

平成 13 年度 春期 ソフトウェア開発技術者 午後 問題

問 1 通信ネットワークに関する次の記述を読んで、設問 1～3 に答えよ。

S 社のネットワークは、LAN1～3 の三つの支線と主系及び従系の二つの幹線から成っている。支線は通信速度が 10M ビット / 秒で、パソコン、サーバ、プリンタなど（以下、総称して PC という）が接続されている。幹線は通信速度が 100M ビット / 秒で、ルータを介して各支線と接続されている。従系幹線は予備であり、通常は主系幹線を使用する（図参照）。

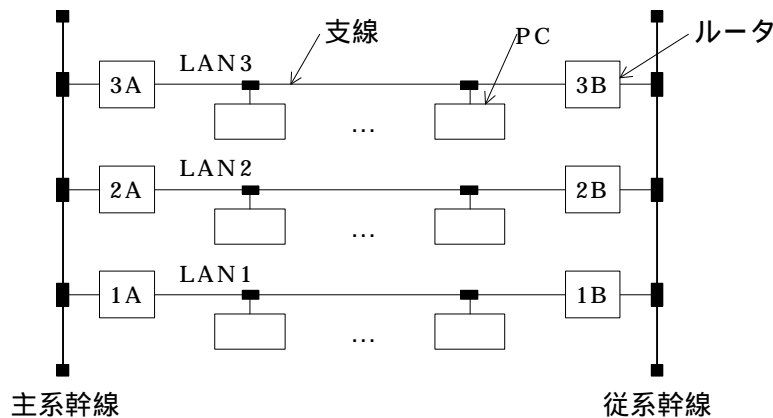


図 S 社のネットワーク

ルータ同士は経路情報を交換し合っており、それぞれのルータはほかのルータから通知される経路情報を常時モニタしている。あるルータからの通知が一定時間以上途切れるとそのルータは故障したものと見なす。

ルータ同士が交換し合う経路情報には、ネットワーク上の距離を整数で表したホップ数が含まれている。ホップ数とは、ある支線から最初にそのルータを経由してほかの支線に到達するまでの間に通過しなければならない最少のルータ数である。この際の通過経路は、一度通った支線又は幹線を再度通ることがないように選ばれる。

ある支線の PC がほかの支線の PC と通信する場合、PC は最少のホップ数で通信できるルータを選択する。そのため、ルータはそのルータに接続している支線の PC に経路情報を通知する。すべてのルータが正常な場合、主系のルータ（1A、2A、3A）だけが経路情報を PC に通知し、従系のルータ（1B、2B、3B）は通知しない。しかし、ほかのルータの故障を検出すると、従系のルータも経路情報を通知し始める。

ルータが PC に通知する経路情報は、“ 目的の支線名：ホップ数 ” で表す。例えば、すべてのルータが正常な場合、ルータ 1A が LAN1 の PC に通知する経路情報は、LAN2：2、LAN3：2 になる。

なお、各支線相互間（各支線内も含む）で転送される平均データ量は表 1 のとおりである。

表 1 各視線間の平均データ量

単位 Mビット/秒

支線	LAN1 向け	LAN2 向け	LAN3 向け
LAN1 から	0.3	1.0	0.6
LAN2 から	0.1	1.5	0.2
LAN3 から	0.5	1.5	0.2

設問 1 次の表は各支線の平均データ量である。  ~  に入れる適切な数値を答えよ。

単位 Mビット/秒

支線	平均データ量
LAN 1	<input type="text" value="ア"/>
LAN 2	<input type="text" value="イ"/>
LAN 3	<input type="text" value="ウ"/>

設問 2 経路情報に関する次の記述中の  ~  に入れる適切な数値を答えよ。

ルータ 1A が故障すると、一定時間後、従系のルータも各支線の PC に経路情報を通知し始める。このとき、ルータ 2A が LAN2 の PC に通知する経路情報は、LAN1: , LAN3:  である。同様に、ルータ 2B が LAN2 の PC に通知する経路情報は、LAN1: , LAN3:  である。

設問 3 ルータ 1A とほかの一つのルータが故障し、一部の支線でレスポンスが悪化したことがある。次の記述は、このとき発生したレスポンス悪化に関する考察の一部である。

