

# 平成 25 年度 春期 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

- 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。
- 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。
- 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つてから始めてください。
- 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

- 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。

- 答案用紙は光学式読み取り装置で読み取った上で採点しますので、B 又は HB の黒鉛筆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正しくマークされていない場合は読み取れません。特にシャープペンシルを使用する際には、マークの濃度に十分注意してください。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、消しきずを残さないでください。
- 受験番号欄に受験番号を、生年月日欄に受験票の生年月日を記入及びマークしてください。答案用紙のマークの記入方法のとおり記入及び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。生年月日欄については、受験票の生年月日を訂正した場合でも、訂正前の生年月日を記入及びマークしてください。
- 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。答案用紙のマークの記入方法のとおりマークされていない場合は、採点されません。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 問を探点します。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 問を探点します。
- 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。答案用紙のマークの記入方法のとおりマークされていない場合は、採点されません。

[例題] 次の [ ] に入る正しい答えを、解答群の中から選べ。

春の情報処理技術者試験は、[ a ] 月に実施される。

解答群 ア 2 イ 3 ウ 4 エ 5

正しい答えは“ウ 4”ですから、次のようにマークしてください。

例題	a	(ア)	(イ)	(ウ)	(エ)	(オ)	(カ)	(キ)	(ク)	(ケ)	(コ)
----	---	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----

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	カラー画像
問 2	ソフトウェア	仮想記憶方式
問 3	データベース	会員情報を管理する関係データベースの設計と運用
問 4	情報セキュリティ	IC カードを利用した入退室管理システム
問 5	ソフトウェア設計	社員の歩合給決定処理
問 6	プロジェクトマネジメント	ソフトウェア開発の品質管理
問 7	経営戦略・企業と法務	市場分析と需要予測

●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	食料品店の値引き処理

●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)	ケーブルテレビ局が提供するサービスの料金計算
問 10	ソフトウェア開発 (COBOL)	従業員へ支給する特別給の算出
問 11	ソフトウェア開発 (Java)	あみだくじ
問 12	ソフトウェア開発 (アセンブラー)	ビット列の置換え
問 13	ソフトウェア開発 (表計算)	製品の生産計画

## 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。
• 変数 $\leftarrow$ 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。 開始時点での変数に初期値（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	
乗除演算	×, ÷, %	
加減演算	+, -	
関係演算	>, <, ≥, ≤, =, ≠	
論理積	and	
論理和	or	

注記 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問以上マークした場合には、はじめの5問について採点します。

問1 カラー画像に関する次の記述を読んで、設問1~3に答えよ。

赤、緑、青の色の光（以下、色という）を、光の3原色という。赤、緑、青の色を発光させて重ね合わせることによって様々な色を表現することができる。緑と青の色を重ね合わせるとシアンに、青と赤の色を重ね合わせるとマゼンタに、赤と緑の色を重ね合わせると黄に、赤緑青全ての色を重ね合わせると白になる。光の3原色のどれも発光していないと黒になる。光の3原色による色の表現を、図1に示す。ここで、図1中の記号は、表1に示す色を表す。

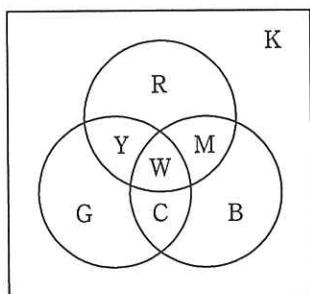


図1 光の3原色による色の表現

表1 記号と色の対応

記号	色
R	赤 (Red)
G	緑 (Green)
B	青 (Blue)
C	シアン (Cyan)
M	マゼンタ (Magenta)
Y	黄 (Yellow)
W	白 (White)
K	黒 (black)

設問 1 ディスプレイにカラー画像を表示するために、1 画素を 3 ビットで表現することにする。3 ビットの先頭（左端）から各ビットに赤、緑、青の色の情報を順に割り当て、2 階調（1 のとき発光、0 のとき非発光）で表現する。この 3 ビットのビットパターンで 8 色を表現することができる。色とビットパターンの対応を表 2 に示す。[ ] に入る正しい答えを、解答群の中から選べ。

表 2 色とビットパターンの対応

色	ビットパターン
赤 (Red)	100
緑 (Green)	010
青 (Blue)	001
シアン (Cyan)	
マゼンタ (Magenta)	[ ] a
黄 (Yellow)	
白 (White)	111
黒 (black)	000

注記 網掛けの部分は表示していない。

解答群

ア 011

イ 101

ウ 110

設問 2 次の記述中の  に入る正しい答えを、解答群の中から選べ。

ディスプレイに画像を表示するとき、画像データは、ビデオ RAM（以下、VRAM という）と呼ばれるメモリに格納されている。

カラー画像データを VRAM に格納する方法の一つに、プレーンドアクセス方式がある。プレーンドアクセス方式では、VRAM 上にディスプレイの画素数と同じ数のビットをもつプレーンという区分を複数用意する。各プレーンの先頭に位置するビットをディスプレイの最左上の画素に対応づけ、ディスプレイの左から右、上から下の画素へと順にプレーンのビットを割り当てる。赤、緑、青それぞれの色を 2 階調で表現する場合、色の情報を格納するために、VRAM 上にプレーン 1、プレーン 2、プレーン 3 と呼ぶ、三つの区分を用意する。プレーン 1、プレーン 2、プレーン 3 のそれぞれを、赤、緑、青の色に割り当て、各プレーンの同じ位置のビットを取り出した 3 ビットで、1 画素を表現する。

例えば、プレーン 1 の先頭ビットが 0、プレーン 2 の先頭ビットが 1、プレーン 3 の先頭ビットが 1 のとき、ディスプレイの最左上の画素の色は  b となる。

VRAM の内容が図 2 のとおりであった場合、各プレーンの先頭から数えて 6 番目のビットに対応するディスプレイの画素の色は  c となる。ここで、VRAM の内容は 16 進数で表記している。

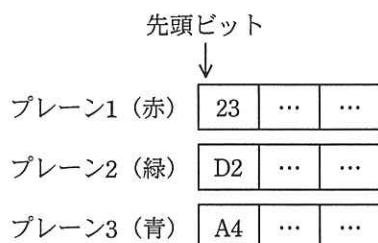


図 2 VRAM の内容（プレーンドアクセス方式）

### 解答群

ア 青	イ 赤	ウ 黄
エ 黒	オ シアン	カ 白
キ マゼンタ	ク 緑	

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。

設問 2 のプレーンドアクセス方式では、赤、緑、青それぞれの色に一つのプレーンを用意することによって、8色を表現することができた。一つの色に複数のプレーンを用意することによって、その色の階調数を増やすことができる。その結果、より多くの色を表現することができるようになる。

- (1) VRAM 上に五つの区分を用意し、各区分をプレーン 1, プレーン 2, …, プレーン 5 と呼ぶ。各プレーンの同じ位置のビットを取り出した 5 ビットで、1 画素を表現する。プレーン 1 を赤に、プレーン 2 とプレーン 3 を緑に、プレーン 4 とプレーン 5 を青に割り当てる。

このとき、赤は 2 階調、緑と青はそれぞれ  d 階調となり、この 5 ビットで  e 色を表現することができる。

- (2) 縦 600×横 800 画素のディスプレイに 16 色を表現するためには、少なくとも  f k バイトの VRAM が必要である。ここで、1k バイトは 1,000 バイトとする。

### 解答群

ア 2	イ 4	ウ 8	エ 16	オ 32
カ 60	キ 64	ク 120	ケ 128	コ 240

問 2 仮想記憶方式に関する次の記述を読んで、設問 1～3 に答えよ。

OS の主記憶管理において、仮想記憶方式は、OS が提供する論理的な記憶領域（以下、仮想記憶という）上のアドレスと主記憶上の物理的なアドレスを対応付けて管理する方式である。仮想記憶方式では、補助記憶装置を仮想記憶として用いるので、仮想記憶上に主記憶の容量を超えるプログラムを格納することができる。仮想記憶上のアドレス空間を仮想アドレス空間、主記憶上のアドレス空間を物理アドレス空間と呼び、それぞれの空間の記憶場所を仮想アドレスと物理アドレスで指定する。

仮想記憶方式の一つに、ページング方式がある。ページング方式は、仮想アドレス空間と物理アドレス空間のそれぞれをページと呼ぶ固定長の領域に分割しておき、ページ単位でアドレス空間を管理する。ページング方式による仮想アドレス空間のページと物理アドレス空間のページの対応例を、図 1 に示す。図 1 では、補助記憶装置に格納されているプログラム A は  $a_1, a_2, a_3, a_4, a_5$  に分割されて、仮想ページ番号 1～5 のページに格納されている。

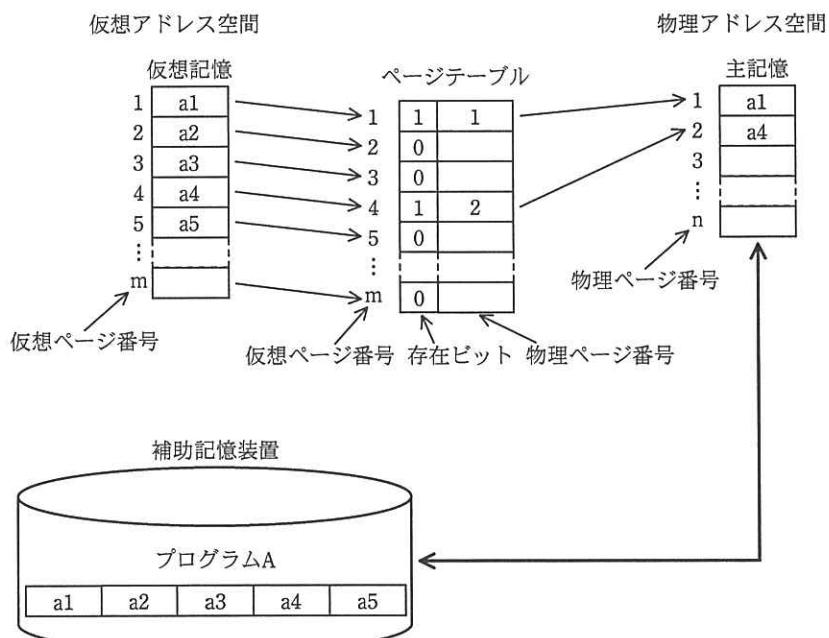


図 1 仮想アドレス空間のページと物理アドレス空間のページの対応例

仮想アドレス空間及び物理アドレス空間の各ページには、先頭から順に番号を付け、それぞれを仮想ページ番号、物理ページ番号と呼ぶ。仮想ページと物理ページの対応は、ページテーブルで管理する。ページテーブルの要素の個数は仮想ページの個数と同じであり、各要素が仮想ページの 1 ページに対応している。ページテーブルでは、仮想ページの内容が物理アドレス空間にも存在しているかどうかを示すビット（以下、存在ビットという）と物理ページ番号が管理されている。存在ビットは、ページが存在しているとき 1、存在していないとき 0 とする。

設問 1 次の記述中の [ ] に入る正しい答えを、解答群の中から選べ。

プログラムの実行過程で存在ビットを調べ、プログラムの実行に必要なページが [ a ] に存在していないときには、ページフォールトという割込みが発生する。ページフォールトが発生すると、ページアウトやページインなどのページ置換処理が実行される。ページ置換処理のアルゴリズムには、ページインしてから最も時間が経過しているページを置換対象とする FIFO アルゴリズムや、参照されていない時間が最も長いページを置換対象とする [ b ] アルゴリズムなどがある。

#### 解答群

ア LFU

イ LIFO

ウ LRU

エ 仮想アドレス空間

オ 物理アドレス空間

**設問2** プログラムAを実行するために割り当てられた物理アドレス空間の物理ページの個数が3の場合を考える。プログラムAの実行過程において、物理アドレス空間にa1, a2, a3が存在している状態でa4を参照するとページフォールトが発生する。このページフォールトが発生した後の処理の流れとして適切な答えを、解答群の中から選べ。ここで、解答群中の処理は左から右に向かって行うものとする。

#### 【処理の単位】

- ① 退避させるページをページアウトする。
- ② ページ置換えアルゴリズムによって、物理アドレス空間からページアウトするページを決定する。
- ③ 実行に必要なページをページインする。
- ④ ページアウトしたページに対応するページテーブルの要素の存在ビットを0にする。
- ⑤ ページインしたページに対応するページテーブルの要素の存在ビットを1にする。
- ⑥ ページアウトしたページに対応するページテーブルの要素の物理ページ番号を設定する。
- ⑦ ページインしたページに対応するページテーブルの要素の物理ページ番号を設定する。

#### 解答群

- |               |               |
|---------------|---------------|
| ア ①→②→③→④→⑤→⑥ | イ ①→③→②→④→⑦→⑤ |
| ウ ②→①→④→③→⑤→⑥ | エ ②→①→④→③→⑦→⑤ |
| オ ②→③→①→④→⑤→⑥ | カ ②→③→⑤→⑥→①→④ |

設問3 次の記述中の [ ] に入る正しい答えを、解答群の中から選べ。

ページ置換えアルゴリズムとして FIFO アルゴリズムを採用する。プログラムの実行過程で仮想ページが次の順で参照されるとき、物理ページの個数が 3 の場合のページフォールトの回数は [ c ] 回である。そして、物理ページの個数を 4 に増やした場合のページフォールトの回数は [ d ] 回である。ここで、プログラムの実行開始時点では、物理アドレス空間にはどのページも存在していないものとする。

【仮想ページの参照順を示す仮想ページ番号の並び】

1 → 4 → 3 → 2 → 1 → 4 → 5 → 1 → 4 → 3 → 2 → 5 → 1

解答群

ア 8

イ 9

ウ 10

エ 11

オ 12

問3 会員情報を管理する関係データベースの設計と運用に関する次の記述を読んで、設問1～4に答えよ。

ある地域で5店舗のヘアサロンを経営しているZ社では、会員登録した顧客に対して紙製の会員証を発行し、氏名や住所などの会員情報は表計算ソフトを利用して管理していた。今回、顧客サービスの向上を目的に、会員情報を管理するシステム（以下、管理システムという）を導入し、会員証のICカード化を行うとともに会員情報と来店記録のデータベース化を実施した。

管理していた会員情報の例とその形式は、図1のとおりである。

会員番号	氏名	住所	電話番号	性別	生年月日
00001	情報花子	東京都文京区桜丘2-28	03-1111-2222	女性	19870613

図1 管理していた会員情報の例とその形式

まず、会員情報と来店記録を管理するために図2に示すデータベースを設計した。下線付きの項目は主キーを表す。

会員表

会員番号	氏名	住所	電話番号	性別	生年月日
00001	情報花子	東京都文京区桜丘2-28	03-1111-2222	02	19870613

店舗表

店舗コード	店舗名	住所	電話番号
01	桜丘店	東京都文京区桜丘1-2	03-8888-9999

メニュー表

メニューコード	商品名	単価
001	カット	5000
002	パーマ	5000

会計表

会計コード	会員番号	店舗コード	来店日	会計金額
120001	00001	01	20130511	10000

明細表

会計コード	メニューコード
120001	001
120001	002

図2 会員情報と来店記録データのデータベースへの格納例

設問1 会員特典として、ポイント制度を導入することにした。ポイント情報の管理に関する次の記述中の [ ] に入る適切な答えを、解答群の中から選べ。

会員が料金を支払う際に会員証を提示すると、会計金額に応じて、千円につき1ポイントが付与される。会計時に付与されたポイント（以下、付与ポイントという）と、会員が現在保有しているポイント（以下、保有ポイントという）は、レシートに印字される。

会計の際、会員が希望すれば、保有ポイントと引き換えに表1に示す割引サービスを受けられる。このとき保有ポイントから引かれるポイントを利用ポイントという。付与ポイントは、割引後の料金から算出する。

表1 ポイントによる割引サービス

利用ポイント数	割引率 (%)
10	5
20	15
30	30

管理システムで会員の保有ポイントを参照する機会が多いので、参照の都度、保有ポイントの集計処理を実行することは避けたい。保有ポイントを管理するのに適した表は、aである。また、会員がポイントをどのように活用しているかの傾向を分析できるよう、ポイントの付与と利用の履歴を管理したい。付与ポイント及び利用ポイントを管理するのに適した表は、bである。

解答群

ア 会員表

イ 会計表

ウ 店舗表

エ 明細表

オ メニュー表

設問2 2012年9月にデータベース及び管理システムの開発が完了し、10月から運用を開始した。2012年12月について、店舗ごとの売上金額、延べ来客数、会計コード単位の平均会計額を表示したい。次のSQL文中のcに入る正しい答えを、解答群の中から選べ。ここで、売上金額は会計金額の合計とする。ポイントによる割引サービスを利用した場合は、会計表の会計金額には割引後の金額が設定されている。

```
SELECT 店舗表.店舗コード, 店舗表.店舗名,  
       c  
FROM 店舗表, 会計表  
WHERE 店舗表.店舗コード = 会計表.店舗コード AND  
      会計表.来店日 BETWEEN '20121201' AND '20121231'  
GROUP BY 店舗表.店舗コード, 店舗表.店舗名
```

