思があるか”という質問に対して，調査対象者の中で“ある”と回答した者の比率である。

(2)　同じ商品であってもその価格を変化させることで，購入意向率は変化する。日用品の場合，価格を下げることで購入意向率は高くなることが多い。価格弾力性とは，価格を変化させた割合（価格の変化率）と，そのときの購入意向率が変化した割合（購入意向率の変化率）との関係を示す指標であり，次の計算式によって表される。

価格弾力性　＝　－（購入意向率の変化率／価格の変化率）

なお，ある商品の価格を P1 から P2 に変化させたとき，(P2 － P1)／P1 が価格の変化率である。価格が P1 のときの購入意向率を Q1，価格が P2 のときの購入意向率を Q2 とした場合，(Q2 － Q1)／Q1 が購入意向率の変化率である。

〔ワークシート：価格弾力性分析〕

ブランドごと価格ごとの価格弾力性を計算するためのワークシート“価格弾力性分析”を作成した。そのワークシートを図 1 に示す。

	A	B	C	D	E
1	ブランド	普及品	高級品		
2	現行価格（円）	600	800		
3	購入意向率	25%	30%		
4					
5		購入意向率		価格弾力性	
6	価格（円）	普及品	高級品	普及品	高級品
7	400	35%	28%	120%	−13%
8	500	30%	33%	120%	27%
9	600	25%	35%	—	67%
10	700	20%	33%	120%	80%
11	800	14%	30%	132%	—
12	900	9%	29%	128%	27%
13	1,000	4%	25%	126%	67%
14	1,100	2%	20%	110%	89%
15	1,200	0%	18%	100%	80%

注　購入意向率及び価格弾力性は，小数第3位を四捨五入した値を百分率で表示している。

図1　ワークシート“価格弾力性分析”

設問1　ワークシート“価格弾力性分析”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。

(1)　普及品，高級品の現行価格は，それぞれ600円，800円であり，それらをセルB2，C2に入力する。

(2)　普及品，高級品の現行価格に対する購入意向率は，調査の結果，それぞれ0.25，0.30であり，それらの値をセルB3，C3に入力する。

(3)　普及品と高級品について，価格を400～1,200円の間で100円刻みに変化させたときの購入意向率の調査結果の値を，セルB7～C15に入力する。

(4)　価格弾力性の計算結果をセルD7～E15に表示する。ただし，現行価格と同じ価格の価格弾力性については，分母が0となるので“−”を表示する。

ここで，セルD7に価格弾力性を算出するための次の計算式を入力して，セルD7～E15に複写した。

IF(［ a ］,'−',［ b ］／［ c ］)

表計算

aに関する解答群

ア　A7＝B2	イ　A$7＝B$2	ウ　A$7＝$B2	エ　$A7＝B$2
オ　$A7＝$B2	カ　B7＝B3	キ　B$7＝B$3	ク　B$7＝$B3
ケ　$B7＝B$3	コ　$B7＝$B3		

bに関する解答群

ア　((A7－B$2)／B$2)	イ　((B7－B$3)／B$3)
ウ　(($A7－B$2)／B$2)	エ　(($B7－B$3)／B$3)
オ　－((A7－B$2)／B$2)	カ　－((B7－B$3)／B$3)
キ　－(($A7－B$2)／B$2)	ク　－(($B7－B$3)／B$3)

cに関する解答群

ア　((B7－B$3)／B$3)	イ　((A$7－B$2)／B$2)
ウ　(($A7－B$2)／B$2)	エ　(($A7－$B2)／$B2)
オ　(($B7－B$3)／B$3)	カ　(($B7－$B3)／$B3)

〔ワークシート：値上〕

N社は，ブランドごとに，現行価格に対して容量と価格を変化させた案1～4について，利益を比較するためのワークシート“値上”を作成した。そのワークシートを図2に示す。

案1　現行容量，現行価格

案2　100ml増量，200円値上げ

案3　現行容量，100円値上げ

案4　100ml減量，現行価格

	A	B	C	D	E	F	G	H	I
1	ブランド	普及品	高級品		案	容量差（ml）	価格差（円）		列位置
2	容量（ml）	600	500		案1	0	0		2
3	価格（円）	600	800		案2	100	200		3
4	市場規模（百万本）	480	90		案3	0	100		
5	製造固定費（百万円）	15,000	7,000		案4	−100	0		
6	原料単価（円／ml）	0.350	0.600						
7	容器単価（円／本）	40	50						
8	販売管理費比率	10%	20%						
9	ブランド	普及品	普及品	普及品	普及品	高級品	高級品	高級品	高級品
10	案	案1	案2	案3	案4	案1	案2	案3	案4
11	容量（ml）	600	700	600	500	500	600	500	400
12	価格（円）	600	800	700	600	800	1,000	900	800
13	購入意向率	25%	15%	20%	24%	30%	33%	29%	27%
14	売上数量（百万本）	120	72	96	115.2	27	29.7	26.1	24.3
15	売上金額（百万円）	72,000	57,600	67,200	69,120	21,600	29,700	23,490	19,440
16	製造変動費（百万円）	30,000	20,520	24,000	24,768	9,450	12,177	9,135	7,047
17	利益（百万円）	19,800	16,320	21,480	22,440	830	4,583	2,657	1,505

注　販売管理費比率及び購入意向率は，小数第3位を四捨五入した値を百分率で表示している。

図2　ワークシート“値上”

設問2　ワークシート“値上”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。

なお，記述中の網掛け部分は表示していない。

図2中のワークシート“値上”の作成手順に関する説明は，次のとおりである。

(1)　セル B2～C8 に，ブランドごとの現行の容量（ml），価格（円），市場規模（百万本），製造固定費（百万円），原料単価（円／ml），容器単価（円／本），販売管理費比率を入力する。