レスポンス悪化に関する次の記述中の  ,  に入れる適切なルータ名と、 に入れる適切な支線の番号を答えよ。

なお、このネットワークは、データ量が支線で 5M ビット / 秒以上、幹線で 40M ビット / 秒以上になると、急激にレスポンスが悪化する。

ルータ 1A が故障して、一定時間経過すると、従系のルータも経路情報を支線の PC に送り始め、その結果、従系幹線が使用できるようになる。したがって、ルータ 2A, 3A などの主系のルータが故障しても、従系を使用できるので通信上の問題は発生しない。

ある支線から目的とする別の支線への通信を行う場合、ルータ 1 台だけの故障であれば、“送信側支線 - 幹線 - 目的とする支線” という経路で通信することができる。しかし、ある支線の主系ルータとほかの支線の従系ルータが故障すると、一部の支線間では“送信側支線 - 幹線 - う回する支線 - 幹線 - 目的とする支線” という経路で送信することになる。

ルータ 1A 以外に従系のルータにも故障が発生し、その結果、う回路に使用された支線に大量のデータが流れたからレスポンスが悪化したと考えられる。

ルータ  が故障した場合、LAN1 から LAN2, LAN2 から LAN1 向けのデータはすべて LAN3 をう回する。同様にルータ  が故障した場合 LAN1 から LAN3, LAN3 から LAN1 向けのデータはすべて LAN2 をう回する。このことから、LAN  のデータ量が 5M ビット / 秒を超え、レスポンスが悪化したものと考えられる。

問 2 オブジェクト指向設計に関する次の記述を読んで、設問 1～3 に答えよ。

図 1 の貸出カードを使って、本と CD の貸出しと返却の管理を行っている図書館がある。この図書館における貸出管理のシステムをオブジェクト指向で開発することになった。まず、扱っている CD に着目し、図 2 に示す“本”クラスと“CD”クラスを定義した。図 2 は、各クラスがタイトルなどの属性と、それらを参照する操作をもっていることを示す。図 3 にクラスの定義方法と、クラス間の関係の定義方法を示す。

タイトル _____		
貸出日	返却日	利用者

図 1 貸出カード

本	CD
タイトル	タイトル
著者	作曲者
キーワード	指揮者
タイトルを参照する	演奏者
著者を参照する	タイトルを参照する
キーワードを参照する	作曲者を参照する
	指揮者を参照する
	演奏者を参照する

図 2 “本”クラスと“CD”クラスの定義

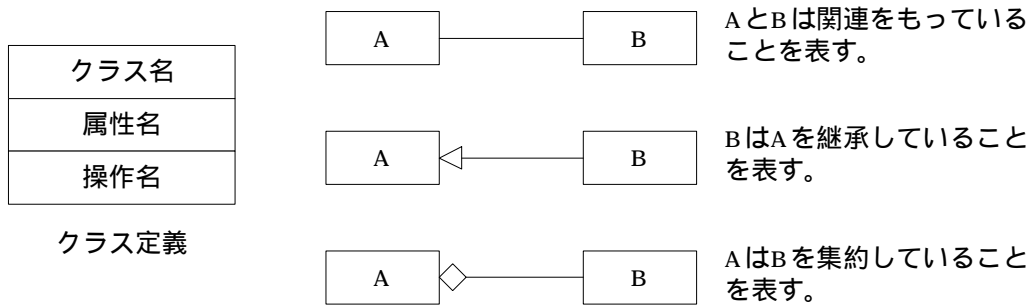


図3 クラスの定義方法と関係の定義方法

設問1 “本”クラスと“CD”クラス共通のスーパークラスとして“貸出物”クラスを設けることにした。クラス間の構造を表現した図4を完成させよ。スーパークラスで定義した方がよいと考えられるものはすべてスーパークラスで定義し、サブクラスでは定義しないものとする。属性名及び操作名は、図2にあるものを使用すること。

