

# 平成 23 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

- 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。
- 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。
- 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マークされていない場合は、読み取れず、採点されないことがありますので、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、消しきずを残さないでください。
  - 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。
  - 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。
  - 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。
- 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)	(ウ)	(エ)
----	---	-----	-----	-----	-----

裏表紙の注意事項も、  
必ず読んでください。

[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	[●]				

## 〔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	A/D 変換
問 2	データベース	従業員データベースの設計と運用
問 3	ネットワーク	ネットワークの構築
問 4	情報セキュリティ	情報セキュリティにおけるリスク
問 5	ソフトウェア設計	書籍の卸売業者の受注システム
問 6	プロジェクトマネジメント	プロジェクトの要員計画
問 7	システム戦略	販売ルート別の売上及び市場の分析

### ●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	代入文の処理

### ●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)	循環小数の出力
問 10	ソフトウェア開発 (COBOL)	受験申込ファイルの集計及び整列
問 11	ソフトウェア開発 (Java)	住所録管理プログラム
問 12	ソフトウェア開発 (アセンブラー)	除算と 2 進 10 進数文字列変換
問 13	ソフトウェア開発 (表計算)	社員の基本給及び賞与計算

## 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 A/D変換に関する次の記述を読んで、設問1~3に答えよ。

A/D変換とは、アナログ信号をデジタル信号に変換することであり、標本化、量子化、符号化の3段階で処理する。直流の電圧を例にnビットのA/D変換を説明する。

### (1) 標本化

標本化では、時間的に連続したアナログ信号である電圧を一定の時間間隔で測定する。図1では、時間軸を $t_0, t_1, \dots$ と等間隔 $d$ で区切り、各時刻での電圧を $v(t_0), v(t_1), \dots$ と表す。

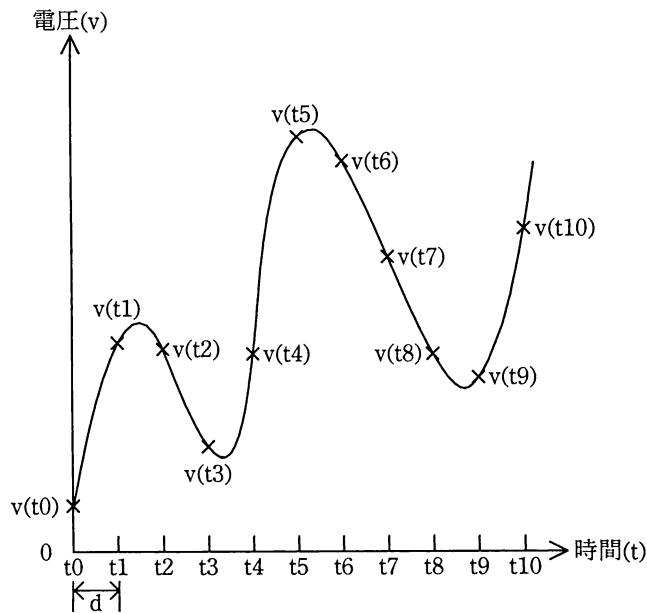


図1 標本化の例

## (2) 量子化

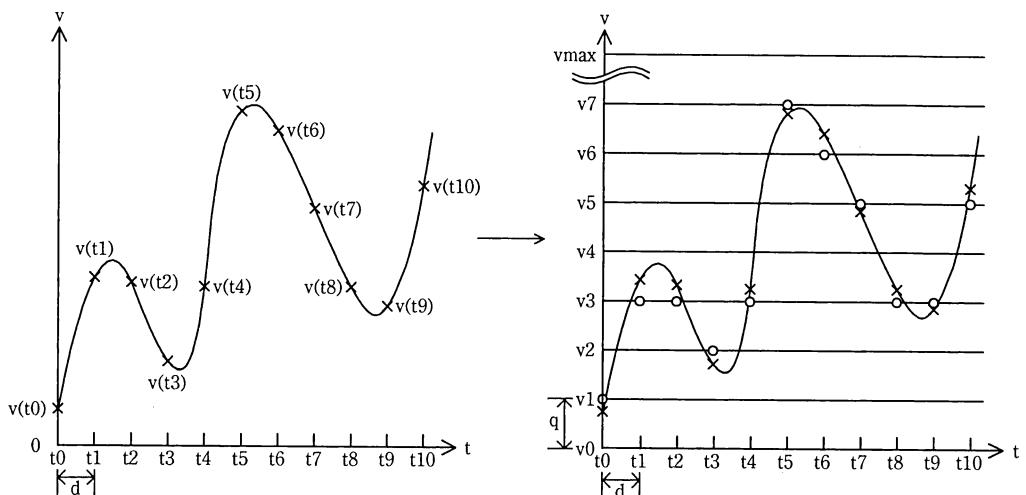
量子化では、(1)で標本化して得られた電圧  $v(t_0), v(t_1), \dots$  を刻み幅  $q$  の整数倍の値で近似する。

量子化を行う場合、まず測定する電圧の最大幅 FSR を決める。次に、 $n$  ビットで量子化するために、FSR を  $(2^n - 1)$  等分して刻み幅  $q$  を求める。このとき、刻み幅  $q$  は  $q = \text{FSR} / (2^n - 1)$  となる。

量子化において、近似値として用いられる電圧の値を、小さい方から順に  $v_0 = 0, v_1 = q, \dots, v_{\max} = (2^n - 1)q$  としたとき、時刻  $t_m$  において標本化で測定されたアナログ信号の電圧  $v(t_m)$  について、次の条件を満たすような  $N$  ( $N$  は 0 以上の整数) を見つけ、電圧  $N \times q$  を電圧  $v(t_m)$  の測定値とする。これを  $n$  ビット量子化という。

$$N \times q - q / 2 \leq v(t_m) < N \times q + q / 2$$

すなわち、図 2 右のように電圧軸を刻み幅  $q$  で  $v_0, v_1, \dots, v_{\max}$  の電圧に分割しておき、 $v(t_0), v(t_1), \dots$  のそれぞれについて、 $v_0 \sim v_{\max}$  のうちで最も近い電圧を測定値とする。例えば、 $v(t_3)$  の測定値は  $v_2$  となる。



注記 “○” は  $v(t_m)$  を  $q$  の整数倍で近似した値

図 2 量子化の例

## (3) 符号化

符号化では、(2)の量子化で用いた電圧  $v_0, v_1, \dots, v_{\max}$  に 2 進符号を対応付ける。この符号によって、各測定値を表す。

設問 1 図 2 左の電圧  $v(t0), v(t1), \dots, v(t10)$ だけの符号化を考える。図 2 右の電圧  $v0, v1, \dots, v7$  を 2 進符号 000, 001, …, 111 に順に対応付けた場合を表 1 に示す。

表 1 電圧と符号の対応

電圧(v)	符号
$v0$	000
$v1$	001
$v2$	010
:	:
$v7$	111

図 2 左の  $v(t0), v(t1), \dots, v(t10)$  の各測定値を、表 1 に基づいて符号化すると表 2 のようになる。表 2 中の [ ] に入る正しい答えを、解答群の中から選べ。

表 2 各時刻における測定値の符号化

時刻	t0	t1	t2	t3	t4	t5	t6	t7	t8	t9	t10
符号	001	011	011	010	[ ]	[ ]	a	[ ]	[ ]	[ ]	b

注記 網掛けの部分は表示していない。

### 解答群

ア 011 イ 100 ウ 101 エ 110 オ 111

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。

アナログ信号の電圧の範囲が 0~9V であるとき、FSR を 9V とし、4 ビットで量子化した場合、 $q$  は [ ] V である。アナログ信号の電圧 7.49…V の測定値は [ ] V となり、表 1 の場合と同様に 2 進符号 0000, 0001, …, 1111 に順に対応付けて符号化すると [ ] となる。

### c, d に関する解答群

ア 0.5625 イ 0.6 ウ 1.2 エ 2.25 オ 5.5  
カ 7.0 キ 7.2 ク 7.5 ケ 7.8 コ 8.0

e に関する解答群

ア 1010 イ 1011 ウ 1100 エ 1101 オ 1110

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。

FSR が 1,022 ミリ V であるアナログ信号の電圧を、50 ミリ秒間隔で 5 秒間標本化した。このとき、A/D 変換後の総データ量を 1,000 ビット以内に納めることができるとみ幅  $q$  の最小値は  f  ミリ V である。

解答群

ア 0.1 イ 0.5 ウ 1.0 エ 1.5 オ 2.0  
カ 2.5 キ 3.0

問 2 従業員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。

C 社は、2011 年 4 月 1 日の組織編成の変更に伴い、従業員データベースの再構築を行った。組織編成の変更前は図 1 に示すとおり、部だけで編成されていたが、事業の拡大及び従業員数の増加に合わせて、図 2 に示すとおり、部と課からなる組織編成となった。

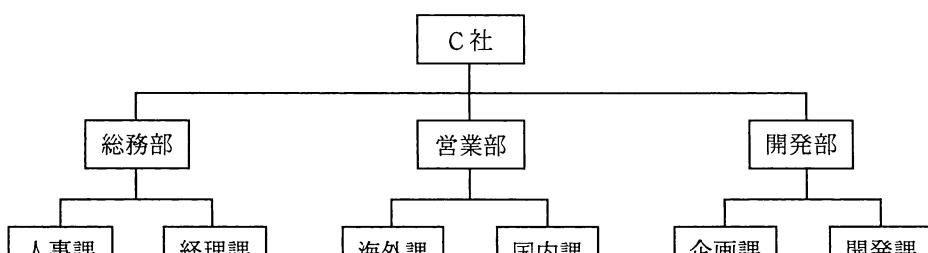
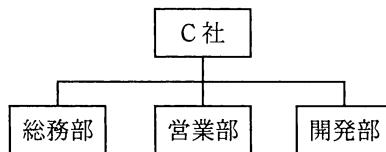


図 2 変更後の組織編成

設問 1 組織編成の変更を反映するために、図 3 に示す表中の部に関する情報の変更について、A 案と B 案を考えて比較検討した。図 4 に示す A 案では、部名と課名の組合せに対して一意の部署コードを割り当てた。図 5 に示す B 案では、部名と課名のそれぞれにコードを割り当て、従業員表の部コードを課コードに変更した。次の記述中の [ ] に入る適切な答えを、解答群の中から選べ。

部表	
部コード	部名

従業員表							
従業員番号	氏名	部コード	内線	入社年月日	住所	自宅電話	年齢

図 3 変更前の従業員データベースの表構成

部署表

部署コード	部署名
D001	総務部人事課

従業員表

従業員番号	氏名	部署コード	内線	入社年月日	住所	自宅電話	年齢
2005012	情報太郎	D001	211	20020401	東京都…	03-123…	31

図 4 A 案の表構成とデータの格納例

部表

部コード	部名
D001	総務部

課表

課コード	課名	部コード
S001	人事課	D001

従業員表

従業員番号	氏名	課コード	内線	入社年月日	住所	自宅電話	年齢
2005012	情報太郎	S001	211	20020401	東京都…	03-123…	31

図 5 B 案の表構成とデータの格納例

最初は、部名と課名の組合せに対して一意の部署コードを割り当てた、A案によって管理しようとした。しかし、これでは、a を変更する必要が生じた場合に複数行を修正する必要があるので、正規化におけるb の観点から好ましくない。また、例えばc を表示する際にLIKE述語を使用したデータ依存の検索が必要になるなど、柔軟性が低いことが分かった。このため、B案の構成でデータベースを再構築した。

#### aに関する解答群

ア 課名

イ 氏名

ウ 表名

エ 部名

b に関する解答群

ア 関係喪失 イ 検索性能 ウ 事前登録 エ 重複更新

c に関する解答群

ア ある課に属する従業員の氏名の一覧 イ ある部に属する従業員の氏名の一覧  
ウ 従業員の氏名の一覧 エ 部署名の一覧

設問2 B 案の構成でデータベースを再構築した後に、課ごとの平均年齢を算出し、表示する。次の SQL 文の [ ] に入る正しい答えを、解答群の中から選べ。

```
SELECT 課表.課コード, 課表.課名, AVG(従業員表.年齢)
  FROM 課表, 従業員表
 WHERE [ ]
```

解答群

- ア 課表.課コード = 従業員表.課コード  
GROUP BY 課表.課コード, 課表.課名
- イ 課表.課コード = 従業員表.課コード  
GROUP BY 課表.部コード, 課表.課名
- ウ 従業員表.年齢 = ANY (SELECT COUNT(従業員表.年齢) FROM 従業員表)
- エ 従業員表.年齢 = ANY  
(SELECT COUNT(従業員表.年齢) FROM 従業員表 GROUP BY 課表.課コード)

設問3 従業員表は、受発注情報を管理する表などから、従業員番号を外部キーとして参照される。このため、従来は特に利用を制限せずに社内公開していたが、個人情報保護の観点から、必要最小限の情報だけを公開するビューを作成することにした。ビューで公開する項目は、従業員番号、氏名、課コード、内線とする。次の SQL 文の [ ] に入る正しい答えを、解答群の中から選べ。

```
CREATE VIEW 従業員公開表 AS [ ]
```

## 解答群

- ア ALTER TABLE 従業員表  
ADD (従業員番号, 氏名, 課コード, 内線)
- イ ALTER TABLE 従業員表  
DROP 入社年月日, 住所, 自宅電話, 年齢
- ウ SELECT \* FROM 従業員表 WHERE 従業員番号 IS NOT NULL
- エ SELECT 従業員番号, 氏名, 課コード, 内線 FROM 従業員表

設問 4 設問 3 で作成したビューと図 6 に示す受注表を使用して、営業部海外課に在籍する従業員が、2011 年 7 月 1 日から 2011 年 9 月 30 日の期間中に受注した案件の受注総額を算出する。営業部海外課の課コードは “S101” で、2011 年 7 月 1 日以降の従業員の異動はない。次の SQL 文の [ ] に入る正しい答えを、解答群の中から選べ。

受注表

伝票番号	受注日	従業員番号	顧客コード	受注額	納品日
------	-----	-------	-------	-----	-----

図 6 受注表の構成

```
SELECT SUM(受注表.受注額)
FROM 受注表, 従業員公開表
WHERE 従業員公開表.課コード = 'S101' AND
[ ]
```