(2)　セル F2～F5 に現行容量に対する案ごとの容量差（ml）を，セル G2～G5 に現行価格に対する案ごとの価格差（円）を入力する。

(3)　セル B11～I12 に，ブランドごと案ごとの容量（ml），価格（円）を求めるための計算式を入力する。

(4)　セル B13～I13 に，ブランドごと案ごとの購入意向率を入力する。

(5) セル B14～I17 に，ブランドごと案ごとに見込まれる売上数量（百万本），売上金額（百万円），製造変動費（百万円）及び利益（百万円）を表示するための計算式を入力する。この分析で使う各項目の計算式を次に示す。ここで，容量が変わっても容器単価は変わらないものとする。

〔計算式〕

売上数量　＝　市場規模×購入意向率
売上金額　＝　価格×売上数量
製造変動費　＝　（原料単価×容量＋容器単価）×売上数量
販売管理費　＝　売上金額×販売管理費比率
利益　＝　売上金額－製造変動費－製造固定費－販売管理費

(6) ワークシート“値上”で用いる関数を表 1 に示す。

表 1　ワークシート“値上”で用いる関数

書式	説明
垂直照合(照合値,照合範囲,列位置)	照合範囲が指す範囲の最左端列を上から下に走査し，照合値と等しい値をもつセルが現れる最初の行を探す。次に，その行内で照合範囲の最左端列から数えた列位置を 1，2，3，…とし，指定された列位置のセルの値を関数値として返す。

ブランドごと案ごとの容量，価格を求めるために，ワークシート“値上”のセル B11 には次の計算式を入力し，セル B11～I12 に複写する。あらかじめセル I2，I3 には，関数垂直照合の列位置用の数値を入力しておく。

IF(B$9＝$B$1,$B2＋垂直照合(［　d　］),$C2＋垂直照合(［　d　］))

セル B14 には，次の計算式を入力し，セル C14～I14 に複写する。

IF(B9＝$B1,［　e　］,［　　］)

セル B15 に売上金額を算出する計算式を入力し，セル C15～I15 に複写する。

セル B16 には，次の計算式を入力し，セル C16～I16 に複写する。

IF(B9＝$B1,［　　］,［　f　］)

セル B17 には，次の計算式を入力し，セル C17～I17 に複写する。

IF(B9＝$B1,［　g　］,［　　］)

表計算

dに関する解答群

ア　B10,E2～G5,I$2　　イ　B10,$E$2～$G$5,$I2

ウ　B$10,$E$2～$G$5,I$2　　エ　B$10,$E$2～$G$5,$I2

オ　$B10,$E$2～$G$5,I$2　　カ　$B10,$E$2～$G$5,$I2

eに関する解答群

ア　B4＊B$13　　イ　$B4＊B13　　ウ　$B4＊$B13

エ　$C4＊B13　　オ　$C4＊B$13　　カ　$C4＊$B13

fに関する解答群

ア　(B6＊B11＋$B7)＊B14　　イ　($B6＊B11＋$B7)＊B14

ウ　($B6＊B11＋$B7)＊$B14　　エ　(C6＊B11＋$C7)＊B14

オ　($C6＊B11＋$C7)＊B14　　カ　($C6＊B11＋$C7)＊$B14

gに関する解答群

ア　B15－B16－B5－B15＊B8　　イ　B15－B16－B$5－B15＊B$8

ウ　B15－B16－$B5－B15＊$B8　　エ　B15－B16－C5－B15＊C8

オ　B15－B16－C$5－B15＊C$8　　カ　B15－B16－$C5－B15＊$C8

〔ワークシート：広告〕

更なる利益増をねらって，広告の実施を検討することにした。広告費に応じた広告の効果を表す広告効果率は，ブランド，価格，容量の違いによらず，表 2 のとおりとする。ここで，広告効果率とは，広告を実施したことによる購入意向率の増加率である。例えば，広告実施前の購入意向率を 30％とし，500 百万円の広告費を投入した場合，表 2 を引くと広告効果率が 10％であることが分かり，広告実施後の購入意向率は，33％になる。

表 2　広告費ごとの広告効果率

広告費（百万円）	広告効果率（％）
150	3
300	5
500	10
1,000	13
1,500	16
2,000	20

一つのブランド，一つの案に対する広告費ごとの利益を算出するために，ワークシート“広告”を作成した。そのワークシートを，図 3 に示す。図 3 では，ブランドとして高級品，案として案 2 を選択した場合の例を示している。

広告費を投じた場合の利益の計算式を，次に示す。

利益＝売上金額－製造変動費－製造固定費－販売管理費－広告費

	A	B	C	D	E	F	G
1	ブランド	高級品					
2	案	案 2					
3	広告費（百万円）	150	300	500	1,000	1,500	2,000
4	広告効果率	3%	5%	10%	13%	16%	20%
5	広告実施後の購入意向率	34%	35%	36%	37%	38%	40%
6	売上数量（百万本）	30.591	31.185	32.670	33.561	34.452	35.640
7	売上金額（百万円）	30,591	31,185	32,670	33,561	34,452	35,640
8	製造変動費（百万円）	12,542	12,786	13,395	13,760	14,125	14,612
9	利益（百万円）	4,780	4,862	5,241	5,089	4,936	4,900

注　広告効果率及び広告実施後の購入意向率は，小数第 3 位を四捨五入した値を百分率で表示している。
売上数量は，小数第 4 位を四捨五入した結果を，売上金額，製造変動費及び利益は，小数第 1 位を四捨五入した結果を表示している。

図 3　ワークシート“広告”