### 解答群

- ア SUM(ALL 会計表.会計金額) AS 売上金額,  
MAX(会計表.会員番号) AS 延べ来客数,  
AVG(売上金額 / 延べ来客数) AS 平均会計額
- イ SUM(ALL 会計表.会計金額) AS 売上金額,  
MAX(会計表.会計コード) AS 延べ来客数,  
AVG(会計表.会計金額) AS 平均会計額
- ウ SUM(会計表.会計金額) AS 売上金額,  
COUNT(DISTINCT 会計表.会計コード) AS 延べ来客数,  
AVG(売上金額 / 延べ来客数) AS 平均会計額
- エ SUM(会計表.会計金額) AS 売上金額,  
COUNT(会計表.会計コード) AS 延べ来客数,  
AVG(会計表.会計金額) AS 平均会計額

設問3 会員の来店を促す目的で、次のSQL文によって表示される会員に対して、割引券を送付することにした。表示される会員の説明として適切な答えを、解答群の中から選べ。ここで、会員は、1度は来店したことがあるものとする。

```
SELECT 会員表.会員番号, 会員表.氏名
      FROM 会員表, 会計表
     WHERE 会員表.会員番号 = 会計表.会員番号
       GROUP BY 会員表.会員番号, 会員表.氏名
      HAVING MAX(会計表.来店日) < '20130101'
```

### 解答群

- ア 2013年になってから、1回以上、来店した会員
- イ 2013年になってから、1回も来店していない会員
- ウ 2013年より前に、1回以上、来店した会員
- エ 2013年より前に、1回も来店していない会員

設問4 顧客サービスを向上させるために，“指名美容師”と“過去の来店時に担当した美容師”を確認できるよう、会員表と会計表に担当美容師の項目を追加した。担当美容師は、図3に示す従業員表の従業員コードを外部キーとして参照する。指名美容師がいない会員の場合、会員表の担当美容師としてNULLを設定する。情報の追加後、従業員コード“2008005”の美容師が所属する店舗が変わるので、会員表の担当美容師に登録している会員と、当該美容師が過去に担当したことがある会員に、所属店舗移動の案内を送付することにした。該当する会員の会員番号を表示する次のSQL文中の [ ] に入る正しい答えを、解答群の中から選べ。

従業員表

従業員コード	氏名	住所	電話番号	生年月日	店舗コード
2008005	美容太郎	東京都文京区青葉3丁目	03-1234-5678	19850105	02

図3 従業員表とデータの格納例

```
SELECT DISTINCT 会員表.会員番号, 会員表.氏名, 会員表.住所
FROM 会員表, 会計表
WHERE [ ] d
```

解答群

- ア 会員表.会員番号 = 会計表.会員番号 AND  
(会員表.担当美容師 = '2008005' OR  
会計表.担当美容師 = '2008005')
- イ 会員表.会員番号 = 会計表.会員番号 AND  
会員表.担当美容師 = '2008005' AND  
会計表.担当美容師 = '2008005' AND  
会計表.店舗コード = (SELECT 店舗コード FROM 従業員表  
WHERE 従業員コード = '2008005')
- ウ 会員表.会員番号 = 会計表.会員番号 AND  
会計表.店舗コード = (SELECT 店舗コード FROM 従業員表  
WHERE 従業員コード = '2008005')
- エ 会員表.会員番号 = 会計表.会員番号 AND  
会計表.店舗コード = ANY (SELECT 店舗コード FROM 従業員表  
WHERE 従業員コード = '2008005')

問4 ICカードを利用した入退室管理システムに関する次の記述を読んで、設問1～5に答えよ。

J社は、中規模のSIベンダーであり、外部の協力が必要なシステム開発のときには、プロジェクトごとに協力会社と契約している。J社には、開発室と執務室があり、開発室には執務室を通って入退室する。各室の出入口の内側と外側にICカード読取り装置が設置されており、社員と、協力会社社員（以下、協力社員という）の入退室は、入退室管理システムで管理されている。社員及び協力社員は入退室時に、ICカードを読み取り装置にかざし、入室時には更にパスワードを入力することによって、出入口の扉が開錠される。また、扉が閉められると、自動的に施錠される。

J社の入退室管理システムのセキュリティ要件は、次のとおりである。

〔J社の入退室管理システムのセキュリティ要件〕

- (1) 社員及び協力社員は、プロジェクトに参画している期間中だけ開発室に入室可能とする。
- (2) ICカードには、①耐タンパ性をもつものを使用し、ICカードIDだけを情報としてもつ。
- (3) ②入退室管理システムは入退室のログを収集する。
- (4) 入退室のログから、開発室又は執務室への入退室ごとの出入りした社員又は協力社員、日時、出入口が特定できる。
- (5) パスワードは8桁の数字（00000000～99999999）とする。
- (6) 有効期間中は、ICカードとパスワードによって開発室や執務室への入室ができる。
- (7) 入室時又はパスワードの変更時に、3回連続してパスワードを誤って入力した場合、開発室や執務室への入室はできなくなる。

なお、J社では、社員や協力社員が、同時に複数のプロジェクトに参画することはない。

[入退室管理システムの説明]

入退室管理システムが管理する、利用者情報のうち主なものを表1に、入退室情報のうち主なものを表2に示す。

表1 主な利用者情報

利用者情報	説明
利用者ID	社員の場合は社員番号を設定し、協力社員の場合は契約時に個人ごとに付与される契約番号を設定する。
ICカードID	ICカードを識別する一意のID
ICカードの状態	“仮パスワード”，“有効”，“返却”，“一時利用停止”的いずれかである。ICカードを発給したときは、“仮パスワード”を設定する。3回連続してパスワードを誤って入力した場合，“一時利用停止”になる。
入室許可の状態	“開発室許可”，“執務室だけ許可”，“入室不可”的いずれかである。
有効期間の終了日	社員の場合は、退職予定の年月日を設定しておく。協力社員の場合は契約期間に基づいて契約終了予定の年月日を設定しておく。
(上記以外の利用者情報)	
氏名、有効期間の開始日、利用者区分、プロジェクト番号、パスワードなど	

表2 主な入退室情報

入退室情報	説明
ICカード利用日時	出入口でICカードをかざした年月日時分秒
ICカード読み取り装置識別番号	出入口に設置しているICカード読み取り装置を識別する一意の番号
ICカードID	出入口でかざしたICカードのID

[入退室管理システムの運用の説明]

セキュリティ管理者は、入室申請の受付、入退室管理システムへの利用者情報の設定、ICカードの発給を担当する。

(1) 社員に対する運用は、次のとおりである。

- (a) 社員の入社時に、入退室管理システムの運用ルールを説明した後、ICカードを発給し、パスワードを仮パスワードから変更させる。これで社員の執務室への入室が可能となる。
- (b) プロジェクトの開始時及び終了時に、プロジェクトマネージャ（以下、PMという）からの申請を受けて、開発室へのプロジェクトメンバの“入室許可の状態”的設定を変更する。

- (c) 退職時には、IC カードを返却させるとともに、“有効期間の終了日”に退職日を、“IC カードの状態”に“返却”を設定する。
- (2) 協力社員に対する運用は、次のとおりである。
- プロジェクトの開始時に、PM からの申請を受けて、当該協力社員の利用者情報を登録すると同時に“入室許可の状態”を設定し、PM に協力社員用の IC カードを発給する。IC カードを受領した PM は入退室管理システムの運用ルールを協力社員に説明した後、IC カードを配布してパスワードを仮パスワードから変更させる。
  - 契約の終了時は、協力社員に配布していた IC カードの返却を PM 経由で受け、 “有効期間の終了日”に契約の終了日を、“IC カードの状態”に“返却”を設定する。
- (3) 利用者情報の削除処理は、次のとおりである。
- “有効期間の終了日”を過ぎ、かつ、“IC カードの状態”が“返却”的利用者情報は、週末のバッチ処理でバックアップメディアに保存した上で、入退室管理システムから削除する。
  - 返却された IC カードは、後日再利用する。

入退室管理システムで管理する、社員を対象にした“IC カードの状態”，“入室許可の状態”及び“入室可能な部屋”的関係を表す状態遷移図を、図 1 に示す。

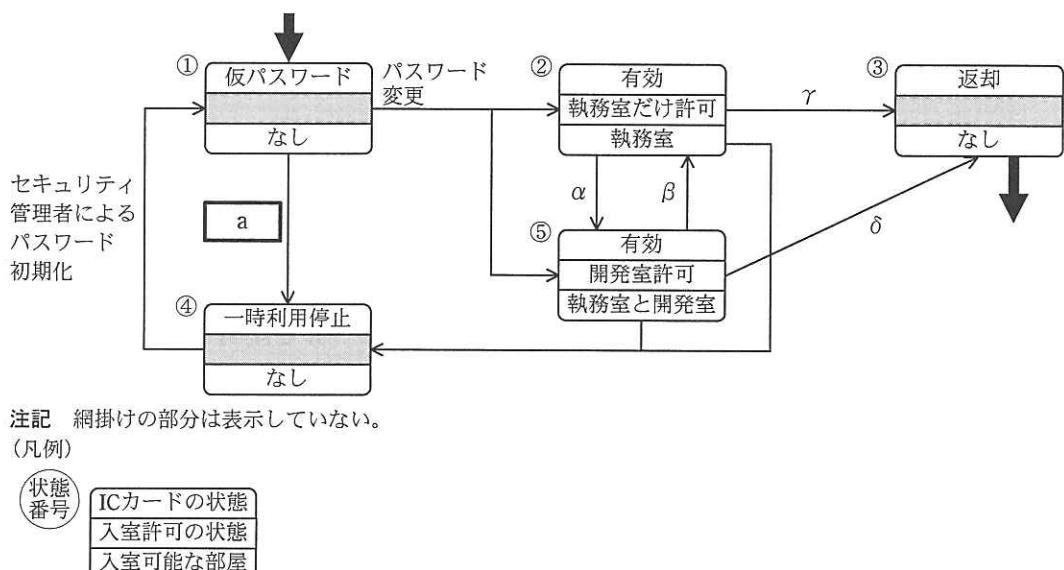


図 1 社員を対象とした状態遷移図

設問 1 〔J 社の入退室管理システムのセキュリティ要件〕の説明中の下線①の IC カードの説明として正しい答えを、解答群の中から選べ。

解答群

- ア 一部が破損しても利用できる IC カード
- イ 外部から強い衝撃があっても変形しない IC カード
- ウ 内部情報に外部から不正にアクセスできない IC カード
- エ 返却後に、再利用できる IC カード

設問 2 〔J 社の入退室管理システムのセキュリティ要件〕の説明中の下線②のログとして収集するのが適切な情報を、解答群の中から選べ。

解答群

- ア IC カード読み取り装置識別番号、IC カード ID、利用者 ID
- イ IC カード利用日時、IC カード ID、利用者 ID
- ウ IC カード利用日時、IC カード読み取り装置識別番号、IC カード ID、利用者 ID
- エ IC カード利用日時、IC カード読み取り装置識別番号、入室許可の状態

設問 3 図 1 の   に入る正しい答えを、解答群の中から選べ。

解答群

- ア 3 回連続してパスワードを誤入力
- イ 社員の入社
- ウ 入退室管理システムの異常
- エ プロジェクトの終了

**設問4** 図1を基に、最少の変更で、協力社員を対象にした状態遷移図を作成するとした場合、協力社員が契約を終了して遷移する矢印として適切な答えを、解答群の中から選べ。

解答群

ア  $\alpha$

イ  $\beta$

ウ  $\gamma$

エ  $\delta$

**設問5** J社では内部監査時に、開発室及び執務室の入退室に関する調査を行ったところ、入退室管理システムのログに、入室履歴のない退室履歴や退室履歴のない入室履歴が見つかった。そこで、更に調査した結果、直前に入退室した者がいるとき、扉が施錠される前に、自分のICカードを使わずに入退室する者がいることが分かった。そこで、入退室時には自分のICカードを必ず読み取り装置にかざせる対策として、教育を実施するとともに入退室管理システムで管理する現在の状態遷移を変更することにした。図1の状態番号のうち、入室履歴又は退室履歴のない者がICカードをかざして退室又は入室しようとした際に、遷移する先として適切な状態番号を、解答群の中から選べ。

解答群

ア ①

イ ②

ウ ③

エ ④

オ ⑤

問5 社員の歩合給決定処理に関する次の記述を読んで、設問1～3に答えよ。

生命保険会社S社の社員の毎月の給与は、基本給と歩合給とから成る。基本給は、社員の等級に応じて固定であり、歩合給は変動する。歩合給は、社員が担当する生命保険の前月末日時点の担当契約数に対して5件ごとに10,000円、担当契約からの月払保険料の合計が50,000円ごとに5,000円である。担当契約には、終了した保険契約は含まれない。例えば、担当契約数が18件、月払保険料の合計が120,000円の場合、歩合給は40,000円となる。

〔契約マスタファイルの説明〕

保険契約の情報は、10桁の数字列から成る証券番号の昇順に並んだ契約マスタファイル（以下、契約マスタという）で管理する。証券番号は、保険契約1件につき一意に付与する。契約マスタのレコード様式は、図1のとおりである。

証券番号	契約年月日	終了年月日	月払保険料	保障額	社員ID	満了年月日
------	-------	-------	-------	-----	------	-------

注記 下線はキー項目を表す。

図1 契約マスタのレコード様式

- (1) 証券番号は、“0000000001”から始まる。
- (2) 契約年月日には、保険契約が有効になる年月日を格納し、終了年月日には、保険契約が終了した年月日を格納する。終了していない保険契約の終了年月日は空白である。

保険契約の終了は、保険契約の解約、満了、死亡時の保険金支払など幾つかの要因で発生する。これら全てを、“終了”として扱う。

- (3) 月払保険料には、契約者が支払う毎月の保険料の金額を格納する。終了した保険契約の月払保険料には0を格納する。
- (4) 保障額には、死亡時に支払う生命保険の保険金の金額を格納する。
- (5) 社員IDには、保険契約を担当する社員のIDを格納する。1件の保険契約は1名で担当し、保険契約の終了まで、担当する社員の変更はないものとする。
- (6) 満了年月日には、保険契約が満了する年月日を格納する。
- (7) 終了した保険契約のレコードは削除しない。

### [異動ファイルの説明]

新たな保険契約の締結、既存の保険契約の終了、及び既存の保険契約の月払保険料・保障額の変動を、異動事由という。異動事由が発生する都度、発生順に異動ファイルに1レコードを作成する。異動事由の発生は、1件の保険契約につき1日1回までである。

異動ファイルは、これまで発生した全ての異動事由を記録したものであり、そのレコード様式は図2のとおりである。

異動年月日	証券番号	異動事由	月払保険料	保障額	社員 ID	満了年月日
-------	------	------	-------	-----	-------	-------

注記 下線はキー項目を表す。

図2 異動ファイルのレコード様式

異動事由には、“新契約”、“増額”、“減額”、“終了”のいずれかを格納する。

その他の項目には、異動事由に応じて契約マスタの更新に必要な値だけを格納する。

異動事由が“新契約”的レコードの証券番号には、最後に発行した証券番号+1が新規に採番される。異動ファイルの内容に誤りはないものとする。

### [契約マスタ更新処理]

毎日、異動ファイル中の前日発生分のレコードと前日の契約マスタ（以下、旧契約マスタという）から、更新した契約マスタ（以下、新契約マスタという）を作成する。契約マスタ更新処理の流れを、図3に示す。

- (1) 異動事由が“新契約”的場合は、異動ファイルの当該レコードの情報から新たにレコードを作成して、新契約マスタに出力する。異動ファイルの当該レコードの異動年月日が、契約年月日となる。
- (2) 異動事由が“増額”又は“減額”的場合は、当該保険契約の月払保険料と保障額を、異動ファイルの当該レコードの情報で更新して、新契約マスタに出力する。
- (3) 異動事由が“終了”的場合は、当該保険契約の月払保険料を0にし、終了年月日を異動年月日に更新して、新契約マスタに出力する。
- (4) 契約マスタ更新処理の実行が日をまたぐことはない。

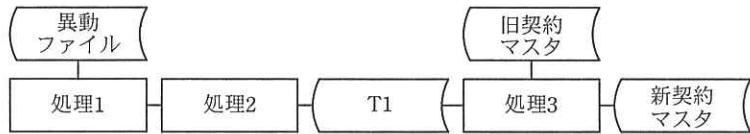


図3 契約マスタ更新処理の流れ

#### 〔歩合給計算処理〕

毎月1日に、契約マスタ更新処理が終了した後、契約マスタから、社員ごとの担当契約数及び月払保険料の合計を求めて当月の歩合給を計算し、歩合給ファイルに社員1人につき1レコードを追加する。

歩合給ファイルのレコード様式を図4に、歩合給計算処理の流れを図5に示す。

支給年月	社員 ID	歩合給額
------	-------	------

注記 下線はキー項目を表す。

図4 歩合給ファイルのレコード様式

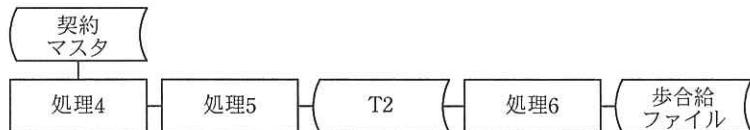


図5 歩合給計算処理の流れ

表1 各処理の説明

処理	番号	内容
契約マスタ 更新処理 (日次)	処理1	異動ファイルから異動年月日が <u>a</u> であるレコードを処理対象として抽出する。
	処理2	処理1で抽出した結果を <u>b</u> の昇順に整列して、中間ファイルT1に出力する。
	処理3	T1と旧契約マスタを、証券番号をキーとして突き合わせ、処理をして、新契約マスタに出力する。この処理をT1と旧契約マスタが最終レコードに達するまで繰り返す。
歩合給計算 処理 (月次)	処理4	契約マスタから <u>c</u> が <u>d</u> であるレコードを処理対象として抽出する。
	処理5	処理4で抽出した結果を社員IDの昇順に整列して、中間ファイルT2に出力する。
	処理6	社員ごとの担当契約数及び月払保険料の合計を求めて歩合給を計算し、歩合給ファイルにレコードを追加する。