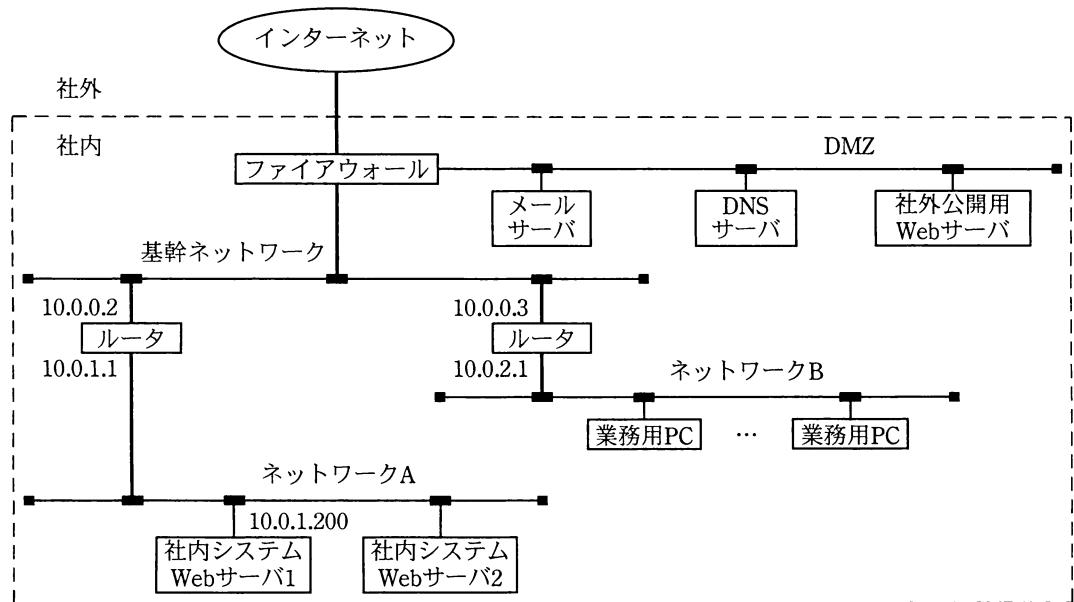
## 解答群

- ア 受注表.従業員番号 = 従業員公開表.従業員番号 AND  
受注表.受注日 BETWEEN '20110701' AND '20110930'
- イ 受注表.受注日 BETWEEN '20110701' AND '20110930'
- ウ 受注表.受注日 IN (SELECT COUNT(\*) FROM 受注表
WHERE 受注表.受注日 BETWEEN '20110701' AND '20110930')
- エ 受注表.受注日 IN (SELECT SUM(受注表.受注額) FROM 受注表
WHERE 受注表.受注日 BETWEEN '20110701' AND '20110930')

問3 ネットワークの構築に関する次の記述を読んで、設問1、2に答えよ。

D社の現在のネットワーク構成を図1に示す。DMZにはメールサーバ、DNSサーバ及び社外公開用Webサーバを接続しており、ネットワークAには社内システムを稼働させるWebサーバを、ネットワークBには社員が通常業務を行うための業務用PCを接続している。

ファイアウォールはインターネットから基幹ネットワークへ向けた通信と基幹ネットワークからインターネットに向けた通信を全て遮断している。したがって、業務用PCから社内にある社外公開用Webサーバや社内システムWebサーバへはアクセスできるが、社外のWebサーバへはアクセスできない。



注記 数字は各ルータ及び社内システム Webサーバ1のそれぞれのネットワークでのIPアドレスである。

図1 D社の現在のネットワーク構成

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。

D 社の各ネットワークに接続された機器の IP アドレスから、ネットワーク A のサブネットマスクは [ a ] であることが分かる。ネットワーク A のネットワークアドレスとサブネットマスクを考慮すると、次に示す IP アドレスのうち、社内システム Web サーバ 2 に設定可能なものは、[ b ] 個ある。

[IP アドレス]

10.0.0.2	10.0.0.3	10.0.0.4	10.0.1.1	10.0.1.2
10.0.1.3	10.0.2.1	10.0.2.2	10.0.2.3	10.0.2.4

a に関する解答群

- |                 |                   |
|-----------------|-------------------|
| ア 255.0.0.0     | イ 255.255.0.0     |
| ウ 255.255.255.0 | エ 255.255.255.128 |

b に関する解答群

- |     |     |     |     |
|-----|-----|-----|-----|
| ア 1 | イ 2 | ウ 3 | エ 4 |
| オ 5 | カ 6 | キ 7 | ク 8 |

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。

D 社は、業務用 PC に IP アドレスなどのネットワークの情報を設定するために、DHCP を利用することにした。DHCP を利用する PC は、DHCP サーバを見つけるためのメッセージをブロードキャストする。D 社は DHCP のメッセージを中継する装置は設置しないので、PC からのメッセージを受信するために、DHCP サーバは [ c ] に設置する必要がある。

さらに、業務用 PC から社外の Web サーバへアクセスするためにプロキシサーバを設置することにした。プロキシサーバはクライアントからの要求に基づき、クライアントの代わりに Web サーバにアクセスし、Web サーバからの応答をクライアントに転送する。インターネットと基幹ネットワーク間の直接の通信は遮

断したままにしておきたいので、プロキシサーバは  d に設置する。

設置するプロキシサーバはキャッシュサーバの機能を備えている。キャッシュサーバは、クライアントから要求された Web ページや画像などが、既にキャッシュに格納されていれば（キャッシュにヒットすれば）、Web サーバに改めてアクセスせずにキャッシュに格納されている内容をクライアントに送るので、応答時間の短縮が見込める。しかし、キャッシュにヒットしなければ Web サーバにアクセスし、Web サーバからの応答をクライアントに転送するとともに、内容をキャッシュに格納するのでオーバヘッドが生じる。

キャッシュサーバを利用しないときの平均応答時間を 100 としたときに、キャッシュサーバ利用時の平均応答時間がキャッシュにヒットしたときで 30、ヒットしなかったときで 110 だとする。このとき、キャッシュのヒット率が  e % 以上であれば、キャッシュサーバ利用時の平均応答時間はキャッシュサーバを利用しないときの平均応答時間の半分以下になる。

#### c, d に関する解答群

- |            |            |
|------------|------------|
| ア DMZ      | イ 基幹ネットワーク |
| ウ ネットワーク A | エ ネットワーク B |

#### e に関する解答群

- |      |      |      |      |
|------|------|------|------|
| ア 50 | イ 55 | ウ 60 | エ 65 |
| オ 70 | カ 75 | キ 80 | ク 85 |

問4 情報セキュリティにおけるリスクに関する次の記述を読んで、設問1、2に答えよ。

E君の所属するF社では、自社の情報セキュリティにおけるリスクを数値化して管理することになり、基準を設定して所有する情報資産のリスク評価を行うことになった。E君はこのうち、サーバX及びサーバYのリスク評価を担当した。

なお、ここでは、試行のための仮の基準と値を扱うが、それぞれに“仮”を表す文言は用いない。

[リスクの値の算出]

F社では、機密性、完全性、可用性のそれぞれについて、情報資産のリスクの値を、次の式で算出する。

$$\text{リスクの値} = \text{情報資産の価値} \times \text{脅威} \times \text{脆弱性}$$

[情報資産の価値の評価基準]

F社では、機密性、完全性、可用性のそれから見た情報資産の価値の評価基準と値を、表1～3のとおりに設定した。

表1 機密性の評価基準と値

評価基準	値
社外に開示できる。	1
社内だけに開示できる。	2
部門内だけに開示できる。	3
必要最小限の関係者だけに開示できる。	4

表2 完全性の評価基準と値

評価基準	値
情報の完全性が失われても、業務への影響はない。	1
情報の完全性が失われても、業務への影響は小さい。	2
情報の完全性が失われると、業務への影響は大きい。	3

表3 可用性の評価基準と値

評価基準	値
定期メンテナンス以外で年間24時間までの利用停止は容認される。	1
定期メンテナンス以外で年間5時間までの利用停止は容認される。	2
定期メンテナンス以外で年間1時間までの利用停止は容認される。	3
定期メンテナンス以外で年間10分までの利用停止は容認される。	4
定期メンテナンス以外で年間1分までの利用停止は容認される。	5

[脅威と脆弱性の判断基準]

F 社では、脅威と脆弱性の判断基準と値を、それぞれ表 4, 5 のおりに設定した。

表 4 脅威の判断基準と値

判断基準	値
発生の可能性が低い。	1
発生の可能性が中程度である。	2
発生の可能性が高い。	3

表 5 脆弱性の判断基準と値

判断基準	値
適切な管理と対策がなされている。	1
ある程度の管理と対策がなされている。	2
管理と対策が不十分である。	3

[サーバ X 及びサーバ Y]

サーバ X では、調達先一般情報のデータベースが稼働している。調達先一般情報とは、調達先コード、調達先の正式な名称、略称、住所、電話番号などである。

サーバ Y では、取引情報のデータベースが稼働している。取引情報とは、調達先コード、購入品コード、単価、購入履歴などである。

E 君は、サーバ X 及びサーバ Y の機密性、完全性、可用性のそれから見た価値を評価するために、調達先一般情報と取引情報に関して、社内関連部門から聴取し、その内容を次のようにまとめた。

[社内関連部門からの聴取内容]

(1) 調達先一般情報

- ① 電話帳や各社の Web ページで公開されている情報であるが、取引があることを F 社の競合他社に知られたくない調達先もあるので、社外には公開できない。
- ② この情報は、調達先との間で行っている EDI (電子データ交換) では利用していないので、誤りがあっても調達業務に与える影響は小さい。
- ③ 社員が、電話番号の確認や、挨拶状の宛先ラベルの印字に利用しているが、サーバ X が利用できない場合には、代替手段での入手が可能である。

(2) 取引情報

- ① 競合する調達先をはじめ、F 社の同業他社に知られてはならない情報である。また、社内でも、他部門には開示できない情報である。

- ② 情報に誤りがあれば、調達や支払などの業務に与える影響は大きい。
- ③ 営業時間内の調達オンライン入力処理、及び夜間のバッチ処理で利用されており、これらを処理するシステムは、メンテナンス以外では、年間4時間以上停止することは許されない。

[脅威と脆弱性の状況]

E君は、各サーバがさらされている脅威とその脅威に対するF社の脆弱性を調査し、表4及び表5の判断基準に基づいて評価した。そのうちの主なものを、表6に示す。

表6 サーバX及びサーバYの主な脅威と脆弱性の値

脅威		脆弱性	
種類	値	種類	値
ウイルス感染	3	ウイルス対策ソフト未導入	3
不正アクセス	3	アクセスコントロールの不備	2
故障	2	メンテナンス不足	3
なりすまし	2	パスワード管理の不備	2
盗聴	2	最新推奨暗号の未使用	1

[受容可能なリスク水準]

F社では、受容可能なリスク水準を、表7のとおりに設定した。情報資産について各リスクの値がこれらの値以下であれば、そのリスクを保有し、そうでなければ、リスク対応を行う。

表7 受容可能なリスク水準

機密性	13
完全性	15
可用性	10

[サーバX及びサーバYのリスク評価]

表1～5の基準、聴取内容及びサーバXとサーバYの状況から、E君は、サーバX及びサーバYに関するリスク評価を行った。F社では、評価に当たって、表1～3の

評価基準では、該当する基準の値のうちで最も小さいものを選ぶことしている。

評価結果の一部を表 8 に示す。

表 8 サーバ X 及びサーバ Y のリスク評価（抜粋）

情報資産		脅威		脆弱性		リスク
名称	価値		内容	値	内容	値
	分類	値				
サーバ X	機密性		⋮			
			なりすまし		パスワード管理の不備	a
			⋮			
			ウイルス感染		ウイルス対策ソフト未導入	18
			不正アクセス		アクセスコントロールの不備	12
	完全性		なりすまし		パスワード管理の不備	8
			⋮			
			⋮			
	可用性		⋮			
サーバ Y	機密性		⋮			
			不正アクセス		アクセスコントロールの不備	b
			⋮			
	完全性		ウイルス感染		ウイルス対策ソフト未導入	c
			⋮			
	可用性	d	⋮			

注記 網掛けの部分は表示していない。“…”は表示の省略を示している。

設問 1 表 8 中の  a ~  d 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 4	イ 6	ウ 8	エ 9	オ 12
カ 16	キ 18	ク 24	ケ 27	コ 36

d に関する解答群

ア 1	イ 2	ウ 3	エ 4	オ 5
-----	-----	-----	-----	-----

**設問2** 表8のサーバXの完全性の破線で囲まれた部分に関し、F社の受容可能なリスク水準から判断されるリスク対応として適切なものを、解答群の中から選べ。

解答群

- ア IDS（侵入検知システム）を導入する。 イ ウイルス対策ソフトを導入する。
- ウ 公開鍵暗号を利用する。 エ 定期メンテナンスの回数を増やす。
- オ パスワード管理を強化する。

問 5 書籍の卸売業者の受注システムに関する次の記述を読んで、設問 1, 2 に答えよ。

書籍の卸売業者の受注システム開発のために、オブジェクトを抽出し、その関係を示すオブジェクト図を作成し、それを基にクラス図を作成することにした。

[受注業務の説明]

- (1) 得意先から注文を受けると、受注伝票に受注日、得意先と注文された商品の商品名などを記入する。受注伝票の例を図 1 に示す。

受注伝票					
受注番号		12345	得意先	P書店	受注日
No	商品番号	商品名	単価	数量	小計
1	5001	UML入門	2,000	2	4,000
2	5011	XML（上）	2,500	1	2,500
3	6001	XMLセット	5,940	1	5,940
4					
:	:	:	:	:	:
10					
合計					12,440

図 1 受注伝票の例

- (2) 商品には、単独商品とセット商品がある。セット商品は、2 種類以上の単独商品を組み合せたものであり、セット商品としての商品名をもつ。セット商品の価格は、そのセットを構成する単独商品の価格の合計から 1% 割り引かれる。一度の注文では、10 種類を超える商品は注文できない。

受注業務のシステム化に当たり、システム化対象となる主なオブジェクトとその主要な属性を抽出し、複数のオブジェクト図を作成した。その一つを図 2 に示す。

図 2 は、P 書店から “UML 入門” を 2 冊、 “XML（上）” を 1 冊、 “XML（上）” と “XML（下）” のセット商品 “XML セット” を 1 セット受注したときのオブジェクト図である。