ワークシート“広告”で用いる関数を表3に示す。

表3　ワークシート“広告”で用いる関数

書式	説明
水平照合(照合値, 照合範囲,行位置)	照合範囲が指す範囲の最上端行を左から右に走査し，照合値と等しい値をもつセルが現れる最初の列を探す。その列内で照合範囲の最上端行から数えた行位置を 1，2，3，… とし，指定された行位置のセルの値を関数値として返す。

設問3　ワークシート“広告”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。

なお，記述中の網掛けの部分は表示していない。

図3中の各項目の説明は，次のとおりである。ここで，複数のワークシート間でデータを参照するには“ワークシート名！セル”又は，“ワークシート名！範囲”という形式で指定する。

(1)　セルB1に選択したブランドを，セルB2に選択した案を入力する。

(2)　セルB3～G3に広告費（百万円），セルB4～G4に広告効果率を入力する。

(3)　セルB5に広告実施後の購入意向率を求めるために，ワークシート“値上”を参照した次の計算式を入力し，セルC5～G5に複写する。

［　h　］

(4)　セルB6～G6に売上数量（百万本）を，セルB7～G7に売上金額（百万円）を求める計算式を入力する。

(5)　セルB8に製造変動費（百万円）を求めるために，ワークシート“値上”を参照した次の計算式を入力し，セルC8～G8に複写する。

IF(\$B1＝値上!\$B1,［　i　］,［　　］)

(6)　セルB9～G9に，利益（百万円）を求める計算式を入力する。

hに関する解答群

ア　B4＋IF(B1＝値上!$B1,水平照合($B2,値上!$B10～$E13,4),
　　水平照合($D1,値上!$F10～$I13,4))

イ　B4＋IF($B1＝値上!$B1,水平照合(B$2,値上!$B10～$E13,3),
　　水平照合(B$2,値上!$F10～$I13,3))

ウ　B4＋IF($B1＝値上!$B1,水平照合($B2,値上!$B10～$E13,4),
　　水平照合($B2,値上!$F10～$I13,4))

エ　B4＊IF(B1＝値上!$B1,水平照合($B2,値上!$B10～$E13,3),
　　水平照合($B2,値上!$F10～$I13,3))

オ　B4＊IF($B1＝値上!$B1,水平照合(B2,値上!$B10～$E13,4),
　　水平照合(B2,値上!$F10～$I13,3))

カ　B4＊IF($B1＝値上!$B1,水平照合($B2,値上!$B10～$E13,3),
　　水平照合($B2,値上!$F10～$I13,3))

キ　(1＋B4)＊IF($B1＝値上!$B1,水平照合(B2,値上!$B10～$E13,3),
　　水平照合(B2,値上!$F10～$I13,3))

ク　(1＋B4)＊IF($B1＝値上!$B1,水平照合($B2,値上!$B10～$E13,3),
　　水平照合($B2,値上!$F10～$I13,3))

ケ　(1＋B4)＊IF($B1＝値上!$B1,水平照合($B2,値上!$B10～$E13,4),
　　水平照合($B2,値上!$F10～$I13,4))

iに関する解答群

ア　(値上!$B6＊水平照合(B2,値上!$B10～$E11,1)＋値上!$B7)＊B6

イ　(値上!$B6＊水平照合(B2,値上!$B10～$E11,2)＋値上!$B7)＊B6

ウ　(値上!$B6＊水平照合($B2,値上!$B10～$E11,1)＋値上!$B7)＊B6

エ　(値上!$B6＊水平照合($B2,値上!$B10～$E11,2)＋値上!$B7)＊B6

オ　(値上!$C6＊水平照合(B2,値上!$B10～$E11,1)＋値上!$C7)＊B6

カ　(値上!$C6＊水平照合(B2,値上!$B10～$E11,2)＋値上!$C7)＊B6

キ　(値上!$C6＊水平照合($B2,値上!$B10～$E11,1)＋値上!$C7)＊B6

ク　(値上!$C6＊水平照合($B2,値上!$B10～$E11,2)＋値上!$C7)＊B6

■Javaプログラムで使用するAPIの説明

java.util

public class LinkedList<E>

インタフェースListに基づく連結リストの実装である。インタフェースListのオプション操作はすべて実装され，nullを含むすべての要素が許される。インタフェースListの実装のほかに，クラスLinkedListには，リストの最初又は最後の要素を取り出したり，先頭又は末尾に要素を追加するメソッドが用意されている。それらの操作により，LinkedListはスタック又はキューとして使用可能である。

この実装は同期化を行わない。複数スレッドが並行して一つのLinkedListのインスタンスにアクセスし，そのうちの少なくとも一つのスレッドがリストに対して構造的な変更をする場合は，リストを使用する側で同期化を行わなければならない（構造的な変更とは一つ以上の要素の追加や削除を伴う操作であり，単に要素の値を設定する操作は構造的な変更ではない）。

コンストラクタ

public LinkedList()
空のリストを作る。

メソッド

public boolean add(E e)
リストの最後に要素を追加する。
引数： e — リストに追加する要素
戻り値：true

public boolean isEmpty()
リストが空かどうか調べる。
戻り値：リストが空のときtrue

public E removeFirst()
リストの最初の要素を取り除く。
戻り値：削除した要素
例外： NoSuchElementException — リストが空のとき

public int size()
リスト内の現在の要素の個数を調べる。
戻り値：現在の要素の個数

java.util

public interface Map<K, V>

型 K のキーに型 V の値を対応付けて保持するインタフェースを提供する。各キーは，一つの値としか対応付けられない。

メソッド

---

public boolean containsValue(Object value)