設問 1 表 1 に示す各処理の説明の [ ] に入る正しい答えを、解答群の中から選べ。

a, d に関する解答群

- |      |         |       |
|------|---------|-------|
| ア 空白 | イ 終了    | ウ 新契約 |
| エ 前日 | オ 前日より前 | カ 当日  |

b, c に関する解答群

- |         |         |         |
|---------|---------|---------|
| ア 異動事由  | イ 異動年月日 | ウ 契約年月日 |
| エ 社員 ID | オ 終了年月日 | カ 証券番号  |

設問 2 図 6 は、T1 と旧契約マスタを突き合わせて、新契約マスタを作成する処理 3 の流れ図である。図中の [ ] に入る正しい答えを、解答群の中から選べ。

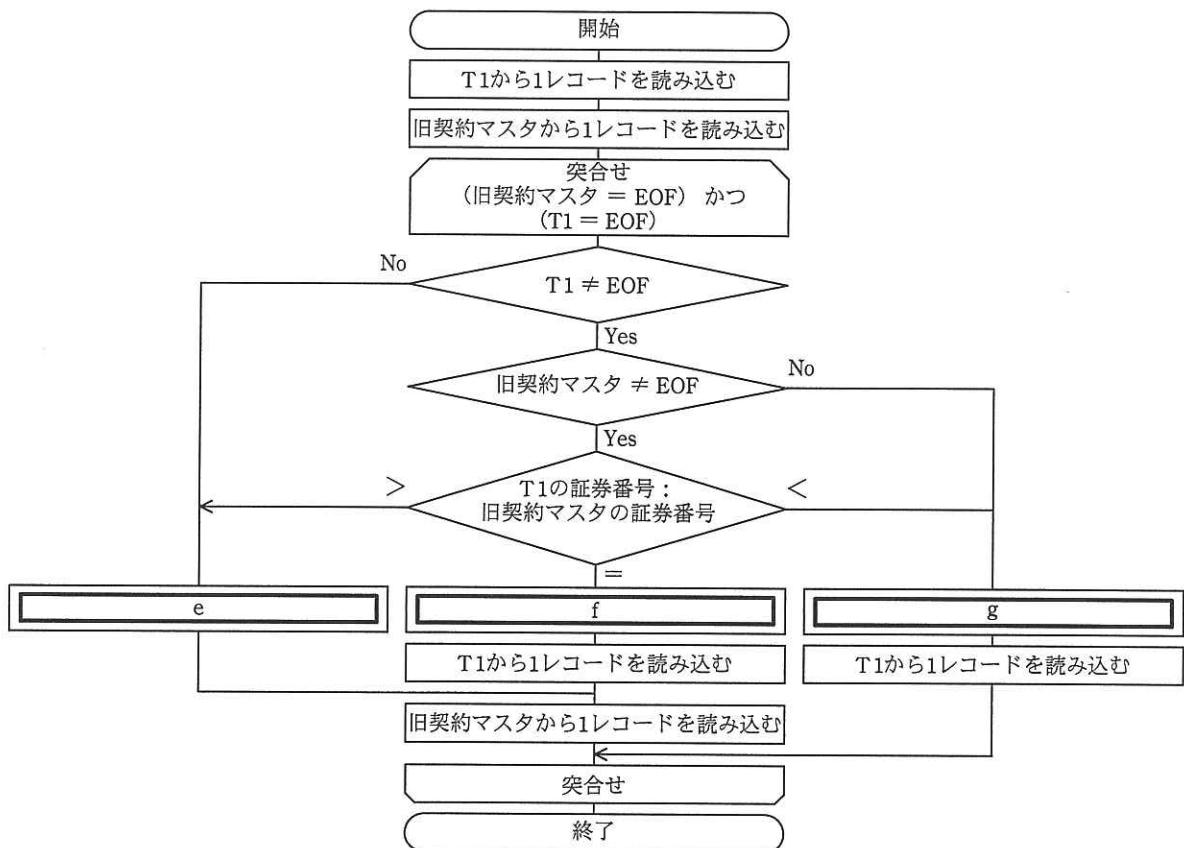


図 6 処理 3 の流れ図

e～g に関する解答群

- ア T1 から 1 レコードを読み込む
- イ T1 のレコードの情報から新たにレコードを作成して新契約マスタに出力する
- ウ 旧契約マスタから 1 レコードを読み込む
- エ 旧契約マスタのレコードの情報を T1 のレコードの情報で更新し、そのレコードを新契約マスタに出力する
- オ 旧契約マスタのレコードをそのまま新契約マスタに出力する
- カ 新契約マスタのレコードをそのまま出力する

**設問 3** 歩合給の計算方法を変更して、新契約を成立させた月が、過去 3 か月間連続した場合は、当月の歩合給を 2 割増にする。過去 3 か月間連続して新契約を成立した社員の歩合給が 0 円である場合は、5,000 円を歩合給として支給する。

表 2 は、2013 年 5 月 1 日時点の契約マスタからの、社員 A00001 と社員 B00001 のレコードの抜粋である。また、表 3 は、2013 年 4 月 30 日の異動ファイルからの、社員 A00001 と社員 B00001 のレコードの抜粋である。2013 年 5 月 1 日に、契約マスタを更新し、変更後の歩合給計算処理を実行したときの社員 A00001 と社員 B00001 の歩合給の組合せとして、正しい答えを、解答群の中から選べ。

表 2 2013 年 5 月 1 日時点の契約マスタ（抜粋）

証券番号	契約年月日	終了年月日	月払保険料	保障額	社員 ID	満了年月日
0004157001	2012/12/01	2013/03/31	0	1,000,000	B00001	2022/07/31
0004157002	2013/01/04		25,000	6,000,000	B00001	2022/09/03
0004157003	2013/01/15		45,000	10,000,000	A00001	2022/09/14
0004157004	2013/02/01		8,000	2,000,000	A00001	2022/09/30
0004157005	2013/02/15		15,000	3,000,000	B00001	2022/10/14
0004157006	2013/02/24		24,000	5,500,000	A00001	2022/10/23
0004157007	2013/03/01		42,000	8,000,000	A00001	2022/10/31
0004157008	2013/03/15		3,000	1,000,000	B00001	2022/11/14
0004157009	2013/03/29		30,000	7,000,000	A00001	2022/11/28

表3 2013年4月30日の異動ファイル（抜粋）

異動年月日	証券番号	異動事由	月払保険料	保障額	社員 ID	満了年月日
2013/04/30	0004157010	新契約	30,000	7,000,000	A00001	2022/12/29
2013/04/30	0004157011	新契約	20,000	4,000,000	A00001	2022/12/29
2013/04/30	0004157012	新契約	10,000	2,500,000	B00001	2022/12/29
2013/04/30	0004157006	減額	22,000	5,000,000		
2013/04/30	0004157007	増額	50,000	12,000,000		
2013/04/30	0004157009	増額	40,000	8,000,000		
2013/04/30	0004157002	終了	0			
2013/04/30	0004157003	終了	0			

解答群

	社員 A00001	社員 B00001
ア	25,000 円	0 円
イ	25,000 円	5,000 円
ウ	25,000 円	6,000 円
エ	30,000 円	0 円
オ	30,000 円	5,000 円
カ	30,000 円	6,000 円

問6 ソフトウェア開発の品質管理に関する次の記述を読んで、設問1、2に答えよ。

システム開発プロジェクトXでは、四つの機能で構成される新規ソフトウェアを4チーム（P～Sチーム）が分担して開発している。このプロジェクトでは、内部設計工程及びプログラミング工程で品質を確保し、できるだけ単体テストを含むテスト工程に欠陥を持ち越さないよう、品質管理及び欠陥の摘出に努めている。

[各工程での品質管理の説明]

- (1) 内部設計工程では、内部設計書の設計レビューを行う。設計レビューでは、設計担当者を含めたチーム内のメンバ3名以上によるチームレビューを行う。
- (2) プログラミング工程では、ソースプログラムのコードレビューを行う。コードレビューには、セルフレビュー及びペアレビューがある。プログラミング担当者が単独で行うのがセルフレビューであり、プログラミング担当者ともう1名でペアを組んで行うのがペアレビューである。セルフレビューの終了後にペアレビューを行う。
- (3) 各工程における品質管理指標の基準値及び許容範囲は、表1及び表2のとおりに設定している。基準値は、レビューの品質を判定する際の基準であり、各機能のソースプログラム規模1kステップ当たりのレビュー時間又は摘出欠陥数を設定している。許容範囲は合格の範囲を示している。これらの値は、プロジェクトメンバーの過去の類似プロジェクトでの実績データを基に設定している。  
なお、ソースプログラム規模は、内部設計工程では開始時点での見積りステップ数であり、プログラミング工程ではコーディング終了時点の実ステップ数である。

表 1 内部設計工程での品質管理指標の計画値

品質管理指標	基準値	許容範囲
設計レビュー時間	3.0 時間／k ステップ	基準値の 1.0～1.2 倍
抽出欠陥数	4.0 件／k ステップ	基準値の 0.9～1.1 倍

表 2 プログラミング工程での品質管理指標の計画値

品質管理指標	基準値	許容範囲
コードレビュー での抽出欠陥数	6.0 件／k ステップ	基準値の 0.9～1.1 倍

- (4) 表 2 で示した値はプログラミング工程の全てのコードレビューが終了した時点での品質管理指標であり、コードレビューの内訳として、セルフレビューが終了した時点での抽出欠陥数の許容範囲は、基準値の 0.4～0.6 倍である。
- (5) 各工程では、品質管理指標の計画値と実績値との差を調べてレビューの品質を評価する。品質管理指標の実績値が全て許容範囲内ならば、その工程での品質は合格の水準に達しているとし、次の工程へ進む。実績値が許容範囲を外れている場合は、レビュー内容などを基に実績値の適切さをプロジェクトリーダが評価し、適切であると判定した場合には合格として次の工程へ進む。適切でないと判定した場合には不合格とみなし、品質確保への改善策を講じる。

設問 1 内部設計工程での品質管理指標の計画値と実績値との差異分析に関する次の記述中の [ ] に入る適切な答えを、解答群の中から選べ。

各チームが分担する総規模及び内部設計工程を終了した時点での品質管理指標の実績値は、表 3 のとおりである。

なお、表中の分担総規模は内部設計工程の開始時点に見積もったソースプログラムのステップ数であり、内部設計期間中にこの値は変わらない。

表 3 各チームの分担総規模及び内部設計工程終了時点の品質管理指標の実績値

チーム	分担総規模 (k ステップ)	設計レビュー時間 (時間)	摘出欠陥数 (件)
P	40	112	168
Q	25	88	88
R	20	50	70
S	15	45	60

プロジェクトリーダは、設計レビュー時間及び摘出欠陥数の実績値がともに許容範囲内である [ a ] チームの設計レビューを合格とした。残りのチームについては、実績値の適切さを評価して、設計レビュー時間及び摘出欠陥数がともに許容範囲を外れている [ b ] チームを不合格と判定した。

a, b に関する解答群

ア P

イ Q

ウ R

エ S

オ P と Q

カ P と R

キ Q と R

ク Q と S

設問2 あるチームのメンバであるYさんは、週ごとにコーディングしたソースプログラムのセルフレビューを、その週内で行っている。Yさんのプログラミング工程での作業開始から4週間の週ごとのセルフレビューの実績値は表4のとおりであった。Yさんが担当するプログラムの品質評価に関する次の記述中の  
[ ]に入る適切な答えを、解答群の中から選べ。

表4 Yさんのプログラミング工程でのセルフレビューの実績値

週	第1週	第2週	第3週	第4週	合計
レビュー規模 (kステップ)	2.5	3.5	2.0	1.0	9.0
抽出欠陥数 (件数)	8	11	8	2	29

まず、Yさんのセルフレビューでの抽出欠陥数の4週間の合計は、[c]。次に、週ごとの1kステップ当たりの抽出欠陥数を計算して、許容範囲による判定及び週ごとの変化を評価した。週ごとの1kステップ当たりの抽出欠陥数は、  
[d]。

プロジェクトリーダは、セルフレビューの内容を評価した結果、セルフレビューを終了し、ペアレビューを実施するように指示した。

なお、セルフレビューでの許容範囲を外れた週に関して、ペアレビューで抽出すべき欠陥数の目標値は次のように求める。

- (1) セルフレビューでの許容範囲を上回っていた週は、コードレビューでの許容範囲の上限を上回る最小の整数値から、セルフレビューでの抽出欠陥数の実績値を減じた値とする。ただし、セルフレビューでの抽出欠陥数の実績値はコードレビューでの許容範囲の上限を上回る最小の整数値を超えないものとする。
- (2) セルフレビューでの許容範囲を下回っていた週は、コードレビューでの許容範囲の下限を上回る最小の整数値から、セルフレビューでの抽出欠陥数の実績値を減じた値とする。

したがって、セルフレビューでの許容範囲から外れていた週のソースプログラム群に対して、ペアレビューで抽出すべき欠陥数の目標値は、合計で  
[e]件である。

c に関する解答群

- ア 許容範囲内にある
- イ 許容範囲の下限に満たない
- ウ 許容範囲の上限を超えている

d に関する解答群

- ア 週が進むにつれて値が段々と上がっている
- イ 週が進むにつれて値が段々と下がっている
- ウ 週によって許容範囲を上回ったり下回ったりムラがある
- エ どの週も許容範囲を上回ることはないが下回る週がある
- オ どの週も許容範囲を下回ることはないが上回る週がある

e に関する解答群

ア 7 イ 8 ウ 9 エ 10 オ 11

問 7 市場分析と需要予測に関する次の記述を読んで、設問 1～3 に答えよ。

清涼飲料メーカーの Z 社は、海外の 5 か国（A 国、B 国、C 国、D 国及び E 国）への進出を検討している。Z 社の企画課では、各国の清涼飲料市場の分析を行うことにした。

設問 1 図 1 は、2002～2011 年（以下、対象期間という）の各国における清涼飲料の年間販売数量の推移である。図 1 に関する記述の中で適切なものを解答群の中から三つ選べ。ここで、対象期間販売数量伸び率とは、次の式で求められる値とする。

$$2011 \text{ 年の年間販売数量} \div 2002 \text{ 年の年間販売数量}$$

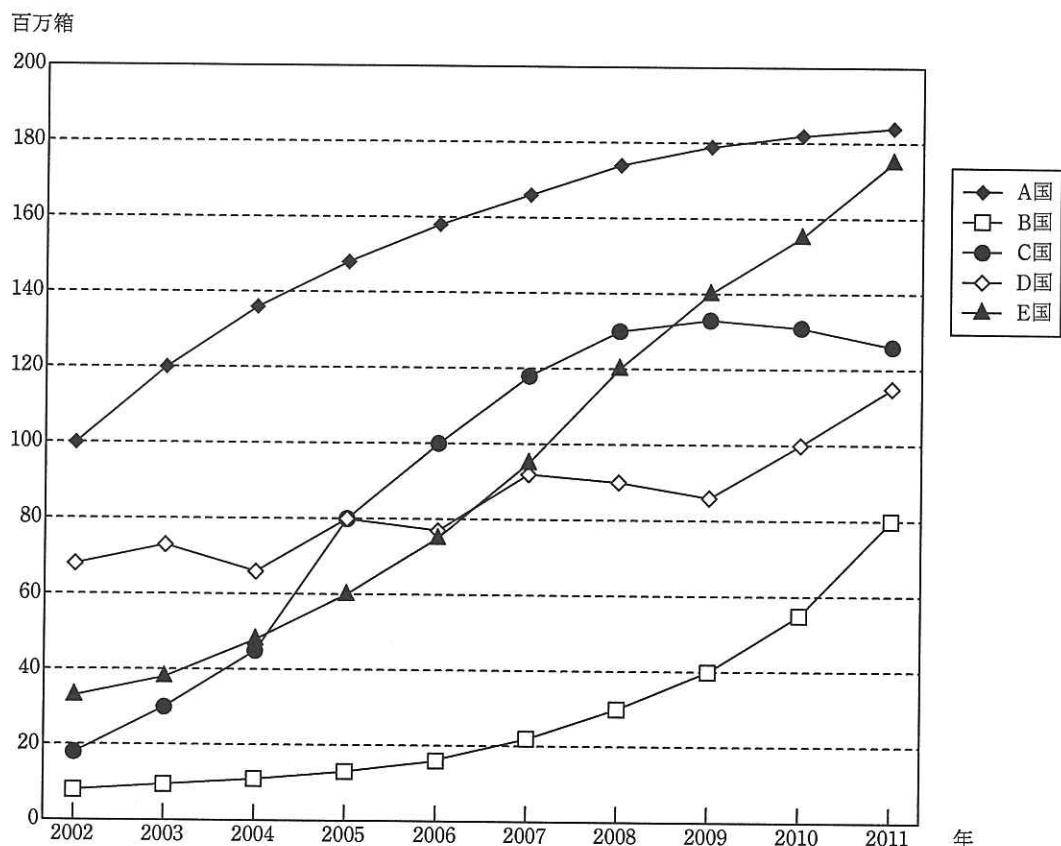


図 1 2002～2011 年の各国における清涼飲料の年間販売数量の推移

## 解答群

- ア 5か国合計の年間販売数量は、対象期間中において毎年増加している。
- イ 5か国の年間販売数量の順位に変動があったのは、2005年と2009年だけである。
- ウ A国は5か国の中で対象期間中の販売数量が最も多く、対象期間販売数量伸び率が最も高い。
- エ B国は5か国の中で対象期間中の販売数量は最も少ないが、対象期間販売数量伸び率は最も高い。
- オ C国は5か国中で唯一、2011年の年間販売数量が2002年よりも少ない。
- カ D国は2008年から2011年までの年間販売数量は毎年増加しているが、対象期間販売数量伸び率は5か国の中で最も低い。
- キ E国は2002年に対する2011年の年間販売数量の増加量が5か国の中で最も多い。
- ク 年間販売数量が対象期間中において毎年増加しているのは2か国である。