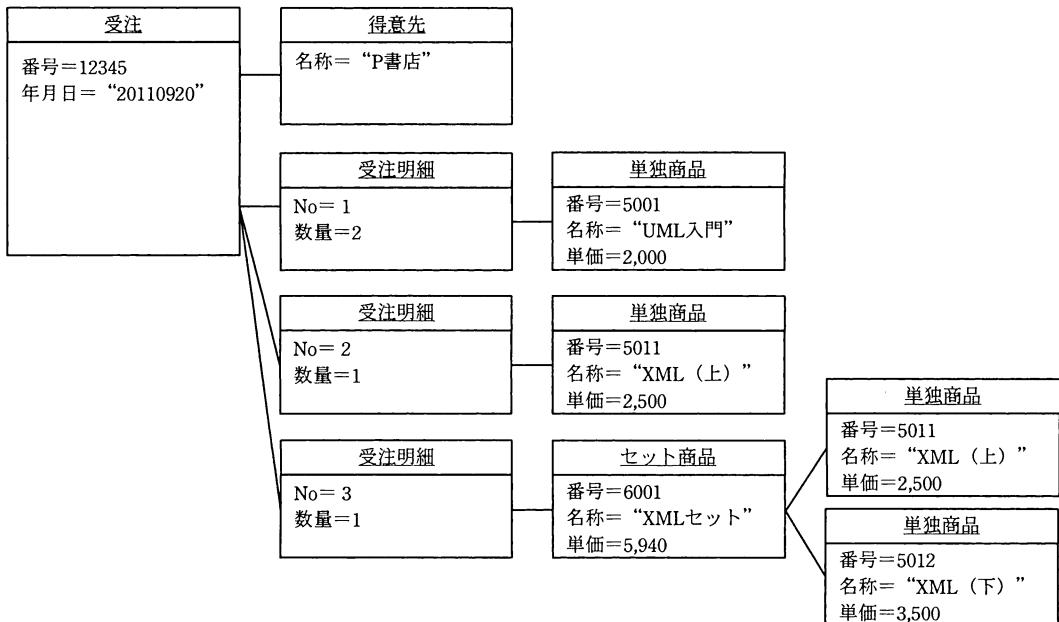


図2 オブジェクト図

次に、クラス抽出と多重度を検討して、図3に示す分析のためのクラス図を作成した。多重度とは、一方のクラスの一つのオブジェクトに対して接続されている、他方のクラスのオブジェクトの個数を示すものである。

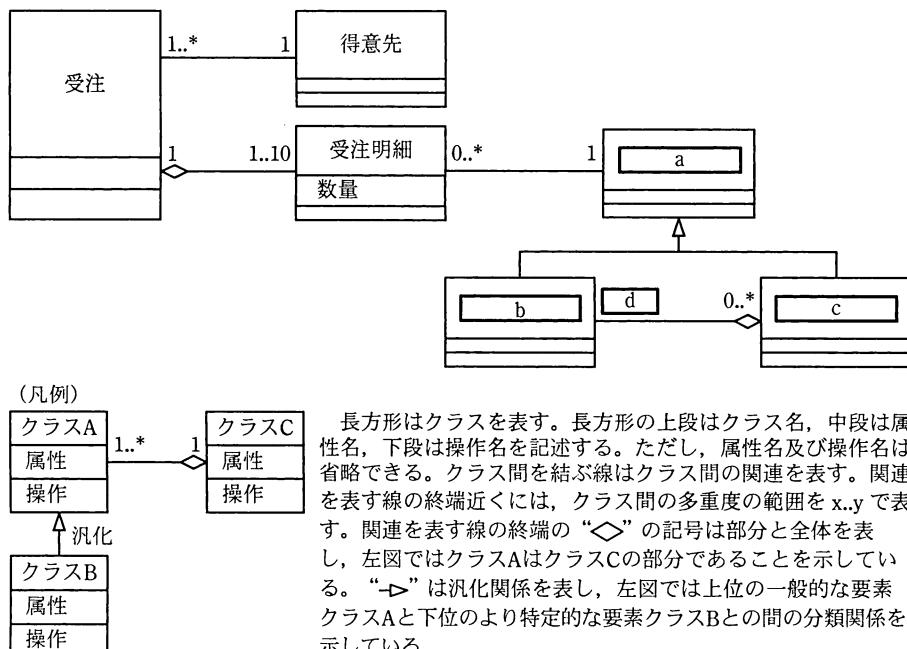


図3 分析のためのクラス図

設問 1 図 3 の [ ] に入る正しい答えを、解答群の中から選べ。

a～c に関する解答群

ア 商品

イ 商品番号

ウ 受注者

エ 受注伝票

オ セット商品

カ 単独商品

d に関する解答群

ア 0

イ 0..\*

ウ 1

エ 1..10

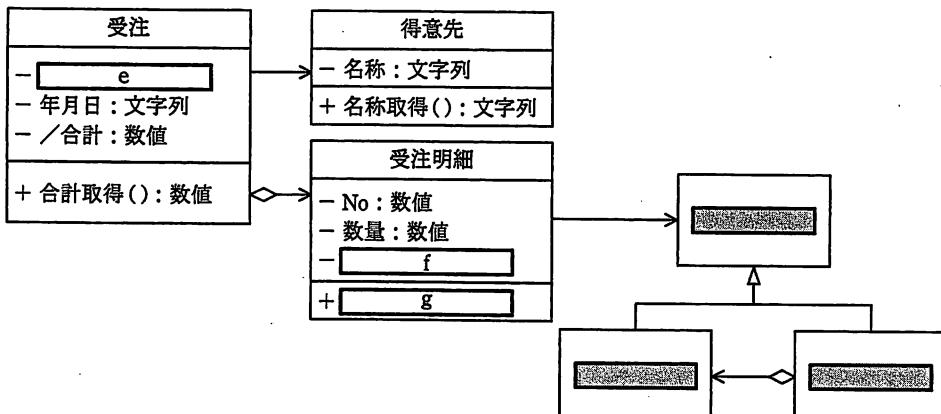
オ 1..\*

カ 2

キ 2..\*

設問 2 図 3 に基づいて、各クラスに必要な属性と操作を検討した。受注には年月日、合計などを保持し、受注明細には No, 数量などを保持することにした。また、受注の合計は、受注明細ごとの金額を求めてから算出することにした。

全てのクラスの属性と操作を検討した結果から、図 4 に示す設計のためのクラス図を作成した。図 4 中の [ ] に入る正しい答えを、解答群の中から選べ。



注記1 網掛け部分は表示しない。

注記2 多重度は省略する。

(凡例)

クラスA
-属性
-／属性
+操作

属性名の前にある “／” は、派生要素であることを示している。すなわち、この属性の値は他の属性から計算できる。  
属性と操作の前にある “+” は、全てのクラスから参照可能であることを示し、“-” は自分自身のクラスからだけ参照可能であることを示している。

図 4 設計のためのクラス図

e に関する解答群

ア 商品番号：文字列	イ 商品名：文字列	ウ セット商品名：文字列
エ 番号：数値	オ 名称：文字列	

f に関する解答群

ア ／小計：数値	イ 商品名：文字列	ウ 年月日：文字列
エ 名称：文字列		

g に関する解答群

ア 受注日取得()：文字列	イ 小計取得()：数値	ウ 数量取得()：数値
エ 単価取得()：数値	オ 名称取得()：文字列	

問6 プロジェクトの要員計画に関する次の記述を読んで、設問1～3に答えよ。

製造会社のA社では、社内の情報システムを再構築することになった。新システムは、将来の保守性を考慮して、既存システムの改修による機能強化ではなく、全て新規に開発する。開発が大規模になることから、要員を確保しやすいよう、2期（第一期と第二期）に分けて開発することになった。新システムの開発は、A社の要員と、システムの保守を委託しているSI企業B社の要員との共同体制で行うこととした。既存システムの開発実績を基に、開発規模や要員の生産性などを推定して、プロジェクトの要員計画を作成する。

[プロジェクトの説明]

- (1) 新システムの開発規模は、第一期及び第二期ともに480kステップである。
- (2) 両期ともにウォータフォールモデルに基づいて、外部設計、内部設計、プログラム開発（単体テストを含む）、結合テスト及び総合テストの五つの工程に分ける。
- (3) 各工程での要員計画の前提条件は、次のとおりである。
  - ① 各月のA社の要員数は、第一期及び第二期の全期間を通して13名に固定する。
  - ② A社の要員には、全期間を通して全員に作業を常に割り当てる。
  - ③ プログラム開発工程には、第一期及び第二期ともにA社の要員を割り当てない。
  - ④ 各月の必要要員のうち、A社の要員だけでは不足する場合には、B社の要員を割り当てる。
- (4) 第一期は平成24年1月から開始する。第二期は第一期のプログラム開発の開始月から並行して開始する予定である。

設問1 要員数の算出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。

第一期における各工程の生産性、工数及び配分月数を表1のとおりに設定した。生産性とは、各工程での1人月当たりの開発規模であり、配分月数とは、あらかじめ各工程に設定した開発期間（月数）である。

各工程の工数は次の式で算出する。

$$\text{各工程の工数 (人月)} = \frac{\text{開発規模 (k ステップ)}}{\text{各工程の生産性 (k ステップ/人月)}}$$

表1 第一期における各工程の生産性、工数及び配分月数

工程	外部設計	内部設計	プログラム開発	結合テスト	総合テスト
生産性(k ステップ/人月)	10.0	6.0	3.0	8.0	10.0
工数(人月)	48	80	160	a	48
配分月数(月)	3	3		2	3

注記 網掛けの部分は表示していない。

各工程の各月の要員数を求めるために、各月の平均要員数を次の式によって算出し、その値を該当する工程の各月の要員数とする。

なお、平均要員数の値は、小数点以下を切り上げた整数値にする。

$$\text{各月の平均要員数 (人)} = \frac{\text{各工程の工数 (人月)}}{\text{各工程の配分月数 (月)}}$$

例えば、外部設計工程では、平成24年1月～3月のB社の要員数は、各月ともに [ b ] 人になる。同様にして、第一期及び第二期の全工程についてB社の要員数を求める。ここで、第二期の各工程の生産性、工数及び配分月数は第一期に同じとする。

aに関する解答群

ア 52 イ 60 ウ 68 エ 72 オ 80

bに関する解答群

ア 2 イ 3 ウ 4 エ 15 オ 16 カ 17

設問2 開発スケジュール案の検討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。

各月でB社の要員数がばらついていることが分かった。そこで、プログラム開発工程の配分月数を調整して、各月のB社の要員数をできるだけ平準化する。納期の制約から、プログラム開発工程の配分月数は、3～5となる。

第一期及び第二期ともにプログラム開発工程の配分月数を3とした開発スケジュール案1と、配分月数を4とした開発スケジュール案2を検討した。しかし、案1では [c]、案2では [d] が、プロジェクトの説明(3)の前提条件②を満たせないことが分かった。

この問題は、プログラム開発工程の配分月数を、“[e]”又は“第一期を5に第二期を4にする”ことで解消できる。後者を選択した場合の開発スケジュールは表2のようになる。

表2 プログラム開発工程の第一期を5に第二期を4にする開発スケジュール案

年		平成24年												平成25年												
月		1	2	3	4	5	6	7	8	9	10	11	12	1	2	3	4	5	6	7	8	9				
工程	第一期	外部設計		内部設計		プログラム開発				結合テスト		総合テスト														
	第二期							外部設計		内部設計		プログラム開発		結合テスト		総合テスト										
A社の要員数	第一期	13	13	13	13	13	13	0	0	0	0	0	0	13	13	13	13									
	第二期							13	13	13	13	13	13	0	0	0	0	13	13	13	13					
B社の要員数	第一期				14	14	14																			
	第二期																									

注記 網掛けの部分は表示していない。

#### c, dに関する解答群

- |            |            |           |
|------------|------------|-----------|
| ア 平成24年11月 | イ 平成24年12月 | ウ 平成25年1月 |
| エ 平成25年2月  | オ 平成25年3月  | カ 平成25年4月 |

#### eに関する解答群

- |                  |                  |
|------------------|------------------|
| ア 第一期を3に第二期を4にする | イ 第一期を3に第二期を5にする |
| ウ 第一期を4に第二期を3にする | エ 第一期を4に第二期を5にする |

設問3 最終的な要員計画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。

最終的な要員計画として、プログラム開発工程の配分月数を、第一期を5に第二期を4にする案を選択した。

第一期及び第二期の各工程における各月の平均要員数を算出する。その値を各月の要員数とすると、各月の総要員数のピーク時は [f] で [g] 人となり、その月だけ他の月よりも要員数が突出する。ピークを減らして、その月と前後の月の3か月の要員数と同じにするためには、当該月の [h] とすればよい。

なお、当該月の作業の一部を同一工程内の前月に移動することを“前倒し”と呼び、後月に移動することを“後ろ倒し”と呼んでいる。

#### fに関する解答群

- |            |             |             |
|------------|-------------|-------------|
| ア 平成24年 7月 | イ 平成24年 10月 | ウ 平成24年 12月 |
| エ 平成25年 1月 | オ 平成25年 2月  |             |

#### gに関する解答群

- |      |      |      |      |      |
|------|------|------|------|------|
| ア 56 | イ 59 | ウ 63 | エ 66 | オ 70 |
|------|------|------|------|------|

#### hに関する解答群

- |                            |
|----------------------------|
| ア 第一期及び第二期の作業とともに前倒し       |
| イ 第一期及び第二期の作業とともに後ろ倒し      |
| ウ 第一期の作業を前倒しとし、第二期の作業を後ろ倒し |
| エ 第一期の作業を後ろ倒しとし、第二期の作業を前倒し |

問7 販売ルート別の売上及び市場の分析に関する記述を読んで、設問1、2に答えよ。

W社は、ある香辛料の製造販売を行っており、販売ルートは、小売店、量販店、コンビニエンスストア（以下、CVSという）、ネット通信販売（以下、ネット通販という）、外食、加工食品メーカ（以下、加工という）の6ルートからなる。

W社では、販売戦略上の問題点を把握し、適切な対応を行うことを目的として、販売ルート別の売上及び市場に関する分析を行っている。そのための指標として売上金額、売上数量、売上単価、市場金額規模（以下、市場規模という）、市場金額占有率（以下、市場占有率という）を利用している。販売ルート別の指標の前年実績及び当年見込を表1に示す。ここで、各数値は、小数第1位を四捨五入した値である。