引数 value で指定された値と等価な値をもつ一つ以上のキーがマップに存在するかどうかを調べる。

引数： value — このマップに存在するかどうかを調べる値

戻り値：このマップの一つ以上のキーが指定された値に対応付けられているとき true

---

public V get(Object key)

指定されたキーに対応付けられた値を得る。

引数： key — キー

戻り値：指定されたキーに対応付けられた型 V の値
このキーと値の対応付けがなければ null

---

public Set<K> keySet()

登録されているキーの集合を得る。

戻り値：登録されているキーの集合

---

public V put(K key, V value)

指定されたキーに指定された値を対応付けて登録する。このキーが既にほかの値と対応付けられていれば，その値を指定された値に置き換える。

引数： key — キー
value — 値

戻り値：指定されたキーに対応付けられていた型 V の値
このキーと値の対応付けがなければ null

---

public V remove(Object key)

指定されたキーの対応付けが登録されていれば，削除する。

引数： key — キー

戻り値：指定されたキーに対応付けられていた型 V の値
このキーと値の対応付けがなければ null

java.util

public class HashMap<K, V>

インタフェース Map のハッシュを用いた実装である。キー及び値は，null でもよい。

この実装は同期化を行わない。複数スレッドが並行して一つの HashMap のインスタンスにアクセスし，そのうちの少なくとも一つのスレッドがマップに対して構造的な変更をする場合は，マップを使用する側で同期化を行わなければならない（構造的な変更とは一つ以上のマッピングの追加や削除を伴う操作であり，単に既存のキーに対応付けられた値を変更する操作は構造的な変更ではない）。

コンストラクタ

public HashMap()
空の HashMap を作る。

メソッド

public boolean containsValue(Object value)
インタフェース Map のメソッド containsValue と同じ

public V get(Object key)
インタフェース Map のメソッド get と同じ

public Set<K> keySet()
インタフェース Map のメソッド keySet と同じ

public V put(K key, V value)
インタフェース Map のメソッド put と同じ

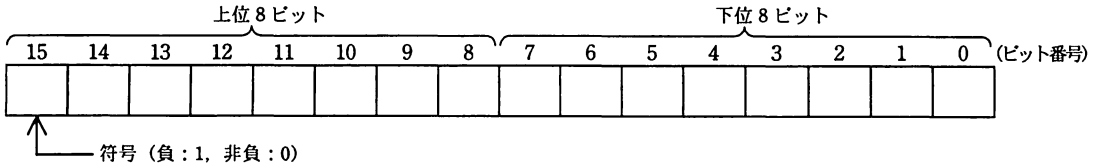
public V remove(Object key)
インタフェース Map のメソッド remove と同じ

■アセンブラ言語の仕様

1. システム COMETⅡの仕様

1.1 ハードウェアの仕様

(1) 1語は16ビットで，そのビット構成は，次のとおりである。



(2) 主記憶の容量は65536語で，そのアドレスは0～65535番地である。

(3) 数値は，16ビットの2進数で表現する。負数は，2の補数で表現する。

(4) 制御方式は逐次制御で，命令語は1語長又は2語長である。

(5) レジスタとして，GR（16ビット），SP（16ビット），PR（16ビット），FR（3ビット）の4種類がある。

GR（汎用レジスタ，General Register）は，GR0～GR7 の 8 個があり，算術，論理，比較，シフトなどの演算に用いる。このうち，GR1～GR7 のレジスタは，指標レジスタ（index register）としてアドレスの修飾にも用いる。

SP（スタックポインタ，Stack Pointer）は，スタックの最上段のアドレスを保持している。

PR（プログラムレジスタ，Program Register）は，次に実行すべき命令語の先頭アドレスを保持している。

FR（フラグレジスタ，Flag Register）は，OF（Overflow Flag），SF（Sign Flag），ZF（Zero Flag）と呼ぶ3個のビットからなり，演算命令などの実行によって次の値が設定される。これらの値は，条件付き分岐命令で参照される。

OF	算術演算命令の場合は，演算結果が−32768～32767 に収まらなくなったとき 1 になり，それ以外のとき 0 になる。論理演算命令の場合は，演算結果が 0～65535 に収まらなくなったとき 1 になり，それ以外のとき 0 になる。
SF	演算結果の符号が負（ビット番号 15 が 1）のとき 1，それ以外のとき 0 になる。
ZF	演算結果が零（全部のビットが 0）のとき 1，それ以外のとき 0 になる。

(6) 論理加算又は論理減算は，被演算データを符号のない数値とみなして，加算又は減算する。

1.2 命令

命令の形式及びその機能を示す。ここで，一つの命令コードに対し 2 種類のオペランドがある場合，上段はレジスタ間の命令，下段はレジスタと主記憶間の命令を表す。

命　令	書き方		命令の説明	FRの設定
	命令コード	オペランド		

(1) ロード，ストア，ロードアドレス命令

ロード LoaD	LD	r1,r2	r1 ← (r2)	○*1
		r,adr [,x]	r ← (実効アドレス)	
ストア STore	ST	r,adr [,x]	実効アドレス ← (r)	−
ロードアドレス Load ADdress	LAD	r,adr [,x]	r ← 実効アドレス	