設問2 企画課では、各国の清涼飲料の年間売上金額についてのデータを収集した。ここで、年間売上金額は、年単位に更新される為替レートで換算された米ドルのデータしか入手できなかった。表1はその抜粋であり、2010年と2011年のC国における清涼飲料の年間販売数量と年間売上金額である。表1に関する次の記述中の [ ] に入る適切な答えを、解答群の中から選べ。

表1 2010年と2011年のC国における年間販売数量と年間売上金額

	2010年	2011年
年間販売数量（百万箱）	131	126
年間売上金額（億米ドル）	10.4	11.3

表1について、企画課では、2011年が2010年に比較して年間販売数量が減少しているものの年間売上金額は増加していること、すなわち平均の商品単価（年間売上金額 ÷ 年間販売数量）が上がっていることに着目した。企画課はこれらの原因として、インフレや増税などに起因する [a]、C国内の経済成長の結果としての所得増により [b]、若しくは為替レートに関係した [c] のいずれか一つ、又はその組合せと考えた。これらの動きは、C国の将来の市場成長に大きな影響を与える可能性があるので、企画課ではその要因を調査すること

にした。

aに関する解答群

ア 商品の種類の減少

イ 商品の種類の増加

ウ 商品の値上げ

エ 商品の値下げ

bに関する解答群

ア 高価格帯商品へ購入がシフト

イ 低価格帯商品へ購入がシフト

ウ 商品の購入頻度が減少

エ 商品の購入頻度が増加

オ 商品の購入量が減少

カ 商品の購入量が増加

cに関する解答群

ア 日本円に対する現地通貨高

イ 日本円に対する現地通貨安

ウ 日本円に対する米ドル高

エ 日本円に対する米ドル安

オ 米ドルに対する現地通貨高

カ 米ドルに対する現地通貨安

設問3 企画課では、各国の将来の需要予測を行うことにした。需要予測に関する次の記述中の    に入る正しい答えを、解答群の中から選べ。

企画課では、各国の将来の清涼飲料の需要予測式を、年間販売数量を目的変数、1人当たりGDPと人口を説明変数として、重回帰分析を使って導出した。その結果、為替レートの変動が少ないE国に関する需要予測式は、次のとおりであった。

$$\begin{aligned} \text{E国の年間販売数量 (百万箱)} &= 16 \times \text{E国の1人当たりGDP (千米ドル)} \\ &\quad + 35 \times \text{E国の人団 (百万人)} - 872 \end{aligned}$$

表2は、企画課で推定したE国の将来の1人当たりGDP、人口及び清涼飲料の1箱当たりの平均単価の予測である。

表2 E国将来の1人当たりGDP、人口及び清涼飲料の1箱当たりの平均単価の予測

	2011年	2015年	2020年
1人当たりGDP(千米ドル)	6.5	7.0	7.2
人口(百万人)	27.1	27.0	26.8
1箱当たり平均単価(米ドル)	10	12	13

企画課では、需要予測式と表2から、E国清涼飲料の年間販売数量は、

dと予測した。また、E国清涼飲料の年間売上金額(米ドル基準)は、

eと予測した。

d, eに関する解答群

- ア 2011年に対して2015年が、2015年に対して2020年がともに減少する
- イ 2011年に対して2015年が、2015年に対して2020年がともに増加する
- ウ 2011年に対して2015年は減少するが、2015年に対して2020年は増加する
- エ 2011年に対して2015年は増加するが、2015年に対して2020年は減少する

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。

ある食料品店では、従来の特売方法に加えて、新たに選択型特売を始めることになった。選択型特売とは、例えば3種類の商品中から合計3個（3商品を各1個、同一商品を3個など）を選択して購入すれば値引きをするという特売方法である。

このプログラムは、レジ用プログラムの一部として、選択型特売の値引き処理をする。プログラムの実行は、顧客がレジに持ち込んだ商品のバーコードを全て読み込んで、購入する商品が確定した時点で行うものとする。

#### 〔プログラムの説明〕

(1) 顧客がレジに持ち込んだ商品の情報（以下、購入情報という）及び1種類の選択型特売についての商品の情報（以下、特売情報という）は、次のように宣言された大域の変数及び配列で他のレジ用プログラムと受け渡す。配列の添字は、1から始まる。

```
/* 購入情報 */
○大域 整数型： ptr 起点、 購入行数
○大域 構造型： 購入[] {整数型： ptr, 整数型： 品番, 文字型： 品名,
                        整数型： 単価, 整数型： 数量, 整数型： 金額}

/* 特売情報 */
○大域 整数型： 指定数量, 対象行数
○大域 構造型： 対象[] {整数型： 品番, 整数型： 数量},
                        特売    {整数型： 品番, 文字型： 品名,
                                整数型： 単価, 整数型： 数量}
```

(2) 購入情報の内容は、次のとおりである。図1は、購入情報のデータ例である。

- ① 購入[]は、ptr、品番、品名、単価、数量、金額のメンバから成る1レコードを1要素とする構造型の配列である。
- ② 購入[]には、購入する商品の情報が、バーコードを読み込んだ順に添字1の要素から格納されている。同一品番の商品の情報は、1レコードにまとめられており、品番の重複はない。
- ③ 購入[]中のptrは、レコードを品番の昇順にたどるポインタであり、次に大きい品番をもつレコードが格納されている要素の添字が入っている。最も大きい品番をもつレコードのptr値は0である。

- ④ **ptr 起点** は、最も小さい品番をもつレコードが格納されている **購入[]** の要素の添字を示す。
- ⑤ **購入行数** は、**購入[]** 中に格納されたレコード件数を示す。
- (3) 特売情報の内容は、次のとおりである。
- ① 特売は、品番、品名、単価、数量のメンバから成る構造型の変数である。品番、品名、単価には、特売を一つの商品とみなして付けた品番、特売の名称、値引き額が、それぞれ格納されている。
  - ② **対象[]** は、品番、数量のメンバから成る 1 レコードを 1 要素とする構造型の配列である。品番には、特売対象の商品の品番が格納されている。**対象[]** には、要素が品番の昇順に添字 1 から格納されている。
  - ③ **対象行数** は、**対象[]** 中に格納されたレコード件数を示す。
  - ④ **指定数量** は、特売対象の商品を合計何個購入する必要があるかを示す。
  - ⑤ **対象[]** 中及び**特売**中の**数量**には、初期値 0 が格納されている。
  - ⑥ 図 2 には、特売情報のデータ例として、次の特売例を示してある。

例 B 社緑茶 500 ml (品番 222), B 社ほうじ茶 500 ml (品番 223), B 社麦茶 500 ml (品番 224) のどれでも、合わせて 3 本の買上げごとに 100 円引きとする。

		ptr 品番 品名		単価	数量	金額	
ptr 起点	購入[1]	4	222	B 社緑茶 500 ml	140	5	700
購入行数	購入[2]	1	111	A 社牛乳 1000 ml	200	2	400
5	購入[3]	0	335	C 社めんつゆ 300 g	150	1	150
5	購入[4]	5	224	B 社麦茶 500 ml	140	2	280
5	購入[5]	3	333	C 社うどん 2 食入	180	2	360
	:						

図 1 購入情報のデータ例

		品番 数量	
指定数量	対象[1]	222	0
3	対象[2]	223	0
対象行数	対象[3]	224	0
3	:		

		品番 品名		単価	数量
特売		229	B 社お茶 3 本 100 円引	-100	0

図 2 特売情報のデータ例

- (4) 処理は、検索部、計算部、更新部から成り、この順に実行する。
- ① 検索部では、対象[]中の各品番が購入[]中にあれば、購入[]中の該当するレコードの数量の値を、対象[]中の数量に格納する。
  - ② 計算部では、対象[]中の各数量の値の合計を指定数量で割った商を、特売中の数量に格納する。
  - ③ 更新部では、特売中の数量の値が1以上であれば、購入[]に特売のレコードを追加し、購入情報を更新する。
- (5) メンバの参照は、例えば、購入[3]の品番を参照するときは、購入[3].品番と記述する。
- (6) 図1及び図2のデータ例を用いてプログラムを実行した結果は、図3及び図4のとおりである。図3及び図4中の網掛け部分は、実行によって内容が変更された箇所であることを示す。

		ptr	品番	品名	単価	数量	金額
ptr 起点	購入[1]	4	222	B社緑茶 500ml	140	5	700
2	購入[2]	1	111	A社牛乳 1000ml	200	2	400
購入行数	購入[3]	0	335	C社めんつゆ 300g	150	1	150
6	購入[4]	6	224	B社麦茶 500ml	140	2	280
	購入[5]	3	333	C社うどん 2食入	180	2	360
	購入[6]	5	229	B社お茶 3本 100円引	-100	2	-200
:							

図3 図1の購入情報のデータ例（実行後）

		品番	数量
指定数量	対象[1]	222	5
3	対象[2]	223	0
対象行数	対象[3]	224	2
3	:		

		品番	品名	単価	数量
特売		229	B社お茶 3本 100円引	-100	2

図4 図2の特売情報のデータ例（実行後）

[プログラム]

○整数型： K, Kp, T, W

/\* 検索部 \*/

- K  $\leftarrow$  ptr 起点
- T  $\leftarrow$  1

■ (K > 0) and (T  $\leq$  対象行数)

▲ 購入[K].品番 = 対象[T].品番

- 対象[T].数量  $\leftarrow$  購入[K].数量

• 

a
---

• 

b
---

▲ 購入[K].品番 < 対象[T].品番

• 

a
---

• 

b
---

/\* 計算部 \*/

- W  $\leftarrow$  0
- T: 1, T  $\leq$  対象行数, 1
  - W  $\leftarrow$  W + 対象[T].数量
- 特売.数量  $\leftarrow$  W  $\div$  指定数量

/\* 更新部 \*/

- Kp  $\leftarrow$  0
- K  $\leftarrow$  ptr 起点
- (K > 0) and (購入[K].品番 < 特売.品番)
  - Kp  $\leftarrow$  K
  - K  $\leftarrow$  購入[K].ptr

▲ [特売.数量 > 0]

- 購入行数  $\leftarrow$  購入行数 + 1

▲ Kp > 0

- 購入[Kp].ptr  $\leftarrow$ 

c
---

- ptr 起点  $\leftarrow$ 

c
---

- 購入[購入行数].ptr  $\leftarrow$ 

d
---

- 購入[購入行数].品番  $\leftarrow$  特売.品番

- 購入[購入行数].品名  $\leftarrow$  特売.品名

- 購入[購入行数].単価  $\leftarrow$  特売.単価

- 購入[購入行数].数量  $\leftarrow$  特売.数量

- 購入[購入行数].金額  $\leftarrow$  特売.単価  $\times$  特売.数量

設問 1 プログラム中の [ ] に入る正しい答えを、解答群の中から選べ。

a, b に関する解答群

- |  |                                |
|--|--------------------------------|
| ア $K \leftarrow K + 1$                   | イ $K \leftarrow \text{ptr 起点}$ |
| ウ $K \leftarrow \text{購入}[K].\text{ptr}$ | エ $T \leftarrow 1$             |
| オ $T \leftarrow T + 1$                   |                                |

c, d に関する解答群

- |                                       |                               |
|---------------------------------------|-------------------------------|
| ア $K$                                 | イ $K_p$                       |
| ウ $\text{購入}[K].\text{ptr}$           | エ $\text{購入}[K_p].\text{ptr}$ |
| オ $\text{購入}[\text{購入行数}].\text{ptr}$ | カ $\text{購入行数}$               |

設問 2 次の記述中の [ ] に入る正しい答えを、解答群の中から選べ。

商品のバーコードを読み込む段階で、その時点までに読み込んだ商品の金額の小計を、値引きを反映した金額で表示したい。そこで、商品のバーコードを読み込んだときに選択型特売に該当する商品ならば、都度このプログラムを実行するよう修正したい。また、読み込み済み商品の購入の取消しにも対応したい。ここで、**対象[]** 中及び**特売**中の**数量**には、初期値 0 を格納してプログラム呼び出す。

この方法では、**特売**のレコードを**購入[]**に書き込むとき、既に**購入[]**中にその**特売**のレコードが存在していることがある。また、値引きが成立していた**特売**のレコード中の**数量**が 0 になったとき、その**特売**のレコードの削除が必要になることがある。

**特売**のレコードの取扱いは、表 1 に示す処理に分けられる。

表 1 特売のレコードの取扱い

	特売.数量 = 0	特売.数量 > 0
購入[] 中にその <b>特売</b> のレコードが存在する	処理①	処理②
購入[] 中にその <b>特売</b> のレコードが存在しない	処理③	処理④

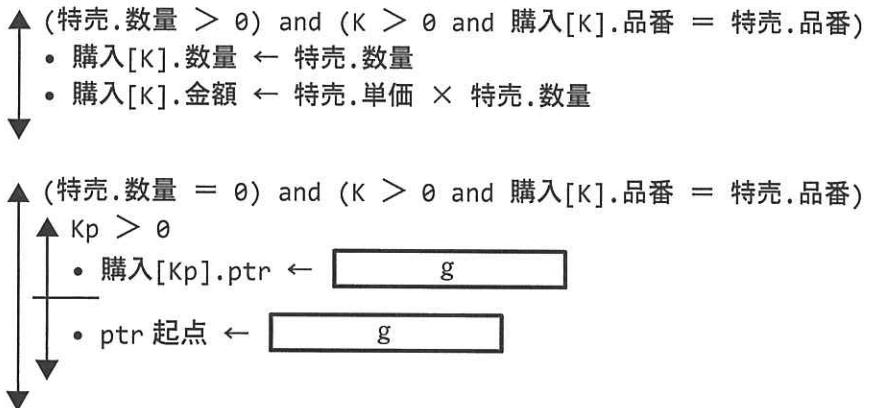
処理 ①～④ のうち、[e] は**特売**のレコードの更新処理であり、  
[f] は**特売**のレコードの削除処理である。

そこで、プログラムの修正方法として、検索部と計算部は変更せず、更新部を次のように変更することにした。

(1) プログラム中の [ ] の部分を、次の条件式で置き換える。

(特売.数量 > 0) and (K = 0 or 購入[K].品番 > 特売.品番)

(2) プログラムの末尾に、次の処理を追加する。



なお、更新処理では、更新前と更新後の特売のレコード中の数量が同じになる場合でも処理を行う。また、削除処理では、該当する特売のレコードを格納していた配列の要素はそのまま残し、ポインタの付替えによってレコードを無効にする。

e, fに関する解答群

ア 処理①

イ 処理②

ウ 処理③

エ 処理④

gに関する解答群

ア K

イ Kp

ウ 購入[K].ptr

エ 購入[Kp].ptr

オ 購入[購入行数].ptr

カ 購入行数

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問以上マークした場合には、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。

C [プログラムの説明]

ケーブルテレビ局を運営するU社では、有線テレビ視聴サービスとインターネット接続サービスを提供している。

- (1) 有線テレビ視聴サービスでは、有線テレビ基本視聴契約（以下、基本視聴契約という）を結ぶと視聴できる基本放送チャンネル、及び基本視聴契約に加えて有料放送視聴契約を結ぶと視聴できる有料放送チャンネルを提供している。
- (2) 基本視聴契約にはプラン1、プラン2の2種類があり、プラン2を選ぶと毎月の基本視聴料金はプラン1よりも高くなるが、有料放送視聴料金は割安になる。毎月の基本視聴料金は、表1のとおりである。

表1 基本視聴料金表（月額）

基本視聴契約	基本視聴料金
プラン1	3,000円
プラン2	5,000円

- (3) 每月の有料放送視聴料金は、基本視聴契約のプランと視聴契約する有料放送のチャンネル数によって決まる。
- (4) 基本視聴料金と有料放送視聴料金の合計を、有線テレビ視聴料金という。
- (5) インターネット接続サービスは、インターネット接続契約を結ぶと利用できる。インターネット接続サービスでは、低速回線と高速回線の2種類を提供している。毎月のインターネット接続サービス利用料金は、基本視聴契約の有無、プランの種類によって変わってくる。毎月のインターネット接続サービス利用料金は、表2のとおりである。

表2 インターネット接続サービス利用料金表（月額）

		インターネット接続契約	
		低速回線	高速回線
基本視聴契約	なし	3,000 円	6,000 円
	プラン 1	2,500 円	6,000 円
	プラン 2	2,500 円	5,000 円