表1 販売ルート別の指標

指標		小売店	量販店	CVS	ネット通販	外食	加工
売上金額 (百万円)	前年実績	3,400	3,330	1,092	805	7,200	6,210
	当年見込	3,360	4,025	1,150	726	5,760	7,200
売上数量 (t)	前年実績	850	900	260	230	2,400	2,700
	当年見込	800	1,150	230	220	1,800	3,000
売上単価 (千円/t)	前年実績	4,000	3,700	4,200	3,500	3,000	2,300
	当年見込	4,200	3,500	5,000	3,300	3,200	2,400
市場規模 (百万円)	前年実績	12,143	16,650	7,280	4,025	18,000	20,700
	当年見込	10,839	18,295	8,214	4,840	15,158	20,000
市場占有率 (%)	前年実績	28	20	15	20	40	30
	当年見込	31	22	14	15	38	36

注記 tは、重量の単位であるトン（1,000kg）を表す。

設問1 販売ルート別の分析に関する次の記述及び図中の□に入る正しい  
答えを、解答群の中から選べ。

当年の販売ルート別の売上及び市場の分析を行うために、表2の説明に従い、  
図1及び図2のバブルチャートを作成した。ここで、バブルの大きさは、売上金額の当年見込の販売ルート別構成比（以下、当年構成比という）を表す。

表2 図1及び図2のバブルチャートの説明

図	横軸の指標	縦軸の指標	バブルの大きさ
図1 当年売上分析	売上数量 当年見込 (t)	売上単価 当年見込 (千円/t)	当年構成比 (%)
図2 当年市場分析	市場規模 当年見込 (百万円)	市場占有率 当年見込 (%)	

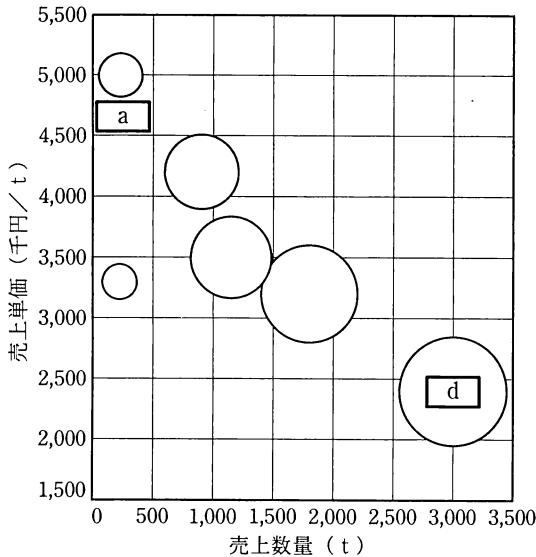


図1 当年売上分析

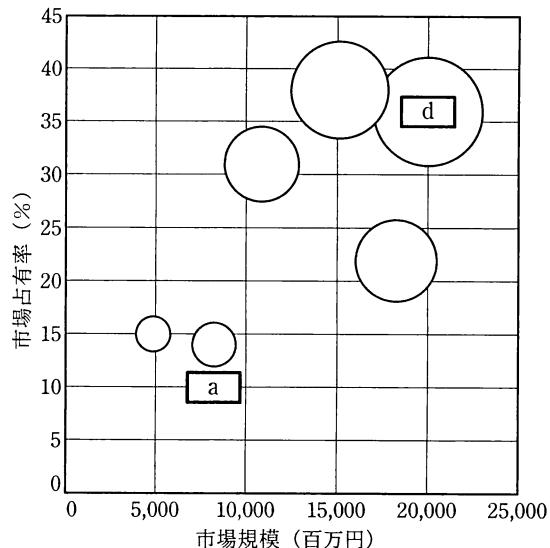


図2 当年市場分析

[当年売上分析・当年市場分析の結果の考察]

□a□は、□b□が最も小さいが、□c□が最も高いので、  
□b□を高めることで、売上金額の増加が見込まれる。□d□は、  
□c□が最も低いが、売上数量及び市場規模が最も多いので、これを維持  
した上で、□c□を高めることができれば、売上金額の増加が見込まれる。

a, dに関する解答群

ア CVS

イ 加工

ウ 外食

エ 小売店

オ ネット通販

カ 量販店

b, cに関する解答群

ア 売上金額

イ 売上数量

ウ 売上単価

エ 市場規模

オ 市場占有率

カ 当年構成比

設問 2 販売ルート別の各指標の前年比分析に関する次の図表中及び記述中の  
 [ ] に入る正しい答えを、解答群の中から選べ。ただし、g1, g2 及び  
 h1, h2 に入る答えは、それぞれ g に関する解答群及び h 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。

売上及び市場に関する各指標の増減傾向を把握し、各販売ルートに対する有効な施策を考えるために、各指標の前年実績に対する当年見込の比率（以下、前年比という）を分析する。表 3 の説明に従い、図 3 及び図 4 のバブルチャートを作成した。ここで、バブルの大きさに指定した前年構成比とは、売上金額の前年実績の販売ルート別の構成比のことである。

表 3 図 3 及び図 4 のバブルチャートの説明

図	横軸の指標	縦軸の指標	バブルの大きさ
図3 前年比分析1	[ ] e		前年構成比 (%)
図4 前年比分析2	[ ] f		

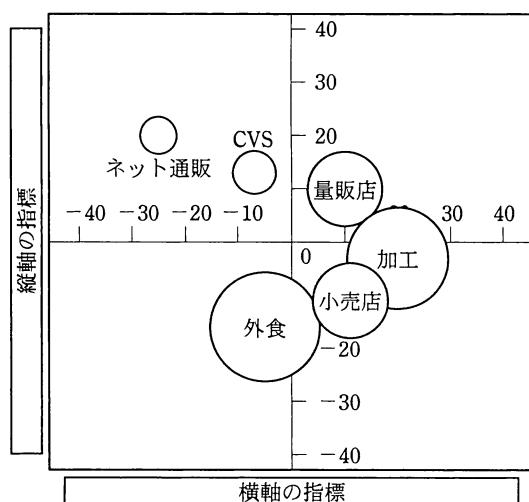


図 3 前年比分析 1

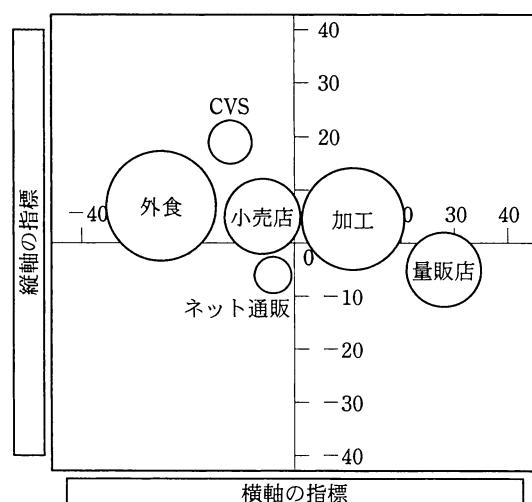


図 4 前年比分析 2

#### 〔前年比分析結果の考察〕

市場規模が 20%ほど増加しているのに市場占有率が 25%ほど減少している販

売ルートは、g1であり、競合に対抗するための経営資源の投入が望まれる。売上数量が25%以上増加している一方、売上単価が低下している販売ルートは、g2であり、今後、値引きを抑える又は高付加価値の商品を投入するなど、売上単価を高めるための施策を検討する。

図3、4から、前年構成比の最も大きい外食の当年見込は、前年に比べて、h1したことが分かる。前年構成比が次に大きい加工の当年見込は、前年に比べて、h2したことが分かる。

#### e, fに関する解答群

	横軸の指標	縦軸の指標
ア	売上金額前年比-100 (%)	売上数量前年比-100 (%)
イ	売上金額前年比-100 (%)	売上単価前年比-100 (%)
ウ	売上数量前年比-100 (%)	売上金額前年比-100 (%)
エ	売上数量前年比-100 (%)	売上単価前年比-100 (%)
オ	売上単価前年比-100 (%)	売上金額前年比-100 (%)
カ	売上単価前年比-100 (%)	売上数量前年比-100 (%)
キ	市場規模前年比-100 (%)	売上金額前年比-100 (%)
ク	市場規模前年比-100 (%)	市場占有率前年比-100 (%)
ケ	市場占有率前年比-100 (%)	売上金額前年比-100 (%)
コ	市場占有率前年比-100 (%)	市場規模前年比-100 (%)

g に関する解答群

	g1	g2
ア	CVS	小売店
イ	CVS	ネット通販
ウ	CVS	量販店
エ	小売店	CVS
オ	ネット通販	CVS
カ	ネット通販	小売店
キ	ネット通販	量販店
ク	量販店	CVS
ケ	量販店	小売店
コ	量販店	ネット通販

h に関する解答群

	h1	h2
ア	売上数量, 売上単価などが増加	売上数量, 売上単価などが増加
イ	売上数量, 市場規模などが減少	売上数量, 市場規模などが増加
ウ	売上数量, 市場規模などが減少	売上単価, 市場占有率などが増加
エ	売上数量, 市場占有率などが減少	売上単価, 市場規模などが減少
オ	売上数量, 市場占有率などが減少	市場規模, 市場占有率などが減少
カ	売上単価, 市場規模などが減少	売上数量, 市場占有率などが増加
キ	売上単価, 市場占有率などが減少	売上数量, 市場占有率などが増加
ク	市場規模, 市場占有率などが減少	市場規模, 市場占有率などが増加
ケ	市場規模, 市場占有率などが減少	売上単価, 市場規模などが増加
コ	市場規模, 市場占有率などが増加	売上単価, 市場占有率などが増加

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。

1行の代入文を解析し、演算の優先順位に従って一つずつ演算を行っていく一連の代入文に変換して出力する。代入文とその出力結果の例を、次に示す。ここで、出力結果にある `wk#1` などは、中間結果を保存するための作業変数である。

(代入文の例)  
`Ans=X1+10*X2`

(出力結果の例)  
`wk#1=10*X2  
wk#2=X1+wk#1  
Ans=wk#2`

代入文の形式は、次のとおりである。

変数 = 式

“=”は代入演算子で、右辺の式の値を評価した結果を、左辺の変数に代入する。

式は、変数又は定数（以下、項という）の一つ以上の並びで、項が二つ以上のときは各項の間に算術演算子を置く。変数は、英字（“A”～“Z”，“a”～“z”）で始まる1文字以上の英数字の列である。定数は、1文字以上の数字（“0”～“9”）の列である。

算術演算子は、加算（“+”）、減算（“-”）、乗算（“\*”）、除算（“/”）の4種類である。乗除算は加減算に優先する。同一優先順位の算術演算子は、左から順に演算する。

#### [代入文の構文解析]

解析処理は、次の手順による。

- (1) 代入文を走査して、英字・数字・代入演算子・算術演算子以外の文字が含まれていれば、`err` に -1 を格納して手順(5)へ進む。`err` は、文法上の誤りがあった場合のエラーコードを格納する変数である。
- (2)  $n$  文字からなる代入文を、文字型配列 `S` の要素番号 1～ $n$  に格納し、`S[0]` に開始マーク “`«`” を、`S[n+1]` に終端マーク “`»`” を、それぞれ格納する。次に、整数型配列 `V` を用意し、`S[i]` の内容に対応した表1のコードを `V[i]` に格納する。その例を、次に示す。

配列 S の内容	«		A		n		s		=		x		1		+		1		0		*		x		2		»	
配列 V の内容	0		2		2		10		2		3		11		3		3		12		2		3		1			

表1 文字とコードの対応

S[i]の内容	«	»	英字	数字	=	+, -	*, /
V[i]に格納する内容	0	1	2	3	10	11	12

- (3) 初期値として st に “開始” を, err 及び i に 0 を, それぞれ格納する。st は, 解析の状態を格納している変数である。
- (4) i に 1 を加算する。表 2 で, 現在の st の状態 (行) と着目している文字 S[i] の内容 (列) が交差するセルの内容を実行する。ただし, セルが空白の場合は, 何も実行しない。実行の結果, err の値が 0 以外となったら手順 (5) へ, st の値が “終端” となったら手順 (6) へそれぞれ進む。それ以外の場合は, この手順 (4) を繰り返す。

表2 状態遷移表

st \ S[i]	英字	数字	=	+, -, *, /	»
“開始”	st ← “左辺”	err ← 12	err ← 13	err ← 14	err ← 15
“左辺”			st ← “代入”	err ← 24	err ← 25
“代入”	st ← “変数”	st ← “定数”	err ← 33	err ← 34	err ← 35
“変数”			err ← 43	st ← “演算”	st ← “終端”
“定数”	err ← 51		err ← 53	st ← “演算”	st ← “終端”
“演算”	st ← “変数”	st ← “定数”	err ← 63	err ← 64	err ← 65

- (5) err の値に応じて適切なエラーメッセージを表示し, 処理を終了する。
- (6) 文法上の誤りがなかった旨を表示し, 処理を終了する。

設問 1 〔代入文の構文解析〕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。

文法上の誤りがある次の代入文①～④を, 解析処理の手順によって解析した。

代入文① Answer=One+Two+Three+  
 代入文② HexaSum=7FFF+0001  
 代入文③ Position=Index++  
 代入文④ X1+10\*X2=Ans

解析処理の手順 (5) に進んだとき, [a] の場合は err の値が 51 に,  
 [b] の場合は err の値が 64 になっている。

## 解答群

ア 代入文①

イ 代入文②

ウ 代入文③

エ 代入文④

### [代入文の変換]

変換処理は、次の手順による。[代入文の構文解析]の手順は実行済みで、代入文には文法上の誤りがないものとする。

- (1) 初期値として、 $w$  に 0 を格納する。
- (2) 配列  $S, V$  の開始マークから終端マークまでの範囲を検査し、優先順位が最も高い最初の算術演算子の要素番号を変数  $next$  に格納する。その例を、次に示す。算術演算子がなければ、 $next$  に 0 を格納する。

配列  $S$  |«|A|n|s|=|X|1|+|1|0|\*|X|2|»|  
↑  $next \leftarrow 10$

- (3)  $next$  が 0 なら、配列  $S$  中の代入文を出力して、処理を終了する。
- (4)  $next$  の位置の演算子とその前後の項からなる文字列を抜き出す(図 1 の①)。 $w$  に 1 を加算し、出力する代入文を編集して出力する。出力する代入文は、文字列 “ $wk\#$ ”、文字列に変換した値  $w$ 、文字 “=” 及び抜き出した文字列をこの順に連結したものである。次に、配列  $S$  について、抜き出した元の文字列を、文字列 “ $wk\#$ ” 及び文字列に変換した値  $w$  で置き換える(図 1 の②)。置き換えるときに、文字数が増える場合は以降の文字を後ろにずらし、減る場合は以降の文字を前に詰める。さらに、更新された配列  $S$  の内容に応じて配列  $V$  の内容を更新する。文字 “#” に対応するコードは、英字のコードと同じとする。