(2) 算術，論理演算命令

算術加算 ADD Arithmetic	ADDA	r1,r2	r1 ← (r1) + (r2)	○
		r,adr [,x]	r ← (r) + (実効アドレス)	
論理加算 ADD Logical	ADDL	r1,r2	r1 ← (r1) $+_L$ (r2)	
		r,adr [,x]	r ← (r) $+_L$ (実効アドレス)	
算術減算 SUBtract Arithmetic	SUBA	r1,r2	r1 ← (r1) − (r2)	
		r,adr [,x]	r ← (r) − (実効アドレス)	
論理減算 SUBtract Logical	SUBL	r1,r2	r1 ← (r1) $-_L$ (r2)	
		r,adr [,x]	r ← (r) $-_L$ (実効アドレス)	
論理積 AND	AND	r1,r2	r1 ← (r1) AND (r2)	○*1
		r,adr [,x]	r ← (r) AND (実効アドレス)	
論理和 OR	OR	r1,r2	r1 ← (r1) OR (r2)	
		r,adr [,x]	r ← (r) OR (実効アドレス)	
排他的論理和 eXclusive OR	XOR	r1,r2	r1 ← (r1) XOR (r2)	
		r,adr [,x]	r ← (r) XOR (実効アドレス)	

(3) 比較演算命令

算術比較 ComPare Arithmetic	CPA	r1,r2 r,adr [,x]	(r1) と (r2)，又は (r) と (実効アドレス) の算術比較又は論理比較を行い，比較結果によって，FR に次の値を設定する。	○*1
論理比較 ComPare Logical	CPL	r1,r2 r,adr [,x]	

比較結果	FR の値 SF	FR の値 ZF
(r1) ＞ (r2) (r) ＞ (実効アドレス)	0	0
(r1) ＝ (r2) (r) ＝ (実効アドレス)	0	1
(r1) ＜ (r2) (r) ＜ (実効アドレス)	1	0

(4) シフト演算命令

算術左シフト Shift Left Arithmetic	SLA	r,adr [,x]	符号を除き (r) を実効アドレスで指定したビット数だけ左又は右にシフトする。 シフトの結果，空いたビット位置には，左シフトのときは 0，右シフトのときは符号と同じものが入る。	○*2
算術右シフト Shift Right Arithmetic	SRA	r,adr [,x]		
論理左シフト Shift Left Logical	SLL	r,adr [,x]	符号を含み (r) を実効アドレスで指定したビット数だけ左又は右にシフトする。 シフトの結果，空いたビット位置には 0 が入る。	
論理右シフト Shift Right Logical	SRL	r,adr [,x]		

(5) 分岐命令

正分岐 Jump on PLus	JPL	adr [,x]	FR の値によって，実効アドレスに分岐する。分岐しないときは，次の命令に進む。	—
負分岐 Jump on MInus	JMI	adr [,x]		
非零分岐 Jump on Non Zero	JNZ	adr [,x]		
零分岐 Jump on ZEro	JZE	adr [,x]		
オーバフロー分岐 Jump on OVerflow	JOV	adr [,x]		
無条件分岐 unconditional JUMP	JUMP	adr [,x]	無条件に実効アドレスに分岐する。	

命令	分岐するときの FR の値 OF	SF	ZF
JPL		0	0
JMI		1	
JNZ			0
JZE			1
JOV	1		

(6) スタック操作命令

プッシュ PUSH	PUSH adr [,x]	SP ← (SP) $-_L$ 1, (SP) ← 実効アドレス	—
ポップ POP	POP r	r ← ((SP)), SP ← (SP) $+_L$ 1	

(7) コール，リターン命令

コール CALL subroutine	CALL adr [,x]	SP ← (SP) $-_L$ 1, (SP) ← (PR), PR ← 実効アドレス	—
リターン RETurn from subroutine	RET	PR ← ((SP)), SP ← (SP) $+_L$ 1	

(8) その他

スーパバイザコール SuperVisor Call	SVC adr [,x]	実効アドレスを引数として割出しを行う。実行後の GR と FR は不定となる。	—
ノーオペレーション No OPeration	NOP	何もしない。	

(注) r, r1, r2　いずれも GR を示す。指定できる GR は GR0 ～ GR7
adr　アドレスを示す。指定できる値の範囲は 0 ～ 65535
x　指標レジスタとして用いる GR を示す。指定できる GR は GR1 ～ GR7
[ ]　[ ] 内の指定は省略できることを示す。
( )　( ) 内のレジスタ又はアドレスに格納されている内容を示す。
実効アドレス　adr と x の内容との論理加算値又はその値が示す番地
←　演算結果を，左辺のレジスタ又はアドレスに格納することを示す。
$+_L$, $-_L$　論理加算，論理減算を示す。
FR の設定　○　：設定されることを示す。
○*1 ：設定されることを示す。ただし，OF には 0 が設定される。
○*2 ：設定されることを示す。ただし，OF にはレジスタから最後に送り出されたビットの値が設定される。
－　：実行前の値が保持されることを示す。

1.3 文字の符号表

(1) JIS X 0201 ラテン文字・片仮名用 8 ビット符号で規定する文字の符号表を使用する。
(2) 右に符号表の一部を示す。1 文字は 8 ビットからなり，上位 4 ビットを列で，下位 4 ビットを行で示す。例えば，間隔，4，H，¥ のビット構成は，16 進表示で，それぞれ 20，34，48，5C である。16 進表示で，ビット構成が 21 ～ 7E（及び表では省略している A1 ～ DF）に対応する文字を図形文字という。図形文字は，表示（印刷）装置で，文字として表示（印字）できる。
(3) この表にない文字とそのビット構成が必要な場合は，問題中で与える。

行＼列	02	03	04	05	06	07
0	間隔	0	@	P	`	p
1	!	1	A	Q	a	q
2	"	2	B	R	b	r
3	#	3	C	S	c	s
4	$	4	D	T	d	t
5	%	5	E	U	e	u
6	&	6	F	V	f	v
7	'	7	G	W	g	w
8	(	8	H	X	h	x
9	)	9	I	Y	i	y
10	*	:	J	Z	j	z
11	+	;	K	[	k	{
12	,	<	L	¥	l	|
13	-	=	M	]	m	}
14	.	>	N	^	n	~
15	/	?	O	_	o	