(6) 毎月の利用料金は、有線テレビ視聴料金とインターネット接続サービス利用料金の合計である。

(7) 関数 `calc_service_fee` は、毎月の利用料金を求めるプログラムである。引数及び返却値は、次のとおりである。引数に誤りはないものとする。

引数：  
`basic_plan` 基本視聴契約  
 (0 : なし, 1 : プラン1, 2 : プラン2)  
`channel_num` 視聴契約する有料放送のチャンネル数  
`inet_plan` インターネット接続契約  
 (0 : なし, 1 : 低速回線, 2 : 高速回線)

返却値： 每月の利用料金

### [プログラム]

(行番号)

```

1 #define BLKNUM 4

2         /* 基本視聴料金, 0 は契約なしの場合 */
3 const int basic_charge[] = {0, 3000, 5000},
4         channel_block[BLKNUM] = {1, 3, 6, 10},
5         channel_charge[2][BLKNUM] = {{1500, 1000, 700, 500},
6                                     {1000, 800, 600, 400}};

7         /* インターネット接続サービス利用料金,
8             0 は契約なしの場合 */
9 const int inet_charge[] = {0, 2500, 3000, 5000, 6000};

10 int calc_service_fee(int, int, int);

11 int calc_service_fee(int basic_plan, int channel_num,
12                      int inet_plan) {
13     int tv_fee, inet_fee;
14     int cnum, i;

```

```

15     /* 有線テレビ視聴料金 tv_fee を求める */
16     tv_fee = basic_charge[basic_plan];
17     if (basic_plan > 0) {
18         for (i = BLKNUM - 1; i >= 0; i--) {
19             cnum = channel_num - channel_block[i] + 1;
20             if (cnum < 0) {
21                 cnum = 0;
22             }
23             channel_num -= cnum;
24             tv_fee += cnum * channel_charge[basic_plan - 1][i];
25         }
26     }
27
28     /* インターネット接続サービス利用料金 inet_fee を求める */
29     if (inet_plan == [e1]) {
30         inet_fee = inet_charge[0];
31     } else if (inet_plan == [e2]) {
32         if (basic_plan != 0) {
33             inet_fee = inet_charge[1];
34         } else {
35             inet_fee = inet_charge[2];
36         }
37     } else {
38         if (basic_plan [f]) {
39             inet_fee = inet_charge[3];
40         } else {
41             inet_fee = inet_charge[4];
42         }
43     }
44 }
```

設問 1 有線テレビ視聴料金に関する次の記述中の [ ] に入る正しい答えを、

解答群の中から選べ。

- (1) calc\_service\_fee(1, 6, 0) を実行した場合、行番号 23 が 2 回目に実行されるときの変数 cnum の値は [a] となり、3 回目に実行されるときの変数 cnum の値は [b] となる。また、行番号 43 が実行されるときの変数 tv\_fee の値は [c] となる。
- (2) 基本視聴契約がプラン 1 で有料放送 6 チャンネル分の視聴契約をする場合の有線テレビ視聴料金と、基本視聴契約がプラン 2 で有料放送 6 チャンネル分の視聴契約をする場合の有線テレビ視聴料金を比較した場合、[d]。

a, b に関する解答群

ア 0

イ 1

ウ 2

エ 3

オ 4

カ 5

キ 6

ク 10

c に関する解答群

ア 6600

イ 7200

ウ 8000

エ 9700

C

d に関する解答群

ア どちらのプランでも有線テレビ視聴料金は等しい

イ プラン1での有線テレビ視聴料金の方が高い

ウ プラン2での有線テレビ視聴料金の方が高い

設問2 プログラム中の  に入る正しい答えを、解答群の中から選べ。ただし、e1 と e2 に入れる答えは、e に関する解答群の中から組合せとして正しいものを選ぶものとする。

e に関する解答群

	e1	e2
ア	0	1
イ	0	2
ウ	1	0
エ	1	2
オ	2	0
カ	2	1

f に関する解答群

ア == 0

イ == 1

ウ == 2

エ != 0

オ != 1

カ != 2

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 1～3 に答えよ。

[プログラムの説明]

1993 年 4 月 1 日に創立した X 社では、創立 20 周年の記念日である 2013 年 4 月 1 日に、勤続年数が 0～20 年の全ての従業員に特別給を支給することにした。特別給は、基本給に、表 1 に示す決定表によって導出される支給係数を掛け合わせて百円未満を切り捨てた金額であり、創立記念日に上司が現金の入った封筒を部下に手渡す。このプログラムは、従業員ファイルに格納されている従業員の情報を読み込み、各従業員への支給金額と、各金種の必要数を支給ファイルに出力する。ここで、金種は 1 万円紙幣、5 千円紙幣、千円紙幣、5 百円硬貨、百円硬貨の 5 種類とする。

表1 支給係数の決定表

条件 1 (勤続年数)	3 年以下	Y	Y	N	N	N	N
	4～9 年	N	N	Y	Y	N	N
	10 年以上	N	N	N	N	Y	Y
条件 2 (役職)	一般職	Y	N	Y	N	Y	N
	管理職	N	Y	N	Y	N	Y
動作 (支給係数)	0.05	X	—	—	—	—	—
	0.10	—	X	X	—	—	—
	0.15	—	—	—	X	X	—
	0.20	—	—	—	—	—	X

(1) 従業員ファイルは、図 1 に示すレコード様式の順ファイルである。

従業員番号 6 枠	役職 1 枠	入社日 8 枠	基本給 6 枠
--------------	-----------	------------	------------

図 1 従業員ファイルのレコード様式

- ① 役職には、一般職の場合は 0、管理職の場合は 1 を格納する。
- ② 入社日には、入社した年、月、日を、それぞれ 4 枠、2 枠、2 枠の西暦で格納する。表 1 に示す支給係数の決定表の条件 1（勤続年数）は、入社日から 2013

年4月1日までの経過年数とする。例えば、入社日が2012年4月1日の場合は1年であり、2012年4月2日の場合は0年である。

(2) 支給ファイルは、図2に示すレコード様式の順ファイルである。

従業員番号 6桁	支給金額 6桁	金種別必要数				
		1万円紙幣 2桁	5千円紙幣 2桁	千円紙幣 2桁	5百円硬貨 2桁	百円硬貨 2桁

図2 支給ファイルのレコード様式

- ① 支給金額には、基本給と支給係数から算出した支給金額を格納する。
- ② 金種別必要数には、それぞれの金種の必要数を格納する。ここで、必要数の合計が最少となるようにする。

### [プログラム]

#### (行番号)

```

1  DATA DIVISION.
2  FILE SECTION.
3  FD EMP-F.
4  01 E-REC.
5    02 E-NO          PIC 9(6).
6    02 E-STATUS      PIC 9(1).
7      88 NON-MANAGER VALUE 0.
8      88 MANAGER     VALUE 1.
9    02 E-JOIN.
10   03 E-YEAR        PIC 9(4).
11   03 E-MMDD        PIC 9(4).
12   02 E-BASE        PIC 9(6).
13 FD PAY-F.
14 01 P-REC.
15  02 P-NO          PIC 9(6).
16  02 P-PAY         PIC 9(6).
17  02 P-CUR-TBL.
18  03 P-CUR         PIC 9(2) OCCURS 5.
19 WORKING-STORAGE SECTION.
20 77 EMP-FLAG       PIC X(1) VALUE SPACE.
21  88 EMP-EOF       VALUE "E".
22 77 W-YEAR         PIC 9(2).
23  88 UPTO-3         VALUE 0 THRU 3.
24  88 UPTO-9         VALUE 4 THRU 9.
25  88 UPTO-20        VALUE 10 THRU 20.
26 77 PAY-RATE       PIC 9V99.
27 77 W-PAY          PIC 9(6).

```

```
28 01 CUR-VALUE      PIC X(25) VALUE "1000005000010000050000100".
29 01 CUR-TBL REDEFINES CUR-VALUE.
30     02 CUR-KIND      PIC 9(5) OCCURS 5.
31 01 CNT            PIC 9(1).
32 PROCEDURE DIVISION.
33 MAIN-PROC.
34     OPEN INPUT  EMP-F
35             OUTPUT PAY-F.
36 PERFORM UNTIL EMP-EOF
37         READ EMP-F AT END      SET EMP-EOF TO TRUE
38                 NOT AT END MOVE E-NO TO P-NO
39                         PERFORM DEC-PROC
40                         PERFORM CUR-PROC
41                         WRITE P-REC
42         END-READ
43 END-PERFORM.
44 CLOSE EMP-F PAY-F.
45 STOP RUN.
46 DEC-PROC.
47 COMPUTE W-YEAR = 2013 - E-YEAR.
48 IF a THEN
49     SUBTRACT 1 FROM W-YEAR
50 END-IF.
51 EVALUATE TRUE
52     WHEN UPTO-3 AND NON-MANAGER
53         MOVE 0.05 TO PAY-RATE
54     WHEN UPTO-3 AND MANAGER
55     WHEN UPTO-9 AND NON-MANAGER
56         MOVE 0.10 TO PAY-RATE
57     WHEN UPTO-9 AND MANAGER
58     WHEN UPTO-20 AND NON-MANAGER
59         MOVE 0.15 TO PAY-RATE
60     WHEN UPTO-20 AND MANAGER
61         MOVE 0.20 TO PAY-RATE
62 END-EVALUATE.
63 COMPUTE P-PAY = E-BASE * PAY-RATE.
64 DIVIDE 100 INTO P-PAY.
65 MULTIPLY 100 BY P-PAY.
66 CUR-PROC.
67     INITIALIZE P-CUR-TBL.
68     MOVE 1 TO CNT.
69     MOVE P-PAY TO W-PAY.
70     PERFORM UNTIL W-PAY = 0
71         IF W-PAY >= CUR-KIND(CNT) THEN
72             ADD 1 TO P-CUR(CNT)
73             b
74         ELSE
75             c
76         END-IF
77 END-PERFORM.
```

設問 1 プログラム中の  に入る正しい答えを、解答群の中から選べ。

a に関する解答群

- |                 |                 |
|-----------------|-----------------|
| ア E-MMDD < 0401 | イ E-MMDD > 0401 |
| ウ W-YEAR < 20   | エ W-YEAR >= 20  |

b, c に関する解答群

- |                                     |
|-------------------------------------|
| ア ADD 1 TO CNT                      |
| イ ADD CNT TO P-CUR(CNT)             |
| ウ MOVE CUR-KIND(CNT) TO P-CUR(CNT)  |
| エ SUBTRACT CUR-KIND(CNT) FROM W-PAY |
| オ SUBTRACT P-CUR(CNT) FROM W-PAY    |

設問 2 プログラム中の、支給係数を求める EVALUATE 文について、正しく判定できる  
ことを確認するために、従業員ファイルのテストデータを作成した。全ての選択  
対象 (WHEN 指定に書かれた作用対象を指す) について真を判定できる組合せを、  
解答群の中から選べ。ここで、解答群では見やすくするために項目の区切りに  
“,” を挿入しているが、実際のデータ中には存在しない。

解答群

- |                            |                            |
|----------------------------|----------------------------|
| ア 000001,0,20100901,117000 | イ 000001,0,20110401,115000 |
| 000002,1,20100401,121000   | 000002,1,20110901,132000   |
| 000003,0,20060401,151000   | 000003,0,20060401,151000   |
| 000004,1,20020901,192000   | 000004,1,20060401,186000   |
| 000005,0,20000401,198000   | 000005,0,20010401,195000   |
| 000006,1,19980401,254000   | 000006,1,19970901,223000   |
| ウ 000001,0,20120401,102000 | エ 000001,0,20120901,110800 |
| 000002,1,20080901,132000   | 000002,1,20100401,132000   |
| 000003,0,20070401,151000   | 000003,0,20080401,151000   |
| 000004,1,20050901,186000   | 000004,1,20020901,176000   |
| 000005,0,20040401,188000   | 000005,0,20010401,198000   |
| 000006,1,20010901,223000   | 000006,1,19950901,254000   |

設問 3 金融機関に現金を用意してもらうために、金種ごとの必要数を集計して、図3に示すとおり画面に表示するようプログラムを変更する。表2中の [ ] に入れる正しい答えを、解答群の中から選べ。

10000 : 00001415
05000 : 00000267
01000 : 00002189
00500 : 00000311
00100 : 00002052

図3 画面表示の例

表2 プログラムの変更

処置	変更内容
行番号 30 と 31 の間に追加	<pre>01 CUR-TBL-X REDEFINES CUR-VALUE.   02 CUR-KIND-X  PIC X(5) OCCURS 5.   01 TOTAL-TBL.     02 CUR-TOTAL  PIC 9(8) OCCURS 5 VALUE ZERO.   01 TOTAL-TBL-X REDEFINES TOTAL-TBL.     02 CUR-TOTAL-X  PIC X(8) OCCURS 5.</pre>
[d]	<pre>PERFORM VARYING CNT FROM 1 BY 1 UNTIL CNT &gt; 5   DISPLAY CUR-KIND-X(CNT) " : " CUR-TOTAL-X(CNT) END-PERFORM.</pre>
行番号 77 の後ろに追加	<pre>PERFORM VARYING CNT FROM 1 BY 1 UNTIL CNT &gt; 5   [e] END-PERFORM.</pre>

d に関する解答群

- ア 行番号 42 と 43 の間に追加
- ウ 行番号 46 と 47 の間に追加

- イ 行番号 44 と 45 の間に追加
- エ 行番号 66 と 67 の間に追加

e に関する解答群

- ア ADD 1 TO CUR-TOTAL(CNT)
- イ ADD CUR-KIND(CNT) TO CUR-TOTAL(CNT)
- ウ ADD P-CUR(CNT) TO CUR-TOTAL(CNT)
- エ ADD P-PAY TO CUR-TOTAL(CNT)

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 1～3 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]

あみだくじの作成と結果の表示を行うプログラムである。

あみだくじとは、複数の平行に並ぶ縦線と、2 本の縦線だけに水平に接続する複数の横線から成るくじである。横線は縦線の上端及び下端には接続せず、縦線の同一箇所に複数の横線が接続されることもない。くじをたどる手順は次のとおりである。

- (1) 1 本の縦線を選択し、上端から下方向にたどり始める。
- (2) 途中に、接続する横線があるときは必ず曲がり、もう一方の縦線までたどる。
- (3) 縦線に到達したら、また下方向にたどる。
- (4) 下端に到達するまで(2)と(3)を繰り返す。
- (5) 下端に到達したときの縦線の位置が、くじを引いた結果である。

あみだくじの例を図 1 に示す。図中の太線は、左端の縦線を選択したときにたどった経路である。

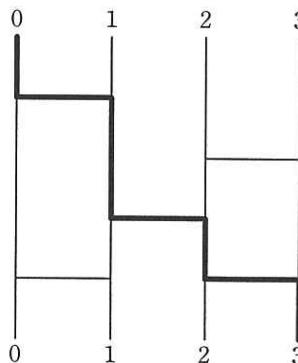


図 1 あみだくじの例

クラス `GhostLeg` は、あみだくじを作成し、作成したあみだくじをたどるプログラムである。

- (1) 縦線を左からの順番で保持する。左端を 0 番目とする。
- (2) 縦線の長さを 1.0 とし、上端の縦軸座標を 1.0、下端の縦軸座標を 0.0 とする。
- (3) 横線の両端の縦軸座標は等しく、0.0 よりも大きく 1.0 未満である。

(4) 縦線の本数を引数とするコンストラクタをもつ。

(5) 次のメソッドをもつ。

① `void addHorizontalLine(int x1, int x2, double y)`

横線を追加するメソッドである。左から  $x_1$  番目の縦線と  $x_2$  番目の縦線を縦軸座標  $y$  で接続する横線を追加する。ただし、 $x_1$  と  $x_2$  が等しいか、2 本の縦線のいずれか又は両方に、既に縦軸座標  $y$  で接続する横線があるときには追加しない。

あみだくじは、クラス `GhostLeg` のインスタンスを生成した後、このメソッドを任意の回数だけ呼び出し、横線を追加することによって完成する。

② `int trace(int x)`

くじを引いた結果を返すメソッドである。左から  $x$  番目の縦線を選択し、あみだくじをたどった結果を返す。

クラス `GhostLegTester` はテスト用のプログラムである。このプログラムは、図 1 に示したあみだくじを作り、左から順にくじを引き、結果を表示する。実行結果を、図 2 に示す。