配列  $S$  |«|A|n|s|=|X|1|+|1|0|\*|X|2|»|  
出力 | $w|k|\#|1|=|1|0|*|X|2|$ |  
①  
配列  $S$  |«|A|n|s|=|X|1|+|w|k|\#|1|»|  
②

図 1 文字列の置換の例

- (5) 手順(2)へ戻り、処理を繰り返す。

設問2 [代入文の変換] に関する次のプログラム中及び記述中の  に入れ  
る正しい答えを、解答群の中から選べ。

なお、プログラム中の関数 `Getpos(配列, 値)` は、値が格納されている配列中の最初の要素番号を返す。また、配列 `S, V` は大域変数として与えられ、副プログラム中から参照できるものとする。

(1) 手順(2)の処理を行う2種類の副プログラム1, 2を作成した。

[副プログラム1]

- 整数型関数：プログラム1
- 整数型：`i, next, priority`
- $i \leftarrow 1$    $\alpha$
- $next \leftarrow 0$
- $priority \leftarrow 10$
- $V[i] \neq 1$ 
  -   $\uparrow$
  - $next \leftarrow i$
  - $priority \leftarrow V[i]$
  - $\downarrow$
  - $i \leftarrow i + 1$
- $return next$

[副プログラム2]

- 整数型関数：プログラム2
- 整数型：`i, next, priority`
- $i \leftarrow Getpos(S[], "»") - 1$
- $next \leftarrow 0$
- $priority \leftarrow 11$
- $V[i] \neq 0$ 
  -   $\uparrow$
  - $next \leftarrow i$
  - $priority \leftarrow V[i]$
  - $\downarrow$
  - $i \leftarrow i - 1$
- $return next$

プログラム1の  $\alpha$  の行では、 $i$  に 1 を格納している。

ここで、 $\alpha$  の行を “ $i \leftarrow$ ” とすれば、繰返し処理の繰返し回数を最小にすることができる。

(2) 手順(4)の処理中で使用する副プログラム3を作成した。このプログラムは、配列 `S, V` の要素番号 `from` 以降（終端マークまで）の要素を前後に移動する。`move` は、移動する桁数と方向を示し、符号が正の場合は後ろにずらし、符号が負の場合は前に詰める。`move` の値が  $-3$  (左に 3 桁詰める) の場合の例を、次に示す。ここで、配列 `S, V` は十分な大きさがあるものとする。

配列 `S` (移動前) | «|A|n|s|=|w|k|#|2|+|1|0|+|w|k|#|1|» |  
 配列 `S` (移動後) | «|A|n|s|=|w|k|#|2|+|w|k|#|1|» |  
 ↑ `from`

[副プログラム 3]

○プログラム 3(整数型 : from, 整数型 : move)

○整数型 : i, to

• to  $\leftarrow \text{Getpos}(S[], "»")$

$\uparrow$  move  $> 0$

i: f

- $S[i+move] \leftarrow S[i]$
- $V[i+move] \leftarrow V[i]$

$\downarrow$  move  $< 0$

i: g

- $S[i+move] \leftarrow S[i]$
- $V[i+move] \leftarrow V[i]$

c, d に関する解答群

ア priority <  $V[i]$

イ priority  $\leq V[i]$

ウ priority  $\geq V[i]$

エ priority >  $V[i]$

e に関する解答群

ア  $\text{Getpos}(S[], "«") + 1$

イ  $\text{Getpos}(S[], "«") + 2$

ウ  $\text{Getpos}(S[], "=") - 1$

エ  $\text{Getpos}(S[], "=") + 2$

f, g に関する解答群

ア from, i < to, 1

イ from, i  $\leq$  to, 1

ウ to, i  $\geq$  from, -1

エ to, i > from, -1

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、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。

[プログラム1の説明]

二つの整数  $x$ ,  $y$  ( $0 < x < y$ ) を受け取り、 $x/y$  の値を10進小数として出力するプログラムである。

- (1) 関数 `printRational` 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。

$x$ : 分子を表す正の整数

$y$ : 分母を表す正の整数

ただし、 $x/y$  は有限小数（小数点以下の桁数が有限桁である小数）であり、循環小数（小数部のある桁以降で、同じ数字の列が無限に繰り返される小数）ではないものとする。

- (2) 次の手順で  $x/y$  を10進小数として出力する。

① “0.”を出力する。

②  $x$  が0になるまで、次の③、④を繰り返す。

③  $x$  を10倍した値を  $y$  で割った商を出力する。

④  $x$  を10倍した値を  $y$  で割った余りを新たに  $x$  とする。

[プログラム1]

(行番号)

```
1 #include <stdio.h>

2 void printRational(int, int);

3 void printRational(int x, int y) {
4     putchar('0');
5     putchar('.');
6     while (x > 0) {
7         putchar('0' + (x * 10 / y));
8         x = x * 10 % y;
9     }
10    putchar('\n');
11 }
```

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。

- (1) プログラム 1 の行番号 7 では小数点以下の各桁の数字を出力している。この行を “`putchar('0' + ( x / y * 10));`” に変えると  a。
- (2) `printRational(3, 8)` を実行した場合、行番号 6 の条件判定が 2 回目に行われるときの  $x$  の値は  b であり、プログラムが終了するまでに、行番号 6 の条件判定は  c 回行われる。

C

a に関する解答群

- ア 乗算が除算よりも先に実行されるが、正しい値が出力される  
イ 乗算が除算よりも先に実行され、正しい値が出力されない  
ウ 除算が乗算よりも先に実行されるが、正しい値が出力される  
エ 除算が乗算よりも先に実行され、正しい値が出力されない

b, c に関する解答群

- |     |     |     |     |     |
|-----|-----|-----|-----|-----|
| ア 0 | イ 1 | ウ 2 | エ 3 | オ 4 |
| カ 5 | キ 6 | ク 7 | ケ 8 | コ 9 |

設問 2 次の説明を読み、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。

$x/y$  が循環小数となる  $x$  と  $y$  を引数に指定して関数 `printRational` を実行した場合、ある桁以降で同じ数字の列を無限に繰り返し出力し続け、プログラムは終了しない。この繰り返し出力される同じ数字の列を循環節と呼ぶ。最初に現れる循環節を “ [ ” と “ ] ” で囲んで出力するプログラムを作成した。

[プログラム 2 の説明]

- (1) 関数 `recurringDecimal` 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。

$x$  : 分子を表す正の整数

$y$  : 分母を表す正の整数

- (2)  $x/y$  の値によって次のように場合分けを行い、それぞれ別の様式の小数点数を出力する。

- ① 小数点以下 100 衡までに割り切れる場合

小数点数をそのまま出力する。例えば、`recurringDecimal(1, 8)`を実行した場合、次のように出力する。

0.125

- ② 小数点以下 100 衡までの間に循環節がある場合

最初の循環節の直前までの小数点数を出力し、その後に循環節の数字の列を“[”と“]”で囲んで出力する。例えば、`recurringDecimal(3, 22)`を実行した場合、次のように出力する。

0.1[36]

- ③ その他の場合

小数点以下 100 衡までの小数点数を出力し、その後に“...”を続けて出力する。例えば、`recurringDecimal(19, 131)`を実行した場合、次のように出力する。

0.1450381679389312977099236641221374045801526717557251908  
396946564885496183206106870229007633587786259...

- (3) 関数 `recurringDecimal` は、 $x/y$  の値によって出力様式を決める部分と小数点数を出力する部分の二つに分かれている。

- (4)  $x/y$  の値によって出力様式を決める手順は、次のとおりである。

- ① 大きさ 100 の配列 `xHistory` と変数 `ri` を用意する。
- ② `ri` を 0 とする。
- ③ `xHistory[ri]` に  $x$  を格納し、`ri` を 1 増やす。
- ④  $x$  を 10 倍した値を  $y$  で割った余りを新たに  $x$  とする。
- ⑤  $x$  が 0 になった場合、出力様式は(2)の①とする。
- ⑥  $x$  と同じ値をもつ `xHistory[i]` ( $0 \leq i < ri$ ) があるかどうかを調べる。
- ⑦ 同じ値をもつものがある場合、出力様式は(2)の②とする。このとき、 $x/y$  の値の小数点以下第  $i+1$  位から第  $ri$  位までの数字の列が最初の循環節となる。

- ⑧ 同じ値をもつものがない場合、 $ri$  が 100 以上であれば出力様式は(2)の③とする。そうでなければ、③に戻る。

[プログラム 2]

```
#include <stdio.h>

#define DIGITMAX 100 /* 出力する小数点以下の最大桁数 */

void recurringDecimal(int, int);

void recurringDecimal(int x, int y) {
    int xHistory[DIGITMAX];
    int i, ri = 0, startRepeat = -1;

    /* 出力様式の決定 */
    while ((x > 0) && (ri < DIGITMAX) && (d)) {
        xHistory[ri] = x;
        ri++;
        x = x * 10 % y;
        for (i = 0; e; i++) {
            if (xHistory[i] == x) {
                startRepeat = i;
            }
        }
    }
    /* 小数点数の出力 */
    putchar('0');
    putchar('.');
    for (i = 0; i < ri; i++) {
        if (f) {
            putchar '[');
        }
        putchar(g);
    }
    if (startRepeat >= 0) {
        putchar(']');
    } else {
        if ((ri >= DIGITMAX) && (x > 0)) {
            putchar('.');
            putchar('.');
            putchar('.');
        }
    }
    putchar('\n');
}
```

d, e に関する解答群

ア i == 0	イ i < DIGITMAX
ウ i < ri	エ i >= 0
オ ri < DIGITMAX	カ ri > 0
キ startRepeat == -1	ク startRepeat >= 0
ケ startRepeat > i	

f に関する解答群

ア i == startRepeat	イ i == startRepeat + 1
ウ i == startRepeat - i	エ i < startRepeat
オ startRepeat < 0	カ startRepeat >= 0

g に関する解答群

ア x * 10 / y	
イ xHistory[i] * 10 / y	
ウ xHistory[ri] * 10 / y	
エ xHistory[startRepeat] * 10 / y	
オ '0' + x * 10 / y	
カ '0' + xHistory[i] * 10 / y	
キ '0' + xHistory[ri] * 10 / y	
ク '0' + xHistory[startRepeat] * 10 / y	

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]

受験の申込みを登録した申込ファイルから、受験区分別に申込者的人数を集計し、集計表を印字するプログラムである。

- (1) 申込ファイルは図 1 に示すレコード様式の順ファイルである。

		受験区分	
申込番号 5 衔	試験区分 1 衔	開催地区分 2 衔	

図 1 申込ファイルのレコード様式

- ① 受験の申込み 1 件に対して、1 件のレコードが作成され、5 衔の申込番号が付与される。
- ② 受験区分は、試験区分と開催地区分から構成される。
- ③ 試験区分は、試験内容に対応付けた A～H のいずれか 1 文字である。
- ④ 開催地区分は、開催地に対応付けた 01～20 のいずれかの数字である。
- ⑤ 各受験区分の申込者数は、999 人以下である。
- ⑥ 申込ファイルの内容には、誤りはないものとする。

- (2) 集計表の印字例は、図 2 のとおりである。申込者がいない受験区分は、印字しない。

SEGMENT	TOTAL
A01	35
A02	120
A03	54
:	:
H20	79

図 2 集計表の印字例

## [プログラム]

(行番号)

```

1  DATA DIVISION.
2  FILE SECTION.
3  FD  TEST-APPLIC-FILE.
4  01  TEST-APPLIC-REC.
5      02  FILLER          PIC X(05).
6      02  TEST-M.
7          03  TEST-CATEG-M  PIC X(01).
8          03  TEST-PLACE-M  PIC 9(02).
9  FD  PRINT-TEST-FILE.
10 01  PRINT-REC.
11     02  PRINT-TEST-REC.
12         03  TEST-CATEG-S  PIC X(01).
13         03  TEST-PLACE-S  PIC 9(02).
14     02  FILLER          PIC X(10).
15     02  PRINT-S          PIC ZZ9.
16  WORKING-STORAGE SECTION.
17 01  FILE-END          PIC X(01) VALUE "N".
18 01  LOOP1              PIC 9(01).
19 01  LOOP2              PIC 9(02).
20 01  TEST-NO             PIC 9(01).
21 01  TEST-W1.
22     02  TEST-W2          PIC X(08) VALUE "ABCDEFGH".
23     02  REDEFINES TEST-W2.
24         03  TEST-W3          PIC X(01) OCCURS 8 INDEXED BY TEST-IDX.
25 01  SUM-WK1.
26     02  OCCURS 8.
27         03  SUM-WK          PIC 9(03) OCCURS 20 VALUE ZERO.
28 01  HEADER              PIC X(16) VALUE "SEGMENT    TOTAL".
29  PROCEDURE DIVISION.
30  MAIN-PROC.
31      OPEN INPUT TEST-APPLIC-FILE OUTPUT PRINT-TEST-FILE.
32      PERFORM SUM-PROC UNTIL FILE-END = "Y".
33      PERFORM PRINT-PROC.
34      CLOSE TEST-APPLIC-FILE PRINT-TEST-FILE.
35      STOP RUN.
36  SUM-PROC.
37      READ TEST-APPLIC-FILE
38          AT END
39              MOVE "Y" TO FILE-END
40          NOT AT END
41          a
42          SEARCH b
43              WHEN TEST-CATEG-M = TEST-W3(TEST-IDX)
44                  SET TEST-NO TO TEST-IDX
45          END-SEARCH
46          ADD 1 TO SUM-WK(TEST-NO TEST-PLACE-M)
47      END-READ.

```

48 PRINT-PROC.  
 49     WRITE PRINT-REC FROM HEADER AFTER PAGE.  
 50     PERFORM VARYING LOOP1 FROM 1 BY 1 UNTIL LOOP1 > 8  
 51         PERFORM VARYING LOOP2 FROM 1 BY 1 UNTIL LOOP2 > 20  
 52         IF C NOT = 0 THEN  
 53             MOVE SPACE TO PRINT-REC  
 54             MOVE TEST-W3(LOOP1) TO TEST-CATEG-S  
 55             MOVE LOOP2 TO TEST-PLACE-S  
 56             MOVE C TO PRINT-S  
 57             WRITE PRINT-REC AFTER 1  
 58         END-IF  
 59     END-PERFORM  
 60 END-PERFORM.