2. アセンブラ言語 CASLⅡの仕様

2.1 言語の仕様

(1) CASLⅡは，COMETⅡのためのアセンブラ言語である。
(2) プログラムは，命令行及び注釈行からなる。
(3) 1命令は1命令行で記述し，次の行へ継続できない。
(4) 命令行及び注釈行は，次に示す記述の形式で，行の1文字目から記述する。

行の種類		記述の形式
命令行	オペランドあり	[ラベル] {空白} {命令コード} {空白} {オペランド} [{空白} [コメント]]
	オペランドなし	[ラベル] {空白} {命令コード} [{空白} [{;} [コメント]]]
注釈行		[空白] {;} [コメント]

(注) [ ] [ ] 内の指定が省略できることを示す。
{ } { } 内の指定が必須であることを示す。
ラベル その命令の（先頭の語の）アドレスを他の命令やプログラムから参照するための名前である。長さは 1～8 文字で，先頭の文字は英大文字でなければならない。以降の文字は，英大文字又は数字のいずれでもよい。なお，予約語である GR0～GR7 は，使用できない。
空白 1文字以上の間隔文字の列である。
命令コード 命令ごとに記述の形式が定義されている。
オペランド 命令ごとに記述の形式が定義されている。
コメント 覚え書きなどの任意の情報であり，処理系で許す任意の文字を書くことができる。

2.2 命令の種類

命令は，4種類のアセンブラ命令（START，END，DS，DC），4種類のマクロ命令（IN，OUT，RPUSH，RPOP）及び機械語命令（COMETⅡの命令）からなる。その仕様を次に示す。

命令の種類	ラベル	命令コード	オペランド	機能
アセンブラ命令	ラベル	START	[実行開始番地]	プログラムの先頭を定義 プログラムの実行開始番地を定義 他のプログラムで参照する入口名を定義
		END		プログラムの終わりを明示
	[ラベル]	DS	語数	領域を確保
	[ラベル]	DC	定数 [,定数] …	定数を定義
マクロ命令	[ラベル]	IN	入力領域,入力文字長領域	入力装置から文字データを入力
	[ラベル]	OUT	出力領域,出力文字長領域	出力装置へ文字データを出力
	[ラベル]	RPUSH		GR の内容をスタックに格納
	[ラベル]	RPOP		スタックの内容を GR に格納
機械語命令	[ラベル]	（「1.2 命令」を参照）		

2.3 アセンブラ命令

アセンブラ命令は，アセンブラの制御などを行う。

(1) START [実行開始番地]

START 命令は，プログラムの先頭を定義する。
実行開始番地は，そのプログラム内で定義されたラベルで指定する。指定がある場合はその番地から，省略した場合は START 命令の次の命令から，実行を開始する。
また，この命令につけられたラベルは，他のプログラムから入口名として参照できる。

(2) | END | |

END 命令は，プログラムの終わりを定義する。

(3) | DS | 語数 |

DS 命令は，指定した語数の領域を確保する。

語数は，10 進定数（≧0）で指定する。語数を 0 とした場合，領域は確保しないが，ラベルは有効である。

(4) | DC | 定数［,定数］… |

DC 命令は，定数で指定したデータを（連続する）語に格納する。

定数には，10 進定数，16 進定数，文字定数，アドレス定数の 4 種類がある。

定数の種類	書き方	命令の説明
10 進定数	n	n で指定した 10 進数値を，1 語の 2 進数データとして格納する。ただし，n が−32768～32767 の範囲にないときは，その下位 16 ビットを格納する。
16 進定数	#h	h は 4 けたの 16 進数（16 進数字は 0～9，A～F）とする。h で指定した 16 進数値を 1 語の 2 進数データとして格納する（0000≦h≦FFFF）。
文字定数	'文字列'	文字列の文字数（＞0）分の連続する領域を確保し，最初の文字は第 1 語の下位 8 ビットに，2 番目の文字は第 2 語の下位 8 ビットに，…と順次文字データとして格納する。各語の上位 8 ビットには 0 のビットが入る。 文字列には，間隔及び任意の図形文字を書くことができる。ただし，アポストロフィ（'）は 2 個続けて書く。
アドレス定数	ラベル	ラベルに対応するアドレスを 1 語の 2 進数データとして格納する。

2.4 マクロ命令

マクロ命令は，あらかじめ定義された命令群とオペランドの情報によって，目的の機能を果たす命令群を生成する（語数は不定）。

(1) | IN | 入力領域,入力文字長領域 |

IN 命令は，あらかじめ割り当てた入力装置から，1 レコードの文字データを読み込む。

入力領域は，256 語長の作業域のラベルであり，この領域の先頭から，1 文字を 1 語に対応させて順次入力される。レコードの区切り符号（キーボード入力の復帰符号など）は，格納しない。格納の形式は，DC 命令の文字定数と同じである。入力データが 256 文字に満たない場合，入力領域の残りの部分は実行前のデータを保持する。入力データが 256 文字を超える場合，以降の文字は無視される。

入力文字長領域は，1 語長の領域のラベルであり，入力された文字の長さ（≧0）が 2 進数で格納される。ファイルの終わり（end of file）を検出した場合は，−1 が格納される。

IN 命令を実行すると，GR の内容は保存されるが，FR の内容は不定となる。

(2) | OUT | 出力領域,出力文字長領域 |

OUT 命令は，あらかじめ割り当てた出力装置に，文字データを，1 レコードとして書き出す。

出力領域は，出力しようとするデータが 1 文字 1 語で格納されている領域のラベルである。格納の形式は，DC 命令の文字定数と同じであるが，上位 8 ビットは，OS が無視するので 0 でなくてもよい。