0 -> 3
1 -> 1
2 -> 2
3 -> 0

図 2 クラス `GhostLegTester` の実行結果

[プログラム 1]

```
import java.util.ArrayList;
import java.util.List;
import java.util.SortedMap;
import java.util.TreeMap;

public class GhostLeg {
    private List<VerticalLine> verticalLines;

    public [ ] a (int n) {
        verticalLines = new ArrayList<VerticalLine>(n);
        for (int i = 0; i < n; i++) {
            verticalLines.add([ ] b );
        }
    }
}
```

```

public void addHorizontalLine(int x1, int x2, double y) {
    // 設問3(α)
    VerticalLine v1 = verticalLines.get(x1);
    // 設問3(β)
    VerticalLine v2 = verticalLines.get(x2);
    if (x1 != x2 && !v1.hasHorizontalLineAt(y) &&
        !v2.hasHorizontalLineAt(y)) {
        // 設問3(γ)
        v1.putHorizontalLine(y, v2);
        v2.putHorizontalLine(y, v1);
    }
    // 設問3(δ)
}

public int trace(int x) {
    double y = 1.0;
    VerticalLine v = verticalLines.get(x);
    while ((y = v.getNextY(y)) > [c]) {
        v = [d];
    }
    return verticalLines.indexOf(v);
}

private static class VerticalLine {
    SortedMap<Double, VerticalLine> horizontalLines =
        new TreeMap<Double, VerticalLine>();

    VerticalLine() {
        horizontalLines.put(0.0, null);
    }

    boolean hasHorizontalLineAt(double y) {
        return horizontalLines.containsKey(y);
    }

    void putHorizontalLine(double y, VerticalLine opposite) {
        horizontalLines.put(y, opposite);
    }

    double getNextY(double y) {
        // マップ horizontalLines が保持するキーの中で、
        // 引数 y よりも小さいもののうち、最大のものを返す。
        return horizontalLines.headMap(y).lastKey();
    }

    VerticalLine getOpposite(double y) {
        return horizontalLines.get(y);
    }
}
}

```

[プログラム2]

```
public class GhostLegTester {
    public static void main(String[] args) {
        GhostLeg gh = new GhostLeg(4);
        gh.addHorizontalLine(0, 1, 0.8);
        gh.addHorizontalLine(0, 1, 0.2);
        gh.addHorizontalLine(1, 2, 0.4);
        gh.addHorizontalLine(2, 3, 0.6);
        gh.addHorizontalLine(2, 3, 0.2);
        for (int i = 0; i < 4; i++) {
            System.out.printf("%d -> %d%n", i, gh.trace(i));
        }
    }
}
```

設問1 次の記述中の [ ] に入る正しい答えを、解答群の中から選べ。

クラス GhostLeg の内部クラス VerticalLine は 1 本の縦線を表すクラスであり、インスタンスは、そのインスタンスが示す縦線に接続している横線を保持している。この内部クラスでは、横線を、 [ ] として表現している。

### 解答群

- ア クラス HorizontalLine のインスタンスの列
- イ その横線が接続する縦軸座標とクラス HorizontalLine のインスタンスのマップ
- ウ その横線が接続する縦軸座標とその横線に接続するもう 1 本の縦線のマップ
- エ その横線に接続する 2 本の縦線のマップ

設問2 プログラム中の [ ] に入る正しい答えを、解答群の中から選べ。

### a に関する解答群

- |                 |                 |
|-----------------|-----------------|
| ア ghostLeg      | イ GhostLeg      |
| ウ void ghostLeg | エ void GhostLeg |

## b に関する解答群

- |                      |        |
|----------------------|--------|
| ア 0                  | イ i    |
| ウ new VerticalLine() | エ null |

## c に関する解答群

- |        |       |     |     |
|--------|-------|-----|-----|
| ア -1.0 | イ 0.0 | ウ x | エ y |
|--------|-------|-----|-----|

## d に関する解答群

- |                          |                          |
|--------------------------|--------------------------|
| ア horizontalLines.get(x) | イ horizontalLines.get(y) |
| ウ v.getOpposite(y)       | エ verticalLines.get(x)   |

設問 3 次の記述中の    に入る正しい答えを、解答群の中から選べ。

メソッド addHorizontalLine の呼出しにおいて、引数  $x_1$  又は  $x_2$  に相当する縦線がないときには IndexOutOfBoundsException が投げられる。一方、引数  $y$  には追加する横線の縦軸座標を指定するので、その値は 0.0 よりも大きく 1.0 未満でなければならないが、それ以外の値を指定しても例外は投げない。

これを、引数  $x_1$  及び  $x_2$  の値によらず、引数  $y$  の値が横線の縦軸座標として取り得る範囲から外れているときに IllegalArgumentException を投げるようにするには、プログラム 1 中のコメント “// 設問 3 ( e )” の 1 行を次の if 文に置き換えればよい。

```
if ( f ) {
    throw new IllegalArgumentException();
}
```

## e に関する解答群

- |            |           |            |            |
|------------|-----------|------------|------------|
| ア $\alpha$ | イ $\beta$ | ウ $\gamma$ | エ $\delta$ |
|------------|-----------|------------|------------|

## f に関する解答群

- |                                    |                                  |
|------------------------------------|----------------------------------|
| ア $y \leq 0.0 \ \&\& \ y \geq 1.0$ | イ $y \leq 0.0 \    \ y \geq 1.0$ |
| ウ $y > 0.0 \ \&\& \ y < 1.0$       | エ $y > 0.0 \    \ y < 1.0$       |

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 1～4 に答えよ。

[プログラム 1 の説明]

連続する 2 語から成るビット列  $\alpha$  について、左端のビット位置を 0 としたとき、ビット位置  $p$  から始まる  $q$  ビットを、別のビット列  $\beta$  で置き換える副プログラム BREP である。ここで、 $0 \leq p < 16$ ,  $1 \leq q \leq 16$  とする。置換えの概要を図 1 に示す。

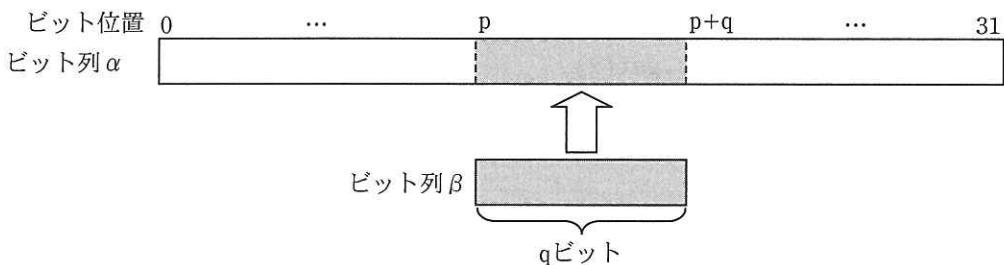


図 1 置換えの概要

- (1) ビット列  $\alpha$  の先頭アドレスは GR1 に、 $p$  は GR2 に、 $q$  は GR3 に、それぞれ設定されて主プログラムから渡される。
- (2) ビット列  $\beta$  は GR0 に左詰めで設定され、GR0 の残りの部分は 0 で埋められて主プログラムから渡される。
- (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]

(行番号)

```

1  BREP    START
2          RPUSH
3          LD     GR4,GR0
4          LD     GR5,GR0
5          LD     GR6, a           ; MASK パターン生成の準備
6          LAD    GR3,-1,GR3
7          SRA    GR6,0,GR3   ; GR6 ← q ビットの MASK パターン生成
8          LD     GR7,GR6
9          LD     GR3,=16
10         SUBA   GR3,GR2   ; GR3 ← 16 - p
11         SRL    GR4,0,GR2   ; 1 語目用置換文字列の準備
12         SLL    GR5,0,GR3   ; 2 語目用置換文字列の準備
13         SRL    GR6,0,GR2   ; 1 語目用 MASK パターン生成
14         SLL    GR7,0,GR3   ; 2 語目用 MASK パターン生成
15         LD     GR2,0,GR1   ; 1 語目の処理
16         XOR    GR6,=#FFFF
17         AND    GR2,GR6
18         b      GR2,GR4
19         ST     GR2,0,GR1
20         LD     GR2,1,GR1   ; 2 語目の処理
21         XOR    GR7,=#FFFF
22         AND    GR2,GR7
23         b      GR2,GR5
24         ST     GR2,1,GR1
25         RPOP
26         RET
27         END

```

設問 1 プログラム 1 中の    に入れる正しい答えを、解答群の中から選べ。

アセンブリ

a に関する解答群

ア =#0001	イ =#000F	ウ =#7FFF	エ =#8000
オ =#F000	カ =#FFFF		

b に関する解答群

ア AND	イ LD	ウ OR	エ SUBA
オ SUBL			

設問 2 次の記述中の  に入る正しい答えを、解答群の中から選べ。

主プログラムから渡された  $p$ ,  $q$  の値及びビット列  $\beta$  が、次のとおりであった。

$p$  (GR2) : 12

$q$  (GR3) : 10

$\beta$  (GR0) : #D6C0

このとき、プログラム 1 の行番号 14 の SLL 命令を実行した直後における GR4 の内容は  c であり、GR6 の内容は  d である。

c, d に関する解答群

ア #0000

イ #000D

ウ #000F

エ #0035

オ #003F

カ #FFC0

キ #FFF0

ク #FFFF

設問 3 プログラム 1 の行番号 16, 17 を、同じ効果をもつ次の命令で置き換えた。

に入る正しい答えを、解答群の中から選べ。

16                        GR6, GR2  
17            XOR            GR2, GR6

解答群

ア AND

イ LD

ウ OR

エ SUBA

オ SUBL

カ XOR

設問4 ビット列 $\alpha$ を連続するn語( $n > 1$ )から成るビット列とした場合に同様の置換えを行う副プログラムXBREPを、BREPを使用して作成した。ここで、 $0 \leq p < 16 \times (n - 1)$ ,  $1 \leq q \leq 16$ とし、それ以外の仕様はBREPと同じとする。プログラム2中の [ ] に入る正しい答えを、解答群の中から選べ。

[プログラム2]

```
XBREP    START
          RPUSH
          LD      GR7,GR2
          [ ]      e
          ADDL   GR1,GR7
          AND    GR2,[ ] f
          CALL   BREP
          RPOP
          RET
          END
```

eに関する解答群

ア SLL GR2,4	イ SLL GR3,4	ウ SLL GR7,4	エ SRL GR2,4
オ SRL GR3,4	カ SRL GR7,4		

fに関する解答群

ア =#0001	イ =#000F	ウ =#7FFF	エ =#8000
オ =#F000	カ =#FFFF		

問 13 次の表計算、ワークシート及びマクロの説明を読んで、設問 1～3 に答えよ。

[表計算の説明]

A 社は、製品コード PA1, PA2, PA3, AS1 の 4 種類の製品の生産計画を立案するために、三つのワークシート “部品一覧”, “受注一覧”, “生産管理” を作成することにした。各製品の生産計画の概要は次のとおりである。

- (1) 各製品の生産に必要な部品とその個数は、図 1 のワークシート “部品一覧” のとおりであり、製品 AS1 の生産には、製品 PA1, PA2, PA3 を部品として用いる。各部品は、生産着手時に投入され、着手した製品は全てその日のうちに完成する。
- (2) PA1, PA2, PA3 は、1 日の生産基準数をあらかじめ設定しておき、これに翌日生産分の AS1 の部品として必要な個数を加えた値を生産数とする。
- (3) 生産基準数は、1 週間ごとに決定する。
- (4) 生産基準数は、これを決定する週と翌週の 2 週間分の納品予定に基づき、納品予定日の前日までに納品数を用意するとともに、一定の余裕をもった在庫数を確保するように決定する。
- (5) PA1, PA2, PA3 は、10 個単位で生産する。各製品の 1 日の生産数の上限は、図 1 のワークシート “部品一覧” の最大生産数のとおりである。
- (6) AS1 は、20 個単位で生産する。AS1 は、受注生産品であり、納品予定日の前日に、翌日の納品に必要な最低限の個数を 20 個単位で生産する。ただし、1 日の最大生産数を超える受注は受け付けない。
- (7) ワークシートの日付は、yyyy-mm-dd の形式で表示する。表計算ソフトの内部では、1970 年 1 月 1 日を 1 とした経過日数を整数で保持している。

[ワークシート：部品一覧]

ワークシート“部品一覧”には、各製品を生産するために必要な部品の個数と、1日に生産可能な最大生産数を、図1のように格納してある。

	A	B	C	D	E	F	G	H	I
1 製品 コード	部品名	部品1	部品2	部品3	部品4	PA1	PA2	PA3	最大生産数
2	PA1	1	2						500
3	PA2	1	3	2					600
4	PA3		2		3				400
5	AS1					1	2	1	100

図1 ワークシート“部品一覧”

[ワークシート：受注一覧]

ワークシート“受注一覧”には、受注データのデータベースから、生産基準数を決定する週及びその翌週に納品予定日を迎えるデータを抽出し、受注日順に格納した後、作業コードを付与する。ワークシート“受注一覧”的例を、図2に示す。

	A	B	C	D	E	F
1	受注日	得意先コード	製品コード	数量	納品予定日	作業コード
2	2013-03-19	3001	PA1	630	2013-04-02	15798PA1
3	2013-03-20	3005	PA1	460	2013-04-03	15799PA1
4	2013-03-20	2012	PA2	450	2013-04-01	15797PA2
5	2013-03-21	2001	PA1	380	2013-04-04	15800PA1
:	:	:	:	:	:	:
34	2013-03-31	1006	AS1	84	2013-04-14	15810AS1
35	2013-03-31	2005	PA1	420	2013-04-14	15810PA1
36	2013-03-31	3009	PA2	280	2013-04-14	15810PA2
37	2013-03-31	2003	PA3	225	2013-04-12	15808PA3

図2 ワークシート“受注一覧”的例

- (1) 抽出した2週間分のデータの件数は、最大でも100件を超えないものとする。
- (2) セルF2には次の式を入力し、セルF3～F101に複写する。

結合(E2, C2)

[ワークシート：生産管理]

ワークシート“生産管理”は、ワークシート“受注一覧”から各製品の納品数を納品予定日ごとに集計する。また、各製品の生産基準数の設定値を変えることによって、在庫や欠品状況などの生産管理に必要な情報のシミュレーションができる。

ワークシート“生産管理”的例を、図3に示す。

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
1	生産管理表	2013-04-01	から	2013-04-14	まで											
2																
3	日付	PA1			PA2			PA3			AS1					
4		納品 数	納品 後在 庫	生産 数	生産 後在 庫	納品 数	納品 後在 庫	生産 数	生産 後在 庫	納品 数	納品 後在 庫	生産 数	生産 後在 庫	納品 数	生産 数	在庫 数
5	前週繰越				460				600				550			8
6	2013-04-01	0	460	380	840	450	150	250	400	0	550	260	810	0	0	8
7	2013-04-02	670	170	340	510	80	320	170	490	40	770	220	990	0	40	48
8	2013-04-03	460	50	340	390	0	490	170	660	0	990	220	1,210	36	0	12
9	2013-04-04	380	10	400	410	480	180	290	470	240	970	280	1,250	0	0	12
10	2013-04-05	60	350	340	690	470	0	170	170	750	500	220	720	0	60	72
11	2013-04-06	0	690	380	1,070	0	170	250	420	0	720	260	980	60	0	12
12	2013-04-07	40	1,030	420	1,450	80	340	330	670	40	940	300	1,240	0	40	52
13	2013-04-08	680	770	340	1,110	480	190	170	360	545	695	220	915	48	80	84
14	2013-04-09	400	710	340	1,050	290	70	170	240	570	345	220	565	75	0	9
15	2013-04-10	360	690	340	1,030	200	40	170	210	405	160	220	380	0	0	9
16	2013-04-11	520	510	340	850	0	210	170	380	420	-40	220	180	0	0	9
17	2013-04-12	480	370	420	790	210	170	330	500	225	-45	300	255	0	0	9
18	2013-04-13	560	230	340	570	380	120	170	290	80	175	220	395	0	80	89
19	2013-04-14	420	150	340	490	280	10	170	180	300	95	220	315	84	0	5
20	合計	5,030		5,060		3,400		2,980		3,615		3,380				
21	平均	359	442	361	804	243	176	213	389	258	488	241	729			
22		生産基準数		340	生産基準数		170	生産基準数		220						
23		欠品日数		0	欠品日数		0	欠品日数		2						
24		生産基準数 の妥当性		true	生産基準数 の妥当性		false	生産基準数 の妥当性		false						

図3 ワークシート“生産管理”的例

設問1 ワークシート“生産管理”に関する次の記述中の [ ] に入る正しい答えを、解答群の中から選べ。

- (1) セル B1 には、生産管理の開始日を、セル E1 には、開始日から数えて 14 日目の日付を表示する式を入力し、この間を生産管理期間とする。また、セル A6～A19 には、生産管理期間中の一連の日付を表示する式を入力する。
- (2) セル B3, F3, J3, N3 には、製品コードを入力する。
- (3) セル E5, I5, M5, P5 には、各製品の前週末の実際の在庫数を入力する。
- (4) セル E22, I22, M22 には、製品 PA1, PA2, PA3 の生産基準数を、最大生産数を超えない範囲で入力する。
- (5) セル N6 には、納品予定日がセル A6 の日付の製品 AS1 の納品数を求める式を入力し、セル N7～N19 に複写する。
- (6) セル B6 には、納品予定日がセル A6 の日付であり、製品コードがセル B3 のコードである製品の納品数を求める次の式を入力し、セル B7～B19, セル F6～F19, セル J6～J19 に複写する。

[ a ] + \$O6 \* 水平照合(B\$3, 部品一覧!\$F\$1～\$H\$5, 5, 0)

- (7) 列 D, 列 H, 列 L の生産数欄には、製品 PA1, PA2, PA3 の 1 日の生産基準数 E22, I22, M22 に、翌日生産する製品 AS1 の部品として必要な個数を加えた値を求める式を入力する。ただし、行 19 の生産数欄は、1 日の生産基準数だけが設定されるようにする。
- (8) 列 C, 列 G, 列 K の納品後在庫欄には、前日の生産後庫から当日の納品数を減じた値を求める式を入力する。また、列 E, 列 I, 列 M の生産後庫欄には、当日の納品後庫に生産数を加えた値を求める式を入力する。
- (9) セル O6 には、製品 AS1 の当日の在庫数が翌日の納品数を満たすために最低限必要な生産数を求める次の式を入力し、セル O7～O19 に複写する。