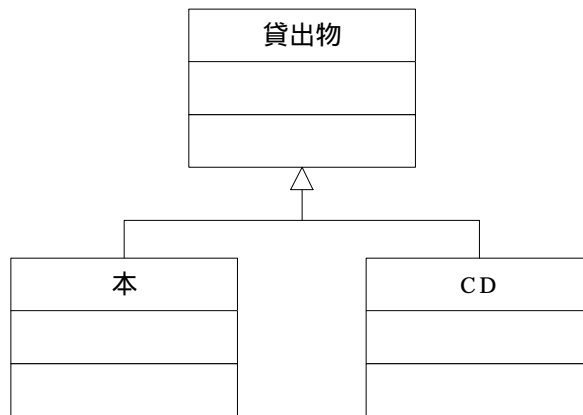
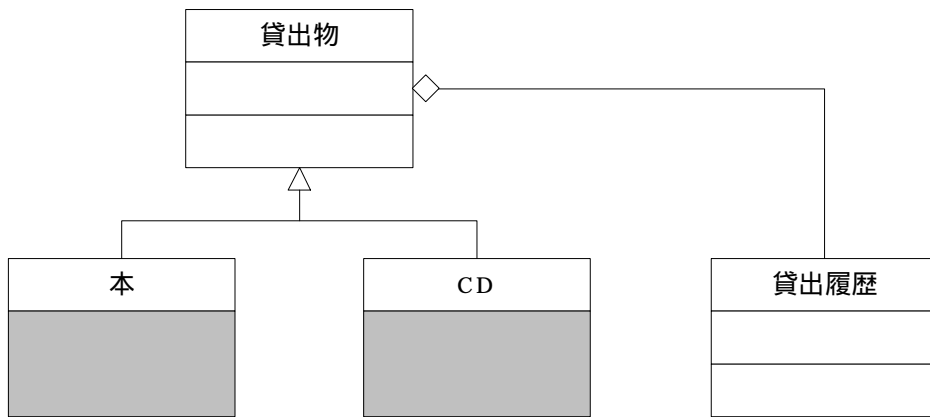


図4 クラス構造(1)

設問2 貸出履歴を管理するために、図1の貸出カードの項目である貸出日、返却日及び利用者をもつ“貸出履歴”クラスを設けることにした。“貸出物”クラスに対する貸出しや返却の処理はすべて、“貸出物”クラスから“貸出履歴”クラスに委譲されるものとする。図5の“貸出物”クラスと“貸出履歴”クラスの属性名及び操作名欄に適切な属性名と操作名を記入し、これらのクラスの定義を完成させよ。属性名及び操作名は要求されているものだけを書くこと。また、“本”クラスと“CD”クラスには何も書かないこと。



注 網掛け部分には何も書かないこと。

図 5 クラス構造(2)

設問 3 オブジェクト指向プログラミングに関する次の記述中の  ~  に入れる適切な字句を答えよ。

オブジェクト指向プログラミングでは、 化の概念を用いて  を介してデータを扱う。これによって、オブジェクト内部の  構造に変更があっても、呼出し側のプログラムを変更する必要がなく、保守性の向上が期待できる。

図 4 のクラス構造の場合、スーパークラスで定義した操作をサブクラスで再定義することもできる。これを  という。操作の再定義によって、同じ名前でも違った実装をもつことになる。実装が異なっても、それを呼び出すプログラムが違いを意識しなくてよいことを  という。

プログラムによって生成されるオブジェクトの実体は、 領域に確保される。使用されなくなったオブジェクトの管理をプログラムで行うのは困難なので、 によって領域内の不必要なオブジェクトの整理を行い、領域の再利用を図る言語処理系がある。

問 3 暗号化と認証に関する次の記述を読んで、設問 1 ～ 5 に答えよ。

X 社では、インターネット経由で取引先と電子メールやデータ（以下、メッセージという）の交換を行うことを計画している。インターネット経由でメッセージの交換を行う場合、メッセージの本文を暗号化するだけではセキュリティ面で安全とは言えない。メッセージの交換相手の認証も行う必要がある。相手の認証を行うに当たって、暗号化と復号を同一の鍵（共通鍵）で行う秘密鍵方式ではなく、暗号化と復号を各個人が所有する秘密鍵と公開鍵とで行う公開鍵方式の暗号化アルゴリズムを採用することにした。公開鍵方式を使って受信者 B が送信者 A の認証を行う場合の手順を図 1 に示す。