出力文字長領域は，1 語長の領域のラベルであり，出力しようとする文字の長さ（≧0）を 2 進数で格納しておく。

OUT 命令を実行すると，GR の内容は保存されるが，FR の内容は不定となる。

(3) | RPUSH |

RPUSH 命令は，GR の内容を，GR1，GR2，…，GR7 の順序でスタックに格納する。

(4) | RPOP |

RPOP 命令は，スタックの内容を順次取り出し，GR7，GR6，…，GR1 の順序で GR に格納する。

2.5 機械語命令

機械語命令のオペランドは，次の形式で記述する。

r, r1, r2	GR は，記号 GR0 ～ GR7 で指定する。
x	指標レジスタとして用いる GR は，記号 GR1 ～ GR7 で指定する。
adr	アドレスは，10 進定数，16 進定数，アドレス定数又はリテラルで指定する。 リテラルは，一つの 10 進定数，16 進定数又は文字定数の前に等号（=）を付けて記述する。CASL Ⅱ は，等号の後の定数をオペランドとする DC 命令を生成し，そのアドレスを adr の値とする。

2.6 その他

(1) アセンブラによって生成される命令語や領域の相対位置は，アセンブラ言語での記述順序とする。ただし，リテラルから生成される DC 命令は，END 命令の直前にまとめて配置される。

(2) 生成された命令語，領域は，主記憶上で連続した領域を占める。

3. プログラム実行の手引

3.1 OS

プログラムの実行に関して，次の取決めがある。

(1) アセンブラは，未定義ラベル（オペランド欄に記述されたラベルのうち，そのプログラム内で定義されていないラベル）を，他のプログラムの入口名（START 命令のラベル）と解釈する。この場合，アセンブラはアドレスの決定を保留し，その決定を OS に任せる。OS は，実行に先立って他のプログラムの入口名との連係処理を行いアドレスを決定する（プログラムの連係）。

(2) プログラムは，OS によって起動される。プログラムがロードされる主記憶の領域は不定とするが，プログラム中のラベルに対応するアドレス値は，OS によって実アドレスに補正されるものとする。

(3) プログラムの起動時に，OS はプログラム用に十分な容量のスタック領域を確保し，その最後のアドレスに 1 を加算した値を SP に設定する。

(4) OS は，CALL 命令でプログラムに制御を渡す。プログラムを終了し OS に制御を戻すときは，RET 命令を使用する。

(5) IN 命令に対応する入力装置，OUT 命令に対応する出力装置の割当ては，プログラムの実行に先立って利用者が行う。

(6) OS は，入出力装置や媒体による入出力手続の違いを吸収し，システムでの標準の形式及び手続（異常処理を含む）で入出力を行う。したがって，IN，OUT 命令では，入出力装置の違いを意識する必要はない。

3.2 未定義事項

プログラムの実行等に関し，この仕様で定義しない事項は，処理系によるものとする。

表計算ソフトの機能・用語

表計算ソフトの機能，用語などは，原則として次による。

1. **ワークシート**

表計算ソフトの作業領域をワークシートという。ワークシートの大きさは256列（列Aから列Z，列AAから列AZ，さらに列BAから列BZと続き，列IVまで続く），10,000行（行1から行10,000まで）とする。