IF([ b ])

- (10) 列 P の在庫数欄には、前日の在庫数から当日の納品数を減じ、当日の生産数を加えた値を求める式を入力する。
- (11) セル B20, D20, F20, H20, J20, L20 の合計欄には、各列の合計、セル B21～M21 の平均欄には、各列の平均を求める式を入力する。

(12) セル E23, I23, M23 の欠品日数には、各製品において納品後在庫がマイナスになった日数を求める式を入力する。

(13) セル E24 には、セル B3 の製品の生産基準数の設定値が、生産管理期間において次の条件を全て満たすときに true, そうでないときに false を表示する式  c を入力し、セル I24, M24 に複写する。

[条件]

- ① 欠品日数が 0 であること
- ② 余裕をもった在庫数を、セル B21 で求めた納品数平均の 80% の値としたとき、毎日の生産後在庫のうちで、その値に満たない日が 1 日もないこと

a に関する解答群

ア 条件付合計(受注一覧!\$D\$2～\$D\$101, = B\$3, 受注一覧!\$C\$2～\$C\$101)

イ 条件付合計(受注一覧!\$D\$2～\$D\$101, = 結合(\$A6, B\$3), 受注一覧!\$F\$2～\$F\$101)

ウ 条件付合計(受注一覧!\$D\$2～\$D\$101, = 結合(B\$3, \$A6), 受注一覧!\$F\$2～\$F\$101)

エ 条件付合計(受注一覧!\$E\$2～\$E\$101, = \$A6, 受注一覧!\$D\$2～\$D\$101)

オ 条件付合計(受注一覧!\$F\$2～\$F\$101, = 結合(\$A6, B\$3), 受注一覧!\$D\$2～\$D\$101)

カ 条件付合計(受注一覧!\$F\$2～\$F\$101, = 結合(B\$3, \$A6), 受注一覧!\$D\$2～\$D\$101)

b に関する解答群

ア  $P5 - N6 < N7, 0, \text{切上げ}((N7 + N6 - P5) / 20, 0) * 20$

イ  $P5 - N6 < N7, 0, \text{整数部}((N7 + N6 - P5) / 20) * 20$

ウ  $P5 - N6 < N7, \text{切上げ}(N7 + N6 - P5, -1), 0$

エ  $P5 - N6 \geq N7, 0, \text{切上げ}((N7 + N6 - P5) / 20, 0) * 20$

オ  $P5 - N6 \geq N7, 0, \text{整数部}((N7 + N6 - P5) / 20) * 20$

カ  $P5 - N6 \geq N7, \text{切上げ}(N7 + N6 - P5, -1), 0$

### cに関する解答群

- ア 論理積(E23 = 0, 条件付個数(E6～E19, >B21 \* 0.8) = 0)  
イ 論理積(E23 = 0, 条件付個数(E6～E19, <B21 \* 0.8) = 0)  
ウ 論理積(E23 > 0, 条件付個数(E6～E19, >B21 \* 0.8) = 0)  
エ 論理和(E23 = 0, 条件付個数(E6～E19, <B21 \* 0.8) = 0)  
オ 論理和(E23 > 0, 条件付個数(E6～E19, <B21 \* 0.8) = 0)  
カ 論理和(E23 > 0, 条件付個数(E6～E19, >B21 \* 0.8) = 0)

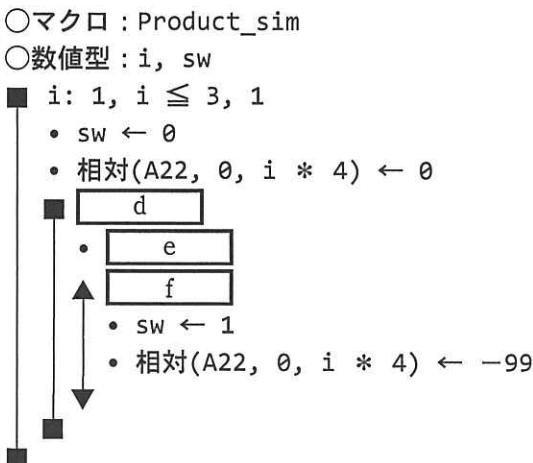
設問2 ワークシート“生産管理”において、セル E24, I24, M24 の値が true となる最小の生産基準数を、セル E22, I22, M22 に求めるマクロ Product\_sim を作成し、ワークシート“生産管理”に格納した。マクロ Product\_sim 中の  
[ ] に入る正しい答えを、解答群の中から選べ。

## [マクロの説明]

- (1) 製品 PA1 の生産基準数を求めるために、セル E22 の値を 0 から生産単位ずつ增加させる処理を、セル E24 の値が true になるまで繰り返す。ここで、もし生産基準数がワークシート “部品一覧” で示した各製品の最大生産数を超えて、セル E24 の条件が成立しない場合は、セル E22 に -99 を代入して、PA1 の生産基準数を求める繰返し処理を終了する。

(2) 製品 PA2, PA3 の生産基準数を求めるために、(1)でセル E22 に対して行った処理と同様の処理を、セル I22, セル M22 でも実行する。

[マクロ : Product sim]



d, f に関する解答群

- ア 相対(A22, 0, i) < 相対(部品一覧!I1, i \* 4, 0)
- イ 相対(A22, 0, i) > 相対(部品一覧!I1, i \* 4, 0)
- ウ 相対(A22, 0, i \* 4) < 相対(部品一覧!I1, i, 0)
- エ 相対(A22, 0, i \* 4) > 相対(部品一覧!I1, i, 0)
- オ 論理積(相対(A24, 0, i \* 4) = false, sw = 0)
- カ 論理積(相対(E24, 0, i) = false, sw = 1)
- キ 論理和(相対(A24, 0, i \* 4) = false, sw = 0)
- ク 論理和(相対(E24, 0, i) = false, sw = 1)

e に関する解答群

- ア  $i \leftarrow i + 1$
- イ  $i \leftarrow i + 4$
- ウ  $\text{相対}(A22, 0, 4) \leftarrow 0$
- エ  $\text{相対}(A22, 0, i * 4) \leftarrow 0$
- オ  $\text{相対}(A22, 0, i * 4) \leftarrow \text{相対}(A22, 0, i * 4) + 1$
- カ  $\text{相対}(A22, 0, i * 4) \leftarrow \text{相対}(A22, 0, i * 4) + 10$
- キ  $\text{相対}(E22, 0, i) \leftarrow \text{相対}(E22, 0, i) + 1$
- ク  $\text{相対}(E22, 0, i) \leftarrow \text{相対}(E22, 0, i) + 10$

設問 3 A 社では、部品の仕入計画を立案するために、ワークシート“生産管理”でマクロ Product\_sim を実行して求めた生産数を基に、1 週間の生産に必要な部品数を求めるワークシート“部品需要”を図 4 の例のように作成した。ワークシート“部品需要”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には、同じ式が入るが、表示していない。

## 〔ワークシート：部品需要〕

- (1) セル A2 には、部品の需要数を求める期間の開始日を入力する。
- (2) 部品需要表の列 A の日付は、セル A2 の日付を初日とした 1 週間分とし、日付ごとの列 B に製品 PA1, PA2, PA3 の欄を用意する。
- (3) セル C5 には、必要な部品数を求める式を入力する。ここで、ワークシート“生産管理”から当該日（セル A5 の日付）の当該製品（セル B5 の製品コード）の生産数を参照する式は、垂直照合（\$A5, 生産管理 !\$A\$6 ~ \$M\$19, [ ] g, 0）である。

また、ワークシート“部品一覧”から当該製品を 1 個生産するために必要な当該部品（セル C4 の部品名）の数を参照する式は、[ ] である。そこで、セル C5 には、次の式を入力し、セル C5～F25 に複写する。

垂直照合（\$A5, 生産管理 !\$A\$6 ~ \$M\$19, [ ] g, 0) \* [ ]

- (4) セル C26～F26 には、各列の部品の 1 週間の必要数合計を求める式を入力する。

	A	B	C	D	E	F
1	部品需要表					
2	2013-04-01	から 1 週間				
3						
4	日付	製品コード	部品 1	部品 2	部品 3	部品 4
5	2013-04-01	PA1	380	760	0	0
6	2013-04-01	PA2	260	780	520	0
7	2013-04-01	PA3	0	540	0	810
8	2013-04-02	PA1	340	680	0	0
9	2013-04-02	PA2	180	540	360	0
10	2013-04-02	PA3	0	460	0	690
:	:	:	:	:	:	:
23	2013-04-07	PA1	420	840	0	0
24	2013-04-07	PA2	340	1,020	680	0
25	2013-04-07	PA3	0	620	0	930
26	合計		4,300	13,960	3,400	5,490

図 4 ワークシート“部品需要”の例

*g*に関する解答群

- ア 照合一致(\$B5, 生産管理!\$A\$3～\$M\$3, 0)
- イ 照合一致(\$B5, 生産管理!\$B\$3～\$J\$3, 0) + 3
- ウ 照合一致(\$B5, 部品一覧!\$A\$1～\$A\$4, 0) \* 4
- エ 照合検索(\$B5, \$B\$5～\$B\$7, 生産管理!\$B\$3～\$J\$3)
- オ 照合検索(\$B5, 生産管理!\$A\$3～\$M\$3, 生産管理!\$A\$6～\$M\$6)
- カ 水平照合(\$B5, 生産管理!\$B\$3～\$J\$3, 1, 0)
- キ 水平照合(\$B5, 生産管理!\$A\$3～\$M\$19, 4, 0)

## ■ Java プログラムで使用する API の説明

```
java.util
public interface SortedMap<K, V>
    型 K のキーに型 V の値を対応付けて保持するマップのインター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。マップはキーの昇順に整列される。
```

メソッド

```
public V get(Object key)
    指定されたキーに対応付けられた値を返す。
    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 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。このキーが既に他の値と対応付けられていれば、その値は指定された値に置き換えられる。
    引数： key — キー
           value — 値
    戻り値：指定されたキーに値が対応付けられていた場合は、その値
            このキーと値の対応付けがなければ null
```

```
public boolean containsKey(K key)
    指定されたキーに値が対応付けられている場合に true を返す。
    引数： key — キー
    戻り値：指定されたキーに値が対応付けられていれば true
            それ以外は false
```

```
public SortedMap<K, V> headMap(K toKey)
    保持しているキーと値の対応付けのうち、キーが toKey より小さい全ての対応付けからなるマップを返す。
    引数： toKey — キー
    戻り値：キーが toKey より小さい全ての対応付けからなるマップ
```

```
public K lastKey()
    ソートマップ内に現在ある最後のキーを返す。
    戻り値：ソートマップ内に現在ある最後のキー
```

```
java.util
public class TreeMap<K, V>
    インタフェース SortedMap の実装である。
    メソッドの説明は、インターフェース SortedMap の項を参照
```

コンストラクタ

```
public TreeMap()
    空の TreeMap を作る。マップはキーの自然順序付けに従って整列される。
```

```
java.util
```

```
public interface List<E>
```

リスト（順序付けられたコレクション）のためのインターフェースを提供する。

#### メソッド

```
public boolean add(E e)
```

指定された要素をリストの最後に追加する。

引数： e — リストに追加する要素

戻り値：true

```
public E get(int index)
```

リスト内の指定された位置にある要素を返す。

引数： index — 返される要素のインデックス

戻り値：リスト内の指定された位置にある要素

例外： IndexOutOfBoundsException — indexが範囲外のとき

```
public int indexOf(Object obj)
```

指定された要素がリスト内で最初に検出された位置を返す。

引数： obj — 検索する要素

戻り値：指定された要素がリスト内で最初に検出された位置

リストにこの要素がない場合は -1

```
java.util
```

```
public class ArrayList<E>
```

インターフェース List の配列による実装である。

メソッドの説明は、インターフェース List の項を参照

#### コンストラクタ

```
public ArrayList(int n)
```

指定された大きさの空リストを作る。

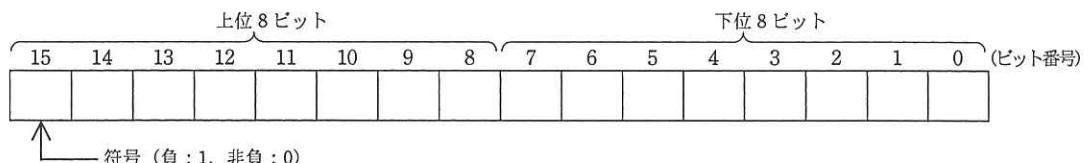
引数： n — リストの大きさ

## ■アセンブラー言語の仕様

### 1. システム COMET II 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 $r_1, r_2$ $r, adr [,x]$	$r_1 \leftarrow (r_2)$ $r \leftarrow (\text{実効アドレス})$	○*1
ストア STore	ST $r, adr [,x]$	$\text{実効アドレス} \leftarrow (r)$	—
ロードアドレス Load Address	LAD $r, adr [,x]$	$r \leftarrow \text{実効アドレス}$	—

## (2) 算術、論理演算命令

算術加算 ADD Arithmetic	ADDA r1, r2 r, adr [,x]	r1 ← (r1) + (r2) r ← (r) + (実効アドレス)	○
論理加算 ADD Logical	ADDL r1, r2 r, adr [,x]	r1 ← (r1) + <sub>L</sub> (r2) r ← (r) + <sub>L</sub> (実効アドレス)	
算術減算 SUBtract Arithmetic	SUBA r1, r2 r, adr [,x]	r1 ← (r1) - (r2) r ← (r) - (実効アドレス)	
論理減算 SUBtract Logical	SUBL r1, r2 r, adr [,x]	r1 ← (r1) - <sub>L</sub> (r2) r ← (r) - <sub>L</sub> (実効アドレス)	
論理積 AND	AND r1, r2 r, adr [,x]	r1 ← (r1) AND (r2) r ← (r) AND (実効アドレス)	
論理和 OR	OR r1, r2 r, adr [,x]	r1 ← (r1) OR (r2) r ← (r) OR (実効アドレス)	
排他的論理和 eXclusive OR	XOR r1, r2 r, adr [,x]	r1 ← (r1) XOR (r2) 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																							
		<table border="1"> <thead> <tr> <th>比較結果</th> <th colspan="2">FRの値</th> </tr> <tr> <th></th> <th>SF</th> <th>ZF</th> </tr> </thead> <tbody> <tr> <td>(r1) &gt; (r2)</td> <td>0</td> <td>0</td> </tr> <tr> <td>(r) &gt; (実効アドレス)</td> <td></td> <td></td> </tr> <tr> <td>(r1) = (r2)</td> <td>0</td> <td>1</td> </tr> <tr> <td>(r) = (実効アドレス)</td> <td></td> <td></td> </tr> <tr> <td>(r1) &lt; (r2)</td> <td>1</td> <td>0</td> </tr> <tr> <td>(r) &lt; (実効アドレス)</td> <td></td> <td></td> </tr> </tbody> </table>		比較結果	FRの値			SF	ZF	(r1) > (r2)	0	0	(r) > (実効アドレス)			(r1) = (r2)	0	1	(r) = (実効アドレス)			(r1) < (r2)	1	0	(r) < (実効アドレス)	
比較結果	FRの値																									
	SF	ZF																								
(r1) > (r2)	0	0																								
(r) > (実効アドレス)																										
(r1) = (r2)	0	1																								
(r) = (実効アドレス)																										
(r1) < (r2)	1	0																								
(r) < (実効アドレス)																										
論理比較 ComPare Logical	CPL r1, r2 r, adr [,x]																									

## (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。	

(6) スタック操作命令

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

(7) コール, リターン命令

コール CALL subroutine	CALL adr [,x]	SP $\leftarrow$ (SP) $-_L$ 1, (SP) $\leftarrow$ (PR), PR $\leftarrow$ 実効アドレス	—
リターン RETurn from subroutine	RET	PR $\leftarrow$ ( (SP) ), SP $\leftarrow$ (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。  
 +<sub>L</sub>, -<sub>L</sub> 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. アセンブラー言語 CASL II の仕様

### 2.1 言語の仕様

- (1) CASL II は、COMET II のためのアセンブラー言語である。
- (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 種類のアセンブラー命令 (START, END, DS, DC) , 4 種類のマクロ命令 (IN, OUT, RPUSH, RPOP) 及び機械語命令 (COMET II 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
アセンブラー命令	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 アセンブラー命令

アセンブラー命令は、アセンブラーの制御などを行う。

- (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)	<b>END</b>	
-----	------------	--

END 命令は、プログラムの終わりを定義する。

(3)	<b>DS</b>	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 ( $\geq 0$ ) 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)	<b>DC</b>	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 進数值を、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 進数值を 1 語の 2 進数データとして格納する ( $0000 \leq h \leq FFFF$ )。
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)	<b>IN</b>	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 ( $\geq 0$ ) 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)	<b>OUT</b>	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 ( $\geq 0$ ) 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) アセンブラーによって生成される命令語や領域の相対位置は、アセンブラー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) アセンブラーは、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、アセンブラー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（基本情報技術者試験用）