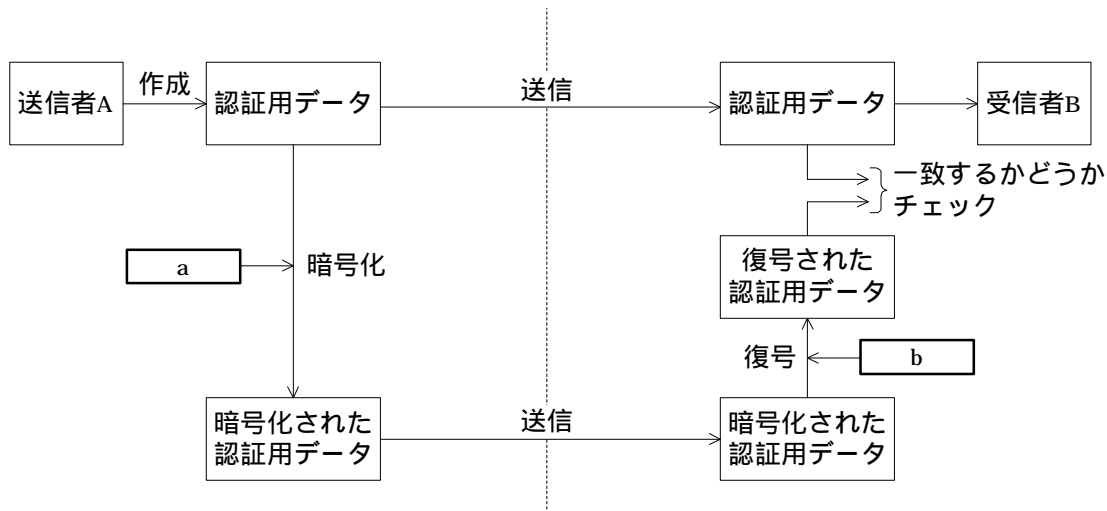


図 1 公開鍵方式による認証の手順

図 1 の手順を応用して認証を可能にしたメッセージ送受信の手順を図 2 に示す。図 2 の

**a** , **b** には、それぞれ図 1 の **a** , **b** と同じ字句が入る。

図 2 に示すメッセージ送受信の手順は次のとおりである。

- (1) 送信者 A は、メッセージ本文を、A と受信者 B が共有している共通鍵を使って秘密鍵方式で暗号化して B へ送る。
- (2) 認証を行うために、A と B とで共有しているハッシュ関数を用いて、A はメッセージダイジェストを生成し、生成したメッセージダイジェストを公開鍵方式で暗号化して B へ送る。ここで、A と B は相手の公開鍵を安全な方法で入手できるものとする。
- (3) B は、受け取ったメッセージダイジェストを公開鍵方式で復号する。さらに、メッセージ本文を共通鍵で復号し、復号した本文からハッシュ関数を使ってメッセージダイジェストを作る。
- (4) B は、A から受け取って復号したメッセージダイジェストと、自分でメッセージ本文から生成したメッセージダイジェストが一致しているかどうかをチェックすることによって、相手を認証する。

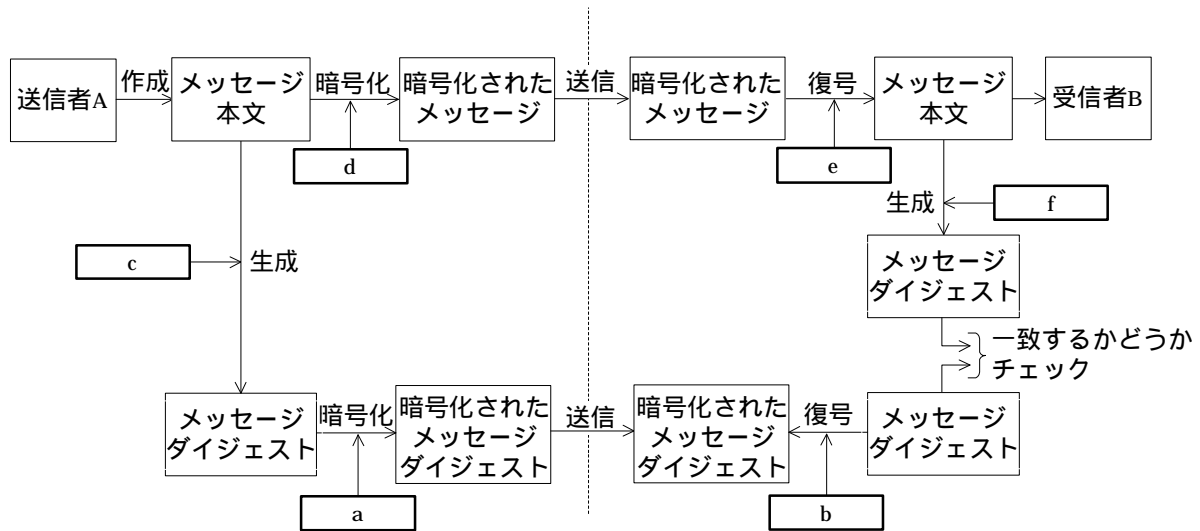


図 2 認証を可能にしたメッセージ送受信の手順

設問 1 図 1 中の  ,  に入れる適切な字句を、それぞれ 10 字以内で答えよ。

設問 2 図 2 中の  ~  に入れる適切な字句を、それぞれ 10 字以内で答えよ。

設問 3 次の記述中の  に入れる適切な字句を 15 字以内で答えよ。

図 2 の手順でメールを送るとき、メッセージ本体の暗号化と、メッセージダイジェストの暗号化では、別々の暗号化方式を用いている。これは、公開鍵方式が秘密鍵方式に比べて、 が長いからである。