2. **セル**

(1) ワークシートを縦・横に分割したときの一つのます目をセルという。列A行1のセルはA1と表す。

(2) 長方形の形をしたセルの集まりを範囲として指定することができる。範囲の指定はA1～B3のように表す。

(3) 範囲に名前を付けることができる。範囲名は[ ]を用いて，“セルA1～B3に[金額]と名前を付ける”などと表す。

(4) データが入力されていないセルを，空白セルという。

3. **セルへの入力**

(1) セルに数値，文字列，計算式を入力できる。

(2) セルを保護すると，そのセルへの入力を不可能にすることができる。セルの保護を解除すると，そのセルへの入力が再び可能になる。

(3) セルA1に数値5を入力するときは，“セルA1に5を入力”と表す。

(4) セルB2に，文字列ABCを入力するときは，“セルB2に'ABC'を入力”と表す。

(5) セルC3に，セルA1とセルB2の和を求める計算式を入力するときは，“セルC3に計算式A1＋B2を入力”などと表す。

4. **セルの内容の表示**

(1) セルに数値を入力すると，右詰めで表示される。

(2) セルに文字列を入力すると，左詰めで表示される。

(3) セルに計算式を入力すると，計算結果が数値ならば右詰めで，文字列ならば左詰めで表示される。

(4) セルの内容の表示については，左詰め，中央揃(そろ)え，右詰めに変更できる。

5. **計算式**

(1) 計算式には，数学で用いられる数式が利用できる。

(2) 計算式で使用する算術演算子は，“＋”（加算），“－”（減算），“＊”（乗算），“／”（除算）及び“＾”（べき算）とする。

(3) 算術演算子による計算の優先順位は，数学での優先順位と同じである。

6. 再計算

(1) セルに計算式を入力すると，直ちに計算結果を表示する。

(2) セルの数値が変化すると，そのセルを参照しているセルも自動的に再計算される。この再計算はA1，A2，A3，…，B1，B2，B3，…の順に1回だけ行われる。

7. 関数

(1) 計算式には次の表で定義する関数を利用することができる。

関数名と使用例	解　　説
合計(A1～A5)	セルA1からセルA5までの範囲のすべての数値の合計を求める。
平均(B2～F2)	セルB2からセルF2までの範囲のすべての数値の平均を求める。
平方根(I6)	セルI6の値（正の数値でなければならない）の正の平方根を求める。
標準偏差(D5～D19)	セルD5からセルD19までの範囲のすべての数値の標準偏差を求める。
最大(C3～E7)	セルC3からセルE7までの範囲のすべての数値のうちの最大値を求める。
最小([得点])	[得点]と名前を付けた範囲のすべての数値のうちの最小値を求める。
IF(B3＞A4，'北海道'，'九州')	第1引数に指定された論理式が真（成立する）ならば第2引数が，偽（成立しない）ならば第3引数が求める値となる。左の例では，セルB3がA4より大きければ文字列'北海道'が，それ以外の場合には文字列'九州'が求める値となる。論理式中では，比較演算子として，＝，≠，＞，＜，≦，≧を利用することができる。第2引数，第3引数に，更にIF関数を利用して，IF関数を入れ子にすることができる。
個数(G1～G5)	セルG1からG5までの範囲のうち，空白セルでないセルの個数を求める。
条件付個数(H5～H9，'＞25')	第1引数に指定された範囲のうち，第2引数に指定された条件を満たすセルの個数を求める。左の例では，セルH5からH9までの範囲のうち，値として25より大きな数値を格納しているセルの個数を求める。
整数部(A3)	セルA3の値（数値でなければならない）を超えない最大の整数を求める。 例えば， 整数部(3.9)＝3 整数部(－3.9)＝－4 となる。
剰余(C4，D4)	セルC4の値を被除数，D4の値を除数とし，被除数を除数で割ったときの剰余を求める。剰余の値は常に除数と同じ符号をもつ。“剰余”関数と“整数部”関数は，次の関係を満たしている。 剰余(x，y)＝x－y＊整数部(x／y)
論理積(論理式1，論理式2，…)	引数として指定された論理式がすべて真であれば，真を返す。引数のうち一つでも偽のものがあれば，偽を返す。引数として指定できる論理式の数は任意である。
論理和(論理式1，論理式2，…)	引数として指定された論理式がすべて偽であれば，偽を返す。引数のうち一つでも真のものがあれば，真を返す。引数として指定できる論理式の数は任意である。
否定(論理式)	引数として指定された論理式が真であれば偽を，偽であれば真を返す。
注　“合計”，“平均”，“標準偏差”，“最大”，“最小”は，引数で指定された範囲のセルのうち，値として数値以外を格納しているものは無視する。	

(2) 関数の引数には，セルを用いた計算式，範囲，範囲名，論理式を指定することができる。

8. **セルの複写**

(1) セルに入力された数値，文字列，計算式を他のセルに複写することができる。

(2) セルに入力された計算式が他のセルを参照している場合は，複写先のセルでは相対的にセルが自動的に変更される。例えば，セルA6に合計（A1～A5）を入力した場合，セルA6をセルB7に複写すると，セルB7の計算式は合計（B2～B6）となる。

9. **絶対参照**

(1) 計算式を複写しても参照したセルが変わらない参照を絶対参照といい，記号＄を用いて＄A＄1などと表す。例えば，セルB1に計算式＄A＄1＋5を入力した場合，セルB1をセルC4に複写してもセルC4の計算式は＄A＄1＋5のままである。

(2) 絶対参照は行と列の一方だけについても指定可能であり，＄A1，A＄1などと表す。例えば，セルD2に計算式＄C1－3を入力した場合，セルD2をセルE3に複写すると，セルE3の計算式は＄C2－3となる。また，セルG3に計算式F＄2－3を入力した場合，セルG3をH4に複写すると，セルH4の計算式はG＄2－3となる。

10. **マクロ**

(1) ワークシートには幾つかのマクロを保存できる。マクロはマクロP，マクロQなどと表す。

(2) マクロについては“マクロPを実行するとワークシートを保存する。”，“セルA1からセルA10までを昇順に並べ替える手続をマクロQに登録する。”，“マクロR：数値を入力。”，“C列のデータがその数値以下のものを抽出する。”などと記述する。

11. **その他**

ワークシートの“保存”，“読出し”，“印刷”や，罫線機能，グラフ化機能など市販されている多くの表計算ソフトに備わっている機能は使用できるものとする。

7. 途中で退室する場合には，手を挙げて監督員に合図し，答案用紙が回収されてから静かに退室してください。

退室可能時間	13:40 ～ 15:20

8. **問題に関する質問にはお答えできません。**文意どおり解釈してください。
9. 問題冊子の余白などは，適宜利用して構いません。
10. Java プログラムで使用する API の説明，アセンブラ言語の仕様及び表計算ソフトの機能・用語は，この冊子の末尾を参照してください。
11. 試験時間中，机上に置けるもの及び使用できるものは，次のものに限ります。
    なお，会場での貸出しは行っていません。
    受験票，B又はHBの黒鉛筆又はシャープペンシル，鉛筆削り，消しゴム，定規，時計（アラームなど時計以外の機能は使用不可），ハンカチ，ティッシュ
    これら以外は机上に置けません。使用もできません。
12. 試験終了後，この問題冊子は持ち帰ることができます。
13. 答案用紙は，いかなる場合でも提出してください。回収時に提出しない場合は，採点されません。
14. 試験時間中にトイレへ行きたくなったり，気分が悪くなったりした場合は，手を挙げて監督員に合図してください。

お知らせ

1. システムの構築や試験会場の確保などの諸準備が整えば，平成23年11月からITパスポート試験においてCBT※方式による試験を実施する予定です。
2. CBT方式による試験の実施に伴い，現行の筆記による試験は，廃止する予定です。
3. 詳細が決定しましたら，ホームページなどでお知らせします。

※CBT（Computer Based Testing）：コンピュータを使用して実施する試験。