設問1 プログラムの  に入れる正しい答えを、解答群の中から選べ。

COBOL

a に関する解答群

- |                                |                           |
|--------------------------------|---------------------------|
| ア MOVE 1 TO TEST-IDX           | イ MOVE 1 TO TEST-NO       |
| ウ MOVE TEST-IDX TO TEST-NO     | エ SET TEST-IDX TO 1       |
| オ SET TEST-IDX TO TEST-CATEG-S | カ SET TEST-IDX TO TEST-NO |

b に関する解答群

- |           |            |           |
|-----------|------------|-----------|
| ア SUM-WK  | イ TEST-IDX | ウ TEST-NO |
| エ TEST-W1 | オ TEST-W2  | カ TEST-W3 |

c に関する解答群

- |                                |                                |
|--------------------------------|--------------------------------|
| ア SUM-WK(LOOP1 LOOP2)          | イ SUM-WK(LOOP2 LOOP1)          |
| ウ SUM-WK(TEST-CATEG-M TEST-NO) | エ SUM-WK(TEST-NO TEST-PLACE-M) |
| オ SUM-WK(TEST-PLACE-M TEST-NO) | カ SUM-WK(TEST-PLACE-S TEST-NO) |

設問2 集計表における受験区分の印字を申込者数の多い順になるようにプログラムを変更したい。表1に示すプログラムの変更内容の [ ] に入る正しい答えを、解答群の中から選べ。ここで、 [ ] c には設問1の正しい答えが入っているものとする。

表1 プログラムの変更内容

処置	変更内容
行番号2と3 の間に追加	<pre> SD TEST-SORT-FILE. 01 SORT-REC.   02 SORT-TEST-REC PIC X(03).   02 SORT-S      PIC 9(03). FD TEST-SUM-FILE. 01 SUM-REC.   02 SUM-TEST-REC.     03 TEST-CATEG-S PIC X(01).     03 TEST-PLACE-S PIC 9(02).   02 SUM-S        PIC 9(03). </pre>
行番号48～ 60を変更	<pre> PRINT-PROC.   OPEN OUTPUT TEST-SUM-FILE.   PERFORM VARYING LOOP1 FROM 1 BY 1 UNTIL LOOP1 &gt; 8     PERFORM VARYING LOOP2 FROM 1 BY 1 UNTIL LOOP2 &gt; 20       IF [ ] c NOT = 0 THEN         MOVE TEST-W3(LOOP1) TO TEST-CATEG-S OF SUM-REC         MOVE LOOP2 TO TEST-PLACE-S OF SUM-REC         MOVE [ ] c TO SUM-S         WRITE SUM-REC       END-IF     END-PERFORM   END-PERFORM.   CLOSE TEST-SUM-FILE.   OPEN INPUT TEST-SUM-FILE.   SORT TEST-SORT-FILE [ ] d     INPUT PROCEDURE IS IN-PROC     OUTPUT PROCEDURE IS OUT-PROC.   CLOSE TEST-SUM-FILE.   IN-PROC.     MOVE "N" TO FILE-END.     PERFORM UNTIL FILE-END = "Y"       READ TEST-SUM-FILE       AT END         MOVE "Y" TO FILE-END       NOT AT END         MOVE SUM-TEST-REC TO SORT-TEST-REC         MOVE SUM-S TO SORT-S </pre>

```

e
END-READ
END-PERFORM.
OUT-PROC.
  WRITE PRINT-REC FROM HEADER AFTER PAGE.
  MOVE "N" TO FILE-END.
  PERFORM UNTIL FILE-END = "Y"
f
  AT END
    MOVE "Y" TO FILE-END
  NOT AT END
    MOVE SPACE TO PRINT-REC
    MOVE SORT-TEST-REC TO PRINT-TEST-REC
    MOVE SORT-S TO PRINT-S
    WRITE PRINT-REC AFTER 1
g
END-PERFORM.

```

## d に関する解答群

- ア ASCENDING KEY SORT-S
- イ ASCENDING KEY SORT-TEST-REC
- ウ DESCENDING KEY SORT-S
- エ DESCENDING KEY SORT-TEST-REC

## e～g に関する解答群

- ア END-READ
- イ END-RETURN
- ウ READ TEST-SORT-FILE
- エ READ TEST-SUM-FILE
- オ RELEASE SORT-REC
- カ RELEASE SUM-REC
- キ RETURN TEST-SORT-FILE
- ク RETURN TEST-SUM-FILE

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]

名前及び住所からなる住所録のエントリの追加、削除及び検索を行う住所録管理プログラムであり、次のクラスからなる。

- (1) クラス `Name` は、姓及び名をそれぞれ文字列で保持する。
- (2) クラス `Address` は、郵便番号及び住所を保持する。郵便番号は、上位 3 衔と下位 4 衔をそれぞれ `int` 型の整数で保持する。住所は、文字列で保持する。
- (3) クラス `AddressBook` は、住所録である。入れ子クラス `AddressBook.Entry` は、住所録のエントリであり、`Name` と `Address` のインスタンスを保持する。クラス `AddressBook` は、エントリを追加及び削除するメソッドをもつ。

クラス `Name`, `Address`, 及び `AddressBook.Entry` のインスタンスは、検索可能である。検索処理を支援するために、次のインターフェースを定義する。

- (1) インタフェース `SearchCriteria` は、検索条件を示すデータ型であり、メソッドをもたない。
- (2) インタフェース `Searchable` は、これを実装するクラスが検索条件を与えて検索可能であることを示す。検索するときは、メソッド `meets` を呼び出す。

クラス `Name` 及び `Address` は、それぞれ入れ子クラス `Criteria` を定義し、その `Criteria` を検索条件とするインターフェース `Searchable` を実装する。

- (1) 入れ子クラス `Name.Criteria` は、引数 `familyName` 及び `givenName` でそれぞれ姓及び名を検索条件として指定する。検索条件に含めないとときは、`null` を指定する。
- (2) クラス `Name` は、インターフェース `Searchable` を実装する。メソッド `meets` は、引数で与えられた `criteria` で実装されたメソッドを呼び出し、このクラスのインスタンスが検索条件に合致するかどうか調べる。
- (3) 入れ子クラス `Address.Criteria` は、引数 `postalCode3`, `postalCode4` 及び `addr` でそれぞれ郵便番号の上位 3 衔、下位 4 衔及び住所を検索条件として指

定する。検索条件に含めないときは、`postalCode3` 及び `postalCode4` は負の値、`addr` は `null` を指定する。住所については、部分文字列が一致する場合も合致しているとみなす。例えば、“京都”は“東京都”的部分文字列なので、合致するとみなす。

- (4) クラス `Address` は、インターフェース `Searchable` を実装する。メソッド `meets` は、引数で与えられた `criteria` で実装されたメソッドを呼び出し、このクラスのインスタンスが検索条件に合致するかどうか調べる。

クラス `AddressBook` のメソッド `meetsAnyOf` は、引数で与えられたインターフェース `SearchCriteria` のどれかに合致するエントリの集合を返す。メソッド `meetsAllOf` は、引数で与えられた `SearchCriteria` の全てに合致するエントリの集合を返す。

入れ子クラス `AddressBook.Entry` は、インターフェース `Searchable` を実装する。メソッド `meets` は、引数で与えられた `SearchCriteria` の具体的な型によって、`name` 又は `addr` のメソッド `meets` を呼び出し、この `AddressBook.Entry` のインスタンスが引数で指定された検索条件に合致するかどうか調べる。

なお、クラス `Name`、`Address`、`AddressBook.Entry` は、インターフェース `Set` で使用できるように、クラス `Object` のメソッド `equals` 及び `hashCode` を上書きしているものとする。また、各構造体及びメソッドの引数は正しいものとする。

クラス `Test` は、この住所録プログラムのテストプログラムである。メソッド `main` を実行すると、次の結果が得られた。

```
[技術 太郎: 〒225-1234 横浜市青葉区, 試験 一朗: 〒980-9876 仙台市青葉区, 情報  
太郎: 〒102-4567 東京都千代田区]  
[技術 太郎: 〒225-1234 横浜市青葉区]
```

図 1 テストプログラムの実行結果

[プログラム 1]

```
public class Name implements Searchable<Name.Criteria> {  
    private final String familyName, givenName;  
  
    public static class Criteria implements a {  
        private final String familyName, givenName;
```

```

public Criteria(String familyName, String givenName) {
    this.familyName = familyName;
    this.givenName = givenName;
}
private boolean areMetBy(Name name) {
    return (familyName == null
        || name.familyName.equals(familyName))
    && (givenName == null
        || name.givenName.equals(givenName));
}
}

public Name(String familyName, String givenName) {
    this.familyName = familyName;
    this.givenName = givenName;
}

public String getFamilyName() { return familyName; }

public String getGivenName() { return givenName; }

public boolean meets(Criteria criteria) {
    return criteria.areMetBy(this);
}

public String toString() {
    return familyName + " " + givenName;
}
}

```

## (プログラム2)

```

public class Address implements Searchable<Address.Criteria> {
    private final int postalCode3, postalCode4;
    private final String addr;

    public static class Criteria implements SearchCriteria {
        private final int postalCode3, postalCode4;
        private final String addr;

        public Criteria(int postalCode3, int postalCode4,
                       String addr) {
            this.postalCode3 = postalCode3;
            this.postalCode4 = postalCode4;
            this.addr = addr;
        }
        private boolean areMetBy(Address address) {
            return (postalCode3 < 0
                || postalCode3 == address.postalCode3)
            && (postalCode4 < 0
                || postalCode4 == address.postalCode4)
        }
    }
}

```

```

        && (addr == null
            || address.addr.contains(addr));
    }

    public Address(int postalCode3, int postalCode4, String addr) {
        this.postalCode3 = postalCode3;
        this.postalCode4 = postalCode4;
        this.addr = addr;
    }

    public boolean meets(Criteria criteria) {
        return criteria.areMetBy(this);
    }

    public String toString() {
        return String.format("〒%03d-%04d %s",
                            postalCode3, postalCode4, addr);
    }
}

```

[プログラム 3]

```

import java.util.HashSet;
import java.util.Set;

public class AddressBook {
    private Set<Entry> book = new HashSet<Entry>();

    public void add(Name name, Address addr) {
        book.add(new Entry(name, addr));
    }

    public void remove(Entry entry) { book.remove(entry); }

    public Set<Entry> meetsAnyOf(SearchCriteria... criteria) {
        Set<Entry> result = new HashSet<Entry>();
        for (SearchCriteria sc : criteria) {
            for (Entry entry : book) {
                if (entry.meets(sc))
                    b;
            }
        }
        return result;
    }

    public Set<Entry> meetsAllOf(SearchCriteria... criteria) {
        Set<Entry> result = new HashSet<Entry>(book);
        for (Entry entry : book) {
            for (SearchCriteria sc : criteria) {
                if (!entry.meets(sc)) {

```

```

    [c];
    break;
}
}
return result;
}

public static class Entry implements
    Searchable<SearchCriteria> {
    private final Name name;
    private final Address addr;

    public Entry(Name name, Address addr) {
        this.name = name;
        this.addr = addr;
    }

    public Name getName() { return name; }

    public Address getAddress() { return addr; }

    public boolean meets([d] criteria) {
        if (criteria instanceof Name.Criteria)
            return name.meets((Name.Criteria) criteria);
        if (criteria instanceof [e])
            return addr.meets(([e]) criteria);
        return false;
    }

    public String toString() { return name + ": " + addr; }
}
}

```

[プログラム 4]

```
public interface SearchCriteria { }
```

[プログラム 5]

```
public interface Searchable<T extends SearchCriteria> {
    public boolean meets(T criteria);
}
```

[プログラム 6]

```
public class Test {
    public static void main(String[] args) {
```

```
AddressBook addrbook = new AddressBook();
addrbook.add(new Name("情報", "太郎"),
            new Address(102, 4567, "東京都千代田区"));
addrbook.add(new Name("情報", "花子"),
            new Address(102, 4567, "東京都千代田区"));
addrbook.add(new Name("技術", "太郎"),
            new Address(225, 1234, "横浜市青葉区"));
addrbook.add(new Name("試験", "一朗"),
            new Address(980, 9876, "仙台市青葉区"));
System.out.println(addrbook.meetsAnyOf(
    new Name.Criteria(null, "太郎"),
    new Address.Criteria(-1, -1, "青葉区")));
System.out.println(addrbook.meetsAllOf(
    new Name.Criteria(null, "太郎"),
    new Address.Criteria(-1, -1, "青葉区")));
}
}
```

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 SearchCriteria         | イ SearchCriteria<Name> |
| ウ SearchCriteria<String> | エ Searchable           |
| オ Searchable<Name>       | カ Searchable<String>   |

b, cに関する解答群

- |                        |                     |                      |
|------------------------|---------------------|----------------------|
| ア book.add(entry)      | イ book.add(sc)      | ウ book.remove(entry) |
| エ book.remove(sc)      | オ result.add(entry) | カ result.add(sc)     |
| キ result.remove(entry) | ク result.remove(sc) |                      |

d, eに関する解答群

- |                    |                  |                  |
|--------------------|------------------|------------------|
| ア Address.Criteria | イ Criteria       | ウ Entry.Criteria |
| エ Name.Criteria    | オ SearchCriteria |                  |

設問2 クラス Testにおいて、住所録 addrbook に登録されている全エントリを取得する方法として正しい答えを、解答群の中から二つ選べ。ここで、プログラム中の [ ] には、正しい答えが入っているものとする。

解答群

- |  |
|--|
| ア addrbook.meetsAllOf(new Name.Criteria("", ""))     |
| イ addrbook.meetsAllOf(new Name.Criteria(null, null)) |
| ウ addrbook.meetsAllOf(null)                          |
| エ addrbook.meetsAnyOf(new Name.Criteria("", ""))     |
| オ addrbook.meetsAnyOf(new Name.Criteria(null, null)) |
| カ addrbook.meetsAnyOf(null)                          |

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, 2 に答えよ。

[プログラム 1 の説明]

除算を行う副プログラム DIV である。

- (1) DIV は表 1 に示すレジスタを介して、被除数と除数を受け取り、商と剰余を返す。数値は全て 32 ビットの符号なし整数とし、除数は 0 でないものとする。

表 1 パラメタの受渡し方法

		上位語（上位 16 ビット）	下位語（下位 16 ビット）
入力	被除数	GR1	GR2
	除数	連続する 2 語に、上位 16 ビット、下位 16 ビットの順に格納し、上位語のアドレスを GR3 に設定	
出力	商	GR1	GR2
	剰余	GR4	GR5