表計算ソフトの機能、用語などは、原則として次による。

なお、ワークシートの保存、読み出し、印刷、<sup>まいじゆ</sup>野線作成やグラフ作成など、ここで示す以外の機能などを使用するときには、問題文中に示す。

### 1. ワークシート

- (1) 列と行で構成される升目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、10,000 行とする。
- (2) ワークシートの列と行のそれぞれの位置は、列番号と行番号で表す。列番号は、最左端列の列番号を A とし、A, B, …, Z, AA, AB, …, AZ, BA, BB, …, BZ, …, IU, IV と表す。行番号は、最上端行の行番号を 1 とし、1, 2, …, 10000 と表す。
- (3) 複数のワークシートを利用することができる。このとき、各ワークシートには一意のワークシート名を付けて、他のワークシートと区別する。

### 2. セルとセル範囲

- (1) ワークシートを構成する各升をセルという。その位置は列番号と行番号で表し、それをセル番地といふ。

[例] 列 A 行 1 にあるセルのセル番地は、A1 と表す。

- (2) ワークシート内のある長方形の領域に含まれる全てのセルの集まりを扱う場合、長方形の左上端と右下端のセル番地及び“～”を用いて、“左上端のセル番地～右下端のセル番地”と表す。これを、セル範囲といふ。

[例] 左上端のセル番地が A1 で、右下端のセル番地が B3 のセル範囲は、A1～B3 と表す。

- (3) 他のワークシートのセル番地又はセル範囲を指定する場合には、ワークシート名と“!”を用い、それぞれ“ワークシート名！セル番地”又は“ワークシート名！セル範囲”と表す。

[例] ワークシート“シート1”的セル範囲 B5～G10 を、別のワークシートから指定する場合には、シート1!B5～G10 と表す。

### 3. 値と式

- (1) セルは値をもち、その値はセル番地によって参照できる。値には、数値、文字列、論理値及び空値がある。

- (2) 文字列は一重引用符 “'” で囲って表す。

[例] 文字列 “A”, “BC” は、それぞれ ‘A’, ‘BC’ と表す。

- (3) 論理値の真を true、偽を false と表す。

- (4) 空値を null と表し、空値をもつセルを空白セルという。セルの初期状態は、空白セルとする。

- (5) セルには、式を入力することができる。セルは、式を評価した結果の値をもつ。
- (6) 式は、定数、セル番地、演算子、括弧及び関数から構成される。定数は、数値、文字列、論理値又は空値を表す表記とする。式中のセル番地は、その番地のセルの値を参照する。
- (7) 式には、算術式、文字式及び論理式がある。評価の結果が数値となる式を算術式、文字列となる式を文字式、論理値となる式を論理式という。
- (8) セルに式を入力すると、式は直ちに評価される。式が参照するセルの値が変化したときには、直ちに、適切に再評価される。

#### 4. 演算子

- (1) 単項演算子は、正符号 “+” 及び負符号 “-” とする。
- (2) 算術演算子は、加算 “+”，減算 “-”，乗算 “\*”，除算 “/” 及びべき乗 “^” とする。
- (3) 比較演算子は、より大きい “>”，より小さい “<”，以上 “≥”，以下 “≤”，等しい “=” 及び等しくない “≠” とする。
- (4) 括弧は丸括弧 “( ” 及び “) ” を使う。
- (5) 式中に複数の演算及び括弧があるときの計算の順序は、次表の優先順位に従う。

演算の種類	演算子	優先順位
括弧	( )	高
べき乗演算	^	
単項演算	+ , -	
乗除演算	* , /	
加減演算	+ , -	
比較演算	> , < , ≥ , ≤ , = , ≠	低

#### 5. セルの複写

- (1) セルの値又は式を、他のセルに複写することができる。
- (2) セルを複写する場合で、複写元のセル中にセル番地を含む式が入力されているとき、複写元と複写先のセル番地の差を維持するように、式中のセル番地を変化させるセルの参照方法を相対参照という。この場合、複写先のセルとの列番号の差及び行番号の差を、複写元のセルに入力された式中の各セル番地に加算した式が、複写先のセルに入る。  
 [例] セル A6 に式  $A1 + 5$  が入力されているとき、このセルをセル B8 に複写すると、セル B8 には式  $B3 + 5$  が入る。
- (3) セルを複写する場合で、複写元のセル中にセル番地を含む式が入力されているとき、そのセル番地の列番号と行番号の両方又は片方を変化させないセルの参照方法を絶対参照という。絶対参照を適用する列番号と行番号の両方又は片方の直前には “\$” を付ける。  
 [例] セル B1 に式  $\$A\$1 + \$A2 + A\$5$  が入力されているとき、このセルをセル C4 に複写す

ると、セル C4 には式  $\$A\$1 + \$A5 + B\$5$  が入る。

- (4) セルを複写する場合で、複写元のセル中に、他のワークシートを参照する式が入力されているとき、その参照するワークシートのワークシート名は複写先でも変わらない。

[例] ワークシート“シート2”のセル A6 に式 シート1!A1 が入力されているとき、このセルをワークシート“シート3”のセル B8 に複写すると、セル B8 には式 シート1!B3 が入る。

## 6. 関数

式には次の表で定義する関数を利用することができます。

書式	解説
合計(セル範囲 <sup>1)</sup>	セル範囲に含まれる数値の合計を返す。 [例] 合計(A1～B5)は、セル範囲 A1～B5 に含まれる数値の合計を返す。
平均(セル範囲 <sup>1)</sup>	セル範囲に含まれる数値の平均を返す。
標本標準偏差(セル範囲 <sup>1)</sup>	セル範囲に含まれる数値を標本として計算した標準偏差を返す。
母標準偏差(セル範囲 <sup>1)</sup>	セル範囲に含まれる数値を母集団として計算した標準偏差を返す。
最大(セル範囲 <sup>1)</sup>	セル範囲に含まれる数値の最大値を返す。
最小(セル範囲 <sup>1)</sup>	セル範囲に含まれる数値の最小値を返す。
IF(論理式, 式1, 式2)	論理式の値が true のとき式 1 の値を, false のとき式 2 の値を返す。 [例] IF(B3 > A4, '北海道', C4) は、セル B3 の値がセル A4 の値より大きいとき文字列 “北海道” を、それ以外のときセル C4 の値を返す。
個数(セル範囲)	セル範囲に含まれるセルのうち、空白セルでないセルの個数を返す。
条件付個数(セル範囲, 検索条件の記述)	セル範囲に含まれるセルのうち、検索条件の記述で指定された条件を満たすセルの個数を返す。検索条件の記述は比較演算子と式の組で記述し、セル範囲に含まれる各セルと式の値を、指定した比較演算子によって評価する。 [例1] 条件付個数(H5～L9, > A1) は、セル範囲 H5～L9 のセルのうち、セル A1 の値より大きな値をもつセルの個数を返す。 [例2] 条件付個数(H5～L9, = 'A4') は、セル範囲 H5～L9 のセルのうち、文字列 “A4” をもつセルの個数を返す。
整数部(算術式)	算術式の値以下で最大の整数を返す。 [例1] 整数部(3.9) は、3 を返す。 [例2] 整数部(-3.9) は、-4 を返す。
剰余(算術式1, 算術式2)	算術式1の値を被除数、算術式2の値を除数として除算を行ったときの剰余を返す。関数“剰余”と“整数部”は、剰余(x,y) = x - y * 整数部(x / y)という関係を満たす。 [例1] 剰余(10,3) は、1 を返す。 [例2] 剰余(-10,3) は、2 を返す。
平方根(算術式)	算術式の値の非負の平方根を返す。算術式の値は、非負の数値でなければならない。
論理積(論理式1, 論理式2, …) <sup>2)</sup>	論理式1, 論理式2, … の値が全て true のとき、true を返す。それ以外のとき false を返す。
論理和(論理式1, 論理式2, …) <sup>2)</sup>	論理式1, 論理式2, … の値のうち、少なくとも一つが true のとき、true を返す。それ以外のとき false を返す。
否定(論理式)	論理式の値が true のとき false を、false のとき true を返す。

切上げ（算術式、桁位置）	算術式の値を指定した桁位置で、関数“切上げ”は切り上げた値を、関数“四捨五入”は四捨五入した値を、関数“切捨て”は切り捨てた値を返す。ここで、桁位置は小数第1位の桁を0とし、右方向を正として数えたときの位置とする。 [例1] 切上げ(-314.159, 2) は、-314.16を返す。 [例2] 切上げ(314.159, -2) は、400を返す。 [例3] 切上げ(314.159, 0) は、315を返す。
四捨五入（算術式、桁位置）	
切捨て(算術式、桁位置)	
結合(式1,式2,...) <sup>2)</sup>	式1, 式2, … のそれぞれの値を文字列として扱い、それらを引数の順につないでできる一つの文字列を返す。 [例] 結合('北海道', '九州', 123, 456) は、文字列“北海道九州123456”を返す。
順位(算術式、セル範囲 <sup>1)</sup> 、順序の指定)	セル範囲の中での算術式の値の順位を、順序の指定が0の場合は昇順で、1の場合は降順で数えて、その順位を返す。ここで、セル範囲の中に同じ値がある場合、それらを同順とし、次の順位は同順の個数だけ加算した順位とする。
乱数()	0以上1未満の一様乱数（実数値）を返す。
表引き(セル範囲、行の位置、列の位置)	セル範囲の左上端から行と列をそれぞれ1, 2, … と数え、セル範囲に含まれる行の位置と列の位置で指定した場所にあるセルの値を返す。 [例] 表引き(A3 ~ H11, 2, 5) は、セル E4 の値を返す。
垂直照合(式、セル範囲、列の位置、検索の指定)	セル範囲の左端列を上から下に走査し、検索の指定によって指定される条件を満たすセルが現れる最初の行を探す。その行に対して、セル範囲の左端列から列を1, 2, … と数え、セル範囲に含まれる列の位置で指定した列にあるセルの値を返す。 ・検索の指定が0の場合の条件：式の値と一致する値を検索する。 ・検索の指定が1の場合の条件：式の値以下の最大値を検索する。このとき、左端列は上から順に昇順に整列されている必要がある。 [例] 垂直照合(15, A2 ~ E10, 5, 0) は、セル範囲の左端列をセル A2, A3, …, A10 と探す。このとき、セル A6 で 15 を最初に見つけたとすると、左端列 A から数えて 5 列目の列 E 中で、セル A6 と同じ行にあるセル E6 の値を返す。
水平照合(式、セル範囲、行の位置、検索の指定)	セル範囲の上端行を左から右に走査し、検索の指定によって指定される条件を満たすセルが現れる最初の列を探す。その列に対して、セル範囲の上端行から行を1, 2, … と数え、セル範囲に含まれる行の位置で指定した行にあるセルの値を返す。 ・検索の指定が0の場合の条件：式の値と一致する値を検索する。 ・検索の指定が1の場合の条件：式の値以下の最大値を検索する。このとき、上端行は左から順に昇順に整列されている必要がある。 [例] 水平照合(15, A2 ~ G6, 5, 1) は、セル範囲の上端行をセル A2, B2, …, G2 と探す。このとき、15以下の最大値をセル D2 で最初に見つけたとすると、上端行 2 から数えて 5 行目の行 6 中で、セル D2 と同じ列にあるセル D6 の値を返す。
照合検索(式、検索のセル範囲、抽出のセル範囲)	1行又は1列を対象とする同じ大きさの検索のセル範囲と抽出のセル範囲に対して、検索のセル範囲を左端又は上端から走査し、式の値と一致する最初のセルを探す。見つかったセルの検索のセル範囲の中での位置と、抽出のセル範囲の中での位置が同じセルの値を返す。 [例] 照合検索(15, A1 ~ A8, C6 ~ C13) は、セル範囲 A1 ~ A8 をセル A1, A2, … と探す。このとき、セル A5 で 15 を最初に見つけたとすると、セル範囲 C6 ~ C13 の上端から数えて 5 番目のセル C10 の値を返す。

照合一致(式, セル範囲, 検索の指定)	<p>1行又は1列を対象とするセル範囲に対して、セル範囲の左端又は上端から走査し、検索の指定によって指定される条件を満たす最初のセルを探す。見つかったセルの位置を、セル範囲の左端又は上端から1, 2, …と数えた値とし、その値を返す。</p> <ul style="list-style-type: none"> <li>・検索の指定が0の場合の条件：式の値と一致する値を検索する。</li> <li>・検索の指定が1の場合の条件：式の値以下の最大値を検索する。このとき、セル範囲は左端又は上端から順に昇順に整列されている必要がある。</li> <li>・検索の指定が-1の場合の条件：式の値以上の最小値を検索する。このとき、セル範囲は左端又は上端から順に降順に整列されている必要がある。</li> </ul> <p>[例] 照合一致(15, B2 ~ B12, -1)は、セル範囲B2 ~ B12をセルB2, B3, …と探す。このとき、15以上の最小値をセルB9で最初に見つけたとすると、セルB2から数えた値8を返す。</p>
条件付合計(検索のセル範囲, 検索条件の記述, 合計のセル範囲 <sup>1)</sup> )	<p>行数及び列数が共に同じ検索のセル範囲と合計のセル範囲に対して、検索と合計を行う。検索のセル範囲に含まれるセルのうち、検索条件の記述で指定される条件を満たすセルを全て探す。検索条件の記述を満たした各セルについての左上端からの位置と、合計のセル範囲中で同じ位置にある各セルの値を合計して返す。</p> <p>検索条件の記述は比較演算子と式の組で記述し、検索のセル範囲に含まれる各セルと式の値を、指定した比較演算子によって評価する。</p> <p>[例1] 条件付合計(A1 ~ B8, &gt; E1, C2 ~ D9)は、検索のセル範囲であるA1 ~ B8のうち、セルE1の値より大きな値をもつ全てのセルを探す。このとき、セルA2, B4, B7が見つかったとすると、合計のセル範囲であるC2 ~ D9の左上端からの位置が同じであるセルC3, D5, D8の値を合計して返す。</p> <p>[例2] 条件付合計(A1 ~ B8, = 160, C2 ~ D9)は、検索のセル範囲であるA1 ~ B8のうち、160と一致する値をもつ全てのセルを探す。このとき、セルA2, B4, B7が見つかったとすると、合計のセル範囲であるC2 ~ D9の左上端からの位置が同じであるセルC3, D5, D8の値を合計して返す。</p>

注<sup>1)</sup> 引数として渡したセル範囲の中で、数値以外の値は処理の対象としない。

2) 引数として渡すことができる式の個数は、1以上である。

## 7. マクロ

### (1) ワークシートとマクロ

ワークシートには複数のマクロを格納することができる。

マクロは一意のマクロ名を付けて宣言する。マクロの実行は、表計算ソフトのマクロの実行機能を使って行う。

[例] ○マクロ : Pro

例は、マクロProの宣言である。

### (2) 変数とセル変数

変数の型には、数値型、文字列型及び論理型があり、変数は宣言することで使用できる。変数名にセル番地を使用することはできない。

[例] ○数値型 : row, col

例は、数値型の変数row, colの宣言である。

セルを変数として使用でき、これをセル変数という。セル変数は、宣言せずに使用できる。

セル変数の表現方法には、絶対表現と相対表現とがある。

セル変数の絶対表現は、セル番地で表す。

セル変数の相対表現は、次の書式で表す。

書式	解説
相対(セル変数, 行の位置, 列の位置)	セル変数で指定したセルを基準のセルとする。そのセルの行番号と列番号の位置を 0 とし、下又は右方向を正として数え、行の位置と列の位置で指定した数と一致する場所にあるセルを表す変数である。

[例1] 相対(B5, 2, 3) は、セル E7 を表す変数である。

[例2] 相対(B5, -2, -1) は、セル A3 を表す変数である。

### (3) 配列

数値型、文字列型又は論理型の配列は宣言することで使用できる。添字を “[ ” 及び “ ] ” で囲み、添字が複数ある場合はコンマで区切る。添字は 0 から始まる。

なお、数値型及び文字列型の変数及び配列の要素には、空値を格納することができる。

[例] ○文字列型 : table[100, 200]

例は、 $100 \times 200$  個の文字列型の要素をもつ 2 次元配列 table の宣言である。

### (4) 宣言、注釈及び処理

宣言、注釈及び処理の記述は、“共通に使用される擬似言語の記述形式” に従う。

処理の記述中に式又は関数を使用する場合、その記述中に変数、セル変数又は配列の要素が使用できる。

[例] ○数値型 : row

```
row : 0, row < 5, 1
    • 相対(B5, row, 0) ← 順位(相対(C5, row, 0), G5~G9, 0)
```

例は、セル C5, C6, …, C9 の各値に対して、セル範囲 G5 ~ G9 の中の順位を調べ、その順位をセル B5, B6, …, B9 に順に代入する。

[ メモ用紙 ]

[ メモ用紙 ]

[ メモ用紙 ]

6. 退室可能時間に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
--------	---------------

7. 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。
8. 問題冊子の余白などは、適宜利用して構いません。
9. Java プログラムで使用する API の説明、アセンブラー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。
10. 試験時間中、机上に置けるものは、次のものに限ります。  
なお、会場での貸出しありません。  
受験票、黒鉛筆及びシャープペンシル（B又はHB）、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、ポケットティッシュ、目薬これら以外は机上に置けません。使用もできません。
11. 試験終了後、この問題冊子は持ち帰ることができます。
12. 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。
13. 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。

試験問題に記載されている会社名又は製品名は、それぞれ各社の商標又は登録商標です。

なお、試験問題では、™ 及び ® を明記していません。