- (2) 副プログラム DIV から戻るとき、汎用レジスタ GR6, GR7 の内容は元に戻す。

[プログラム 1]

DIV      START    ; 減算を用いた 32 ビット除算  
          PUSH  0,GR6  
          PUSH  0,GR7  
          LD    GR6,GR1  
          LD    GR7,GR2  
          LD    GR1,=#FFFF    ; 商の初期化  
          LD    GR2,=#FFFF  
LP        LD    GR4,GR6  
          LD    GR5,GR7  
          ADDL GR2,=1    ; 商のカウントアップ  
          JOV   ADJ1  
          JUMP CONT  
ADJ1     a  
CONT     SUBL GR6,0,GR3  
          JOV   FIN  
          SUBL GR7,1,GR3  
          b  
          JUMP LP  
ADJ2     SUBL GR6,=1  
          JOV   FIN  
          JUMP LP  
FIN      POP   GR7

```
POP      GR6  
RET  
END
```

設問 1 プログラム 1 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 ADDL GR1,0,GR3	イ ADDL GR1,=1	ウ ADDL GR2,1,GR3
エ ADDL GR2,=1	オ ADDL GR4,0,GR3	カ ADDL GR5,1,GR3

b に関する解答群

ア JMI ADJ2	イ JMI CONT	ウ JOV ADJ2
エ JOV CONT	オ JPL ADJ2	カ JPL CONT

設問 2 10 進表記で 0 ~ 99999 の値の整数を 32 ビット符号なし整数として与えたとき、  
DIV を用いて 10 進数文字列に変換する副プログラム BTOD を異なるアルゴリズムで 2 種類作成した。プログラム 2 は 10 進表記の上位桁から求めるものであり、  
プログラム 3 は 10 進表記の下位桁から求めるものである。プログラム 2、プログラム 3 中の [ ] に入る正しい答えを、解答群の中から選べ。

- (1) BTOD は 32 ビット符号なし整数の上位語を GR1 に、下位語を GR2 に、変換結果を格納する領域の先頭アドレスを GR3 に設定して呼び出される。
- (2) 変換後の 10 進数文字列は 5 語の領域に格納される。ただし、5 衍に満たない場合は上位衍に “0” が補われる。対象数値が 16 進表記で 0000054D の場合の実行結果を、次に示す。

GR1	0000
GR2	054D

(GR3) → [ '0' '1' '3' '5' '7' ]

- (3) 副プログラム BTOD 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 2]

```

BTOD      START
          RPUSH
          LD      GR6,GR3
          LAD    GR7,4,GR3
          LAD    GR3,UDAT      ; 除数の初期設定
LP       CALL  DIV
          OR     [c],='0'
          ST     [c],0,GR6   ; 10進数字1文字格納
          CPL   GR6,GR7
          JZE   FIN
          LAD   GR6,1,GR6
          LD    GR1,GR4      ; 被除数の再設定
          LD    GR2,GR5
          [d]
          JUMP  LP
FIN      RPOP
          RET
UDAT    DC    0
          DC    10000
          DC    0
          DC    1000
          DC    0
          DC    100
          DC    0
          DC    10
          DC    0
          DC    1
          END

```

[プログラム 3]

```

BTOD      START
          RPUSH
          LAD   GR6,4,GR3
          LD    GR7,GR3
          LAD   GR3,DAT
LP       CALL  DIV
          OR     [e],='0'
          ST     [e],0,GR6
          CPL   GR6,GR7
          JZE   FIN
          LAD   GR6,-1,GR6
          JUMP  LP
FIN      RPOP
          RET
DAT     DC    0
          DC    [f]
          END

```

c, e に関する解答群

ア GR1 イ GR2 ウ GR3 エ GR4 オ GR5

d に関する解答群

ア LAD GR3,1,GR3	イ LAD GR3,2,GR3	ウ LAD GR7,1,GR7
エ LAD GR7,2,GR7	オ LD GR3,1,GR3	カ LD GR3,2,GR3
キ LD GR7,1,GR7	ク LD GR7,2,GR7	

f に関する解答群

ア 1 イ 10 ウ 100 エ 1000 オ 10000

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、2に答えよ。

〔表計算の説明〕

三つの事業本部（以下、本部という）からなるK社では、次のルールに従って、社員の基本給及び賞与の額を決めている。

(1) 基本給

- ① 基本給は、毎月支給され、次の式で計算する。

$$\text{基本給} = \text{年齢給} + \text{役職給}$$

- ② 年齢給は、4月1日時点の年齢が50歳未満の場合、その年齢に7,000円を乗じて算出され、50歳以上の場合は、固定額350,000円となる。

- ③ 役職給の金額は、表1に示すとおりである。

表1 役職給

役職	役職給（千円）
本部長	200
部長	150
課長	100
主任	50
なし	0

(2) 賞与

- ① 賞与は、次の式で計算する。

$$\text{賞与} = \text{基本給} \times (\text{役職係数} + \text{個人評価係数})$$

- ② 役職係数は、表2に示すとおりである。

- ③ 個人評価係数は、表3に示すとおり個人評価点の範囲によって決まる。

表2 役職係数

役職	役職係数
本部長	3.5
部長	3.0
課長	2.5
主任	2.0
なし	1.5

表3 個人評価係数

個人評価点の範囲	個人評価係数
1点～20点	0.75
21点～40点	0.85
41点～60点	1.00
61点～80点	1.15
81点～100点	1.25

[ワークシート：役職給]

基本給の計算の基となる役職ごとの役職給を登録した図 1 のワークシート “役職給” を作成した。

	A	B	C
1	役職	役職ランク	役職給（千円）
2	本部長	1	200
3	部長	2	150
4	課長	3	100
5	主任	4	50
6	なし	5	0

図 1 ワークシート “役職給”

役職をセル A2～A6 に、役職の序列を示す数値である役職ランクをセル B2～B6 に、役職給をセル C2～C6 に入力する。

[ワークシート：賞与係数]

賞与の計算の基となる役職ごとの役職係数及び個人評価点に対応した個人評価係数を登録した図 2 のワークシート “賞与係数” を作成した。

	A	B	C
1	役職係数		
2	役職	役職ランク	役職係数
3	本部長	1	3.5
4	部長	2	3.0
5	課長	3	2.5
6	主任	4	2.0
7	なし	5	1.5
8			
9	個人評価係数		
10	個人評価点の範囲の下限値	個人評価係数	
11		1	0.75
12		21	0.85
13		41	1.00
14		61	1.15
15		81	1.25

図 2 ワークシート “賞与係数”

(1) 役職をセル A3～A7 に、役職ランクをセル B3～B7 に、役職係数をセル C3～C7 に入力する。

(2) 個人評価点の範囲の下限値をセル A11～A15 に、個人評価係数をセル B11～B15 に入力する。

[ワークシート：基本給・賞与計算]

K 社の三つの本部（第 1 本部、第 2 本部、第 3 本部）に所属する社員 107 名の基本給及び賞与を計算するためのワークシート“基本給・賞与計算”を作成した。そのワークシートを図 3 に示す。

	A	B	C	D	E	F	G	H	I	J	K	L
	社員番号	役職	役職ランク	本部	年齢	役職給(千円)	年齢給(千円)	基本給(千円)	役職係数	個人評価点	個人評価係数	賞与(千円)
2	H0006	本部長	1	第1本部	56	200	350	550	3.5	22	0.85	2,393
3	H0011	本部長	1	第2本部	54	200	350	550	3.5	60	1.00	2,475
4	H0003	本部長	1	第3本部	58	200	350	550	3.5	60	1.00	2,475
5	H0010	部長	2	第1本部	54	150	350	500	3.0	66	1.15	2,075
6	H0018	部長	2	第1本部	50	150	350	500	3.0	34	0.85	1,925
7	H0008	部長	2	第2本部	55	150	350	500	3.0	72	1.15	2,075
	:	:	:	:	:	:	:	:	:	:	:	:
25	H0067	課長	3	第2本部	35	100	245	345	2.5	60	1.00	1,208
26	H0009	課長	3	第3本部	55	100	350	450	2.5	40	0.85	1,508
27	H0013	課長	3	第3本部	53	100	350	450	2.5	80	1.15	1,643
28	H0002	主任	4	第1本部	58	50	350	400	2.0	52	1.00	1,200
29	H0016	主任	4	第1本部	51	50	350	400	2.0	98	1.25	1,300
	:	:	:	:	:	:	:	:	:	:	:	:
107	H0106	なし	5	第3本部	23	0	161	161	1.5	36	0.85	378
108	H0107	なし	5	第3本部	23	0	161	161	1.5	20	0.75	362

注記 賞与は、小数第 1 位を四捨五入した値を表示している。

図 3 ワークシート“基本給・賞与計算”

設問 1 ワークシート“基本給・賞与計算”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) 社員番号をセル A2～A108 に、役職をセル B2～B108、本部をセル D2～D108 に、4月1日時点での年齢をセル E2～E108 に入力する。役職ランクを参照する計算式をセル C2～C108 に入力する。
- (2) 基本給を求めるための計算式を入力する。
- ① 役職給を求める計算式  をセル F2 に入力し、セル F3～F108 に複写する。
  - ② 年齢給を求める計算式  をセル G2 に入力し、セル G3～G108 に複写する。
  - ③ 基本給を求める計算式をセル H2～H108 に入力する。
- (3) 賞与を求めるための計算式を入力する。
- ① 役職係数を求める計算式をセル I2～I108 に入力する。
  - ② 個人評価点をセル J2～J108 に入力する。
  - ③ 個人評価点に対応した個人評価係数を求めるための計算式  をセル K2 に入力し、セル K3～K108 に複写する。
  - ④ 賞与を求める計算式をセル L2～L108 に入力する。

#### aに関する解答群

- ア 垂直照合(B1, 役職給!B\$2～C\$6, 2, 0)
- イ 垂直照合(B1, 役職給!B\$2～C\$6, 3, 0)
- ウ 垂直照合(C2, 役職給!A\$2～C\$6, 2, 0)
- エ 垂直照合(C2, 役職給!B\$2～C\$6, 2, 0)
- オ 水平照合(B1, 役職給!B\$2～C\$6, 2, 0)
- カ 水平照合(B1, 役職給!B\$2～C\$6, 3, 0)
- キ 水平照合(C2, 役職給!A\$2～C\$6, 2, 0)
- ク 水平照合(C2, 役職給!B\$2～C\$6, 2, 0)

#### bに関する解答群

- |                          |                          |
|--------------------------|--------------------------|
| ア IF(E2<49, C2 * 7, 350) | イ IF(E2<49, E2 * 7, 350) |
| ウ IF(E2<50, C2 * 7, 350) | エ IF(E2<50, E2 * 7, 350) |
| オ IF(E2≤51, C2 * 7, 350) | カ IF(E2≤51, E2 * 7, 350) |
| キ IF(E2≥50, C2 * 7, 350) | ク IF(E2≥50, E2 * 7, 350) |

### c に関する解答群

- ア 垂直照合(J2, 賞与係数!A\$11～B\$15, 2, 0)
- イ 垂直照合(J2, 賞与係数!A\$11～B\$15, 2, 1)
- ウ 垂直照合(J2, 賞与係数!B\$3～C\$7, 2, 0)
- エ 垂直照合(J2, 賞与係数!B\$3～C\$7, 2, 1)
- オ 水平照合(J2, 賞与係数!A\$11～B\$15, 2, 0)
- カ 水平照合(J2, 賞与係数!A\$11～B\$15, 2, 1)
- キ 水平照合(J2, 賞与係数!B\$3～C\$7, 2, 0)
- ク 水平照合(J2, 賞与係数!B\$3～C\$7, 2, 1)

設問2 賞与の見直し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。

K社では、役職及び個人評価に加えて、本部ごとの利益（以下、本部利益という）を賞与に反映するために、本部利益に応じた評価点（以下、本部加点という）を加味するよう賞与の計算式を見直すことにした。

#### [見直した賞与計算方式に関する説明]

(1) 見直した賞与計算式は次のとおりである。

① 役職が本部長、部長又は課長のとき

$$\text{賞与} = \text{基本給} \times (\text{役職係数} + \text{個人評価係数} + \text{本部加点} / 10)$$

② 役職が主任又はなしのとき

$$\text{賞与} = \text{基本給} \times (\text{役職係数} + \text{個人評価係数} + \text{本部加点} / 20)$$

(2) 本部に属する社員の賞与の合計（以下、本部賞与合計という）が、本部利益の 15 % の金額（以下、本部賞与合計上限という）以下となる条件を満足する 1～20 の整数があるときは、その最大値を本部加点とする。無いときは、本部加点を 0 とする。

条件を満足する本部加点を求め、それを加味した賞与計算を行うために、ワークシート“新賞与計算”とマクロ BonusCalc を作成した。そのワークシート及びマクロ BonusCalc の実行結果の表示例を図 4 に示す。

	A	B	C	D	…	H	I	J	K	L	M
1	…						本部	本部利益 (千円)	本部賞与合計 上限(千円)	本部 加点	本部賞与 合計(千円)
2	…						第1本部	298,060	44,709	20	43,234
3	…						第2本部	227,560	34,134	15	33,854
4	…						第3本部	358,760	53,814	6	53,071
5	社員 番号	役職	役職 ランク	本部	…	基本給 (千円)	役職 係数	個人 評価点	個人 評価係数	本部 加点	賞与 (千円)
6	H0006	本部長	1	第1本部	…	550	3.5	22	0.85	20	3,493
7	H0011	本部長	1	第2本部	…	550	3.5	60	1.00	15	3,300
8	H0003	本部長	1	第3本部	…	550	3.5	60	1.00	6	2,805
9	H0010	部長	2	第1本部	…	500	3.0	66	1.15	20	3,075
10	H0018	部長	2	第1本部	…	500	3.0	34	0.85	20	2,925
11	H0008	部長	2	第2本部	…	500	3.0	72	1.15	15	2,825
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
29	H0067	課長	3	第2本部	…	345	2.5	60	1.00	15	1,725
30	H0009	課長	3	第3本部	…	450	2.5	40	0.85	6	1,778
31	H0013	課長	3	第3本部	…	450	2.5	80	1.15	6	1,913
32	H0002	主任	4	第1本部	…	400	2.0	52	1.00	20	1,600
33	H0016	主任	4	第1本部	…	400	2.0	98	1.25	20	1,700
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
111	H0106	なし	5	第3本部	…	161	1.5	36	0.85	6	427
112	H0107	なし	5	第3本部	…	161	1.5	20	0.75	6	411

注記 本部賞与合計上限、本部賞与合計及び賞与は、小数第1位を四捨五入した値を表示している。

図4 ワークシート“新賞与計算”

#### [ワークシート：新賞与計算]

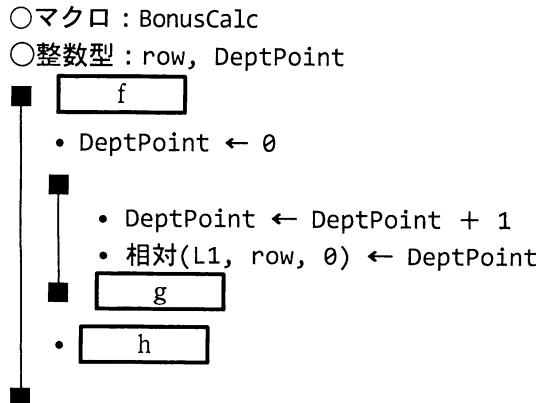
- (1) 本部名をセル I2～I4 に、本部利益をセル J2～J4 に、本部賞与合計上限を求める計算式をセル K2～K4 に入力する。
- (2) ワークシート“基本給・賞与計算”のセル A1～K108 をワークシート“新賞与計算”のセル A5～K112 に複写する。
- (3) 所属本部によって決まる社員ごとの本部加点を求める計算式をセル L6～L112 に入力する。社員ごとの賞与を求める計算式 d をセル M6 に入力し、セル M7～M112 に複写する。

- (4) 本部ごとの賞与合計を求める計算式  e  をセル M2 に入力し、セル M3～M4 に複写する。

[マクロの説明]

- (1) マクロ BonusCalc は、各本部の条件を満足する本部加点を求めるマクロである。
- (2) マクロ BonusCalc の処理手順は次のとおりである。
- ① 第1本部の本部加点の初期値を0とする。
  - ② 本部加点に1を加算する。
  - ③ 手順②で設定された本部加点に基づき、第1本部に所属する社員の賞与の合計である第1本部の本部賞与合計をワークシートの計算式から算出する。
  - ④ 第1本部の本部賞与合計と本部賞与合計上限の値を比較し、前者が後者を上回る、又は、本部加点が20を超えたときは、手順⑤に進む。それ以外のときは、手順②に戻る。
  - ⑤ このときの本部加点から1を減じた値を第1本部の本部加点とし、対応するセルに代入する。
  - ⑥ 同様に、第2本部及び第3本部の本部加点をそれぞれ対応するセルに代入する。

[マクロ : BonusCalc]



#### dに関する解答群

- ア  $H6 * (I6 + K6) + IF(C6 < 3, L6 / 10, L6 / 20)$
- イ  $H6 * (I6 + K6) + IF(C6 < 3, L6 / 20, L6 / 10)$
- ウ  $H6 * (I6 + K6) + IF(C6 \leq 3, L6 / 10, L6 / 20)$
- エ  $H6 * (I6 + K6) + IF(C6 \leq 3, L6 / 20, L6 / 10)$
- オ  $H6 * (I6 + K6 + IF(C6 < 3, L6 / 10, L6 / 20))$
- カ  $H6 * (I6 + K6 + IF(C6 < 3, L6 / 20, L6 / 10))$
- キ  $H6 * (I6 + K6 + IF(C6 \leq 3, L6 / 10, L6 / 20))$
- ク  $H6 * (I6 + K6 + IF(C6 \leq 3, L6 / 20, L6 / 10))$

#### eに関する解答群

- ア 条件付合計( $C\$6 \sim C\$112, = I2, L\$6 \sim L\$112$ )
- イ 条件付合計( $C\$6 \sim C\$112, = I2, M\$6 \sim M\$112$ )
- ウ 条件付合計( $D\$6 \sim D\$112, = I2, L\$6 \sim L\$112$ )
- エ 条件付合計( $D\$6 \sim D\$112, = I2, M\$6 \sim M\$112$ )
- オ 照合検索( $I2, C\$6 \sim C\$112, L\$6 \sim L\$112$ )
- カ 照合検索( $I2, C\$6 \sim C\$112, M\$6 \sim M\$112$ )
- キ 照合検索( $I2, D\$6 \sim D\$112, L\$6 \sim L\$112$ )
- ク 照合検索( $I2, D\$6 \sim D\$112, M\$6 \sim M\$112$ )

#### fに関する解答群

- |                           |                             |
|---------------------------|-----------------------------|
| ア $row: 1, row < 3, 1$    | イ $row: 1, row < 20, 1$     |
| ウ $row: 1, row \leq 3, 1$ | エ $row: 1, row \leq 20, 1$  |
| オ $row: 2, row < 4, 1$    | カ $row: 2, row \leq 4, 1$   |
| キ $row: 6, row < 112, 1$  | ク $row: 6, row \leq 112, 1$ |

g に関する解答群

- ア 論理積( $M2 \leq K2$ ,  $DeptPoint \leq 20$ )
- イ 論理積( $M2 \leq K2$ ,  $L2 \leq 20$ )
- ウ 論理積(相対( $L1$ ,  $row$ , 1) > 相対( $L2$ ,  $row$ , -1),  $DeptPoint > 20$ )
- エ 論理積(相対( $L1$ ,  $row$ , 1)  $\leq$  相対( $L1$ ,  $row$ , -1),  $DeptPoint \leq 20$ )
- オ 論理和( $M2 > K2$ ,  $DeptPoint > 20$ )
- カ 論理和( $M2 > K2$ ,  $L2 > 20$ )
- キ 論理和(相対( $L1$ ,  $row$ , 1) > 相対( $L2$ ,  $row$ , -1),  $DeptPoint > 20$ )
- ク 論理和(相対( $L1$ ,  $row$ , 1)  $\leq$  相対( $L1$ ,  $row$ , -1),  $DeptPoint \leq 20$ )

h に関する解答群

- ア 相対( $L1$ ,  $row$ , 0)  $\leftarrow DeptPoint$
- イ 相対( $L1$ ,  $row$ , 0)  $\leftarrow DeptPoint + 1$
- ウ 相対( $L1$ ,  $row$ , 0)  $\leftarrow DeptPoint - 1$
- エ 相対( $L1$ ,  $row$ , -1)  $\leftarrow DeptPoint$
- オ 相対( $L1$ ,  $row$ , -1)  $\leftarrow DeptPoint + 1$
- カ 相対( $L1$ ,  $row$ , -1)  $\leftarrow DeptPoint - 1$
- キ 相対( $L2$ ,  $row$ , 0)  $\leftarrow DeptPoint$
- ク 相対( $L2$ ,  $row$ , 0)  $\leftarrow DeptPoint + 1$
- ケ 相対( $L2$ ,  $row$ , -1)  $\leftarrow DeptPoint + 1$
- コ 相対( $L2$ ,  $row$ , -1)  $\leftarrow DeptPoint - 1$

## ■ Java プログラムで使用する API の説明

```
java.util
public interface Set<E>
    型 E の要素を集合（セット）として管理するインターフェースを提供する。
    インタフェース Collection を継承する。
```

### メソッド

```
public boolean add(E e)
    指定された要素が集合に含まれていなければ、集合に追加する。
    引数： e — 集合に追加される要素
    戻り値：指定された要素が集合に含まれていなければ true
            それ以外は false
```

```
public boolean remove(Object o)
    指定された要素が集合に含まれていれば、集合から削除する。
    引数： o — 集合から削除する要素
    戻り値：指定された要素が集合に含まれていれば true
            それ以外は false
```

```
java.util
public class HashSet<E>
    インタフェース Set のハッシュを用いた実装である。
```

### コンストラクタ

```
public HashSet()
    空の HashSet を作る。
```

```
public HashSet(Collection<? extends E> c)
    指定されたコレクションの全要素を含む HashSet を作る。
    引数： c — コレクション
```

### メソッド

```
public boolean add(E e)
    インタフェース Set のメソッド add と同じ
```

```
public boolean remove(Object o)
    インタフェース Set のメソッド remove と同じ
```

```
java.lang
public final class String
    クラス String は文字列を表す。なお、String は、インターフェース CharSequence を実装
    している。
```

#### メソッド

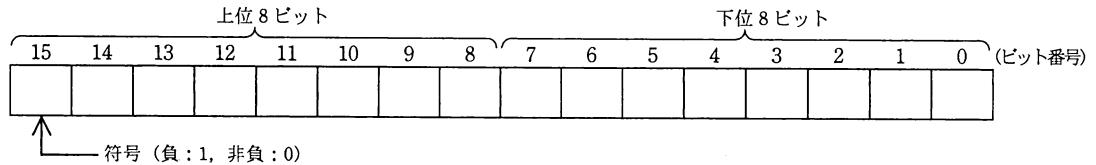
```
public boolean contains(CharSequence s)
    この文字列が指定された文字の並びを含むかどうかを判定する。
    引数： s — 文字の並び
    戻り値：この文字列が指定された文字の並びを含んでいれば true
            それ以外は false
```

## ■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 $\frac{r1, r2}{r, adr [,x]}$	$r1 \leftarrow (r2)$ $r \leftarrow (\text{実効アドレス})$	<input type="radio"/> *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	$r_1, r_2$ $r, \text{adr} [, x]$	$r_1 \leftarrow (r_1) + (r_2)$ $r \leftarrow (r) + (\text{実効アドレス})$	○
論理加算 ADD Logical	ADDL	$r_1, r_2$ $r, \text{adr} [, x]$	$r_1 \leftarrow (r_1) +_L (r_2)$ $r \leftarrow (r) +_L (\text{実効アドレス})$	
算術減算 SUBtract Arithmetic	SUBA	$r_1, r_2$ $r, \text{adr} [, x]$	$r_1 \leftarrow (r_1) - (r_2)$ $r \leftarrow (r) - (\text{実効アドレス})$	○
論理減算 SUBtract Logical	SUBL	$r_1, r_2$ $r, \text{adr} [, x]$	$r_1 \leftarrow (r_1) -_L (r_2)$ $r \leftarrow (r) -_L (\text{実効アドレス})$	
論理積 AND	AND	$r_1, r_2$ $r, \text{adr} [, x]$	$r_1 \leftarrow (r_1) \text{ AND } (r_2)$ $r \leftarrow (r) \text{ AND } (\text{実効アドレス})$	○*1
論理和 OR	OR	$r_1, r_2$ $r, \text{adr} [, x]$	$r_1 \leftarrow (r_1) \text{ OR } (r_2)$ $r \leftarrow (r) \text{ OR } (\text{実効アドレス})$	
排他的論理和 eXclusive OR	XOR	$r_1, r_2$ $r, \text{adr} [, x]$	$r_1 \leftarrow (r_1) \text{ XOR } (r_2)$ $r \leftarrow (r) \text{ XOR } (\text{実効アドレス})$	

(3) 比較演算命令

算術比較 ComPare Arithmetic	CPA	$r_1, r_2$ $r, \text{adr} [, x]$	( $r_1$ ) と ( $r_2$ ) , 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	$r_1, r_2$ $r, \text{adr} [, x]$	<table border="1"> <thead> <tr> <th rowspan="2">比較結果</th> <th colspan="2">FRの値</th> </tr> <tr> <th>SF</th> <th>ZF</th> </tr> </thead> <tbody> <tr> <td>(<math>r_1 &gt; r_2</math>)</td> <td>0</td> <td>0</td> </tr> <tr> <td>(<math>r &gt; (\text{実効アドレス})</math>)</td> <td>0</td> <td>1</td> </tr> <tr> <td>(<math>r_1 = r_2</math>)</td> <td>0</td> <td>1</td> </tr> <tr> <td>(<math>r = (\text{実効アドレス})</math>)</td> <td>1</td> <td>0</td> </tr> <tr> <td>(<math>r_1 &lt; r_2</math>)</td> <td>1</td> <td>0</td> </tr> <tr> <td>(<math>r &lt; (\text{実効アドレス})</math>)</td> <td>—</td> <td>—</td> </tr> </tbody> </table>		比較結果	FRの値		SF	ZF	( $r_1 > r_2$ )	0	0	( $r > (\text{実効アドレス})$ )	0	1	( $r_1 = r_2$ )	0	1	( $r = (\text{実効アドレス})$ )	1	0	( $r_1 < r_2$ )	1	0	( $r < (\text{実効アドレス})$ )	—
比較結果	FRの値																									
	SF	ZF																								
( $r_1 > r_2$ )	0	0																								
( $r > (\text{実効アドレス})$ )	0	1																								
( $r_1 = r_2$ )	0	1																								
( $r = (\text{実効アドレス})$ )	1	0																								
( $r_1 < r_2$ )	1	0																								
( $r < (\text{実効アドレス})$ )	—	—																								

(4) シフト演算命令

算術左シフト Shift Left Arithmetic	SLA	$r, \text{adr} [, x]$	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	$r, \text{adr} [, x]$	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。	
論理左シフト Shift Left Logical	SLL	$r, \text{adr} [, x]$	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。	○*2
論理右シフト Shift Right Logical	SRL	$r, \text{adr} [, x]$		

(5) 分岐命令

正分岐 Jump on Plus	JPL	$\text{adr} [, x]$	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	$\text{adr} [, x]$	命令	
非零分岐 Jump on Non Zero	JNZ	$\text{adr} [, x]$	分岐するときの FR の値	
零分岐 Jump on Zero	JZE	$\text{adr} [, x]$	OF SF ZF	
オーバフロー分岐 Jump on Overflow	JOV	$\text{adr} [, x]$	JPL 0 0	
無条件分岐 unconditional JUMP	JUMP	$\text{adr} [, x]$	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]	$SP \leftarrow (SP) -_L 1,$ $(SP) \leftarrow$ 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

スーパーバイザコール 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	o
15	/	?	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] ]
命令行	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）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 $\leq$ 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>書式</sup>罫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) 列と行と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う。  
〔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、セル範囲という。  
〔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	( )	↑ ↓
べき乗演算	^	
単項演算	+ , -	
乗除演算	* , /	
加減演算	+ , -	
比較演算	>, <, ≥, ≤, =, ≠	

#### 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)	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] 組合('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、合計のセル範囲内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。

<sup>2)</sup> 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 の中での順位を調べ、その順位をセル B5, B6, …, B9 に順に代入する。

[ メモ用紙 ]

7. 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
--------	-------------

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。
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、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、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。
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。

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。