設問 4 認証用データにメッセージダイジェストを用いることは、セキュリティ上、何を防止することを目的としているか。10 字以内で答えよ。

設問 5 メッセージダイジェストを生成するハッシュ関数のもつべき特徴を解答欄の中から一つ選び記号で答えよ。

#### 解答群

- ア 異なる入力データから同じ出力データを作成できる。
- イ 入力データから出力データを求める時間が短い。
- ウ 入力データから出力データを求める時間が長い。
- エ 入力されるデータの長さより出力されるデータの長さが長い。
- オ 出力データから入力データを復元できない。



問 4 システムの性能に関する次の記述を読んで、設問 1 ～ 3 に答えよ。

証券業を営む Y 社では、契約・照会システムを構築することになった。このシステムは、1 台のサーバと 100 台のクライアントパソコン（以下、PC という）を LAN で接続し、さらにサーバを二つの証券取引所と専用線で接続するものである（図参照）。

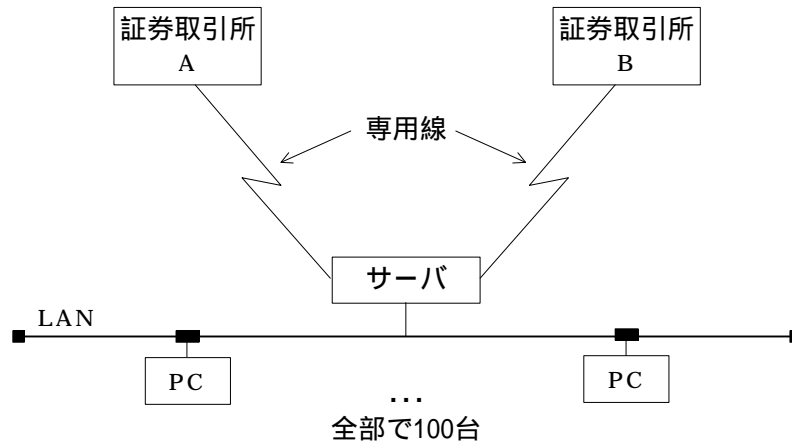


図 Y 社の契約・照会システム

サーバの CPU は増設可能な対称型マルチプロセッサ方式であり、OS は仮想記憶方式である。OS は、CPU の数と関係なく、60M バイトのメモリ領域を必要とする。このシステムでは、証券取引所からのデータ受信と、業務 1 ～ 3 の 3 種類の業務処理の合わせて四つの処理を同時に実行している。データ受信処理はサーバが単独で実行する。一方、業務 1 ～ 3 については、サーバと PC が連携して実行する。PC は共通のクライアントソフトを使用していて、業務 1 ～ 3 のいずれか一つだけを選択して実行する。実行が終了すると、また次の業務を一つ選択して実行する。1 台の PC で複数の業務を同時に実行することはない。

それぞれの処理の概要は、次のとおりである。

(1) 証券取引所からのデータ受信処理

証券取引所 A, B は、それぞれ専用線を介して最新の取引状況を 10 秒ごとにサーバに送る。サーバは、このデータを受信し、編集、解析する。これを処理するタスクが受信タスクである。一つの受信タスクはメモリ領域を 5M バイト使用し、一つの CPU で処理した場合、CPU 処理時間は 200 ミリ秒である。

(2) 業務 1

業務 1 を選択した PC は、証券取引所 A の最新の取引状況をリアルタイムで表示する。サーバ上の業務 1 を処理するタスク(業務 1 タスク)は、証券取引所 A の最新の取引情報を PC に送る。このデータは 1 レコード 64 バイト固定長、合計 2,000 レコードである。サーバは 10 秒以内にすべてのレコードを送る。

業務 1 タスクはメモリ領域を 6M バイト使用し、一つの CPU で処理した場合、CPU 処理時間は 300 ミリ秒である。

(3) 業務 2

業務 2 を選択した PC は、証券取引所 B の最新の取引状況をリアルタイムで表示する。サーバ上の業務 2 を処理するタスク(業務 2 タスク)は、証券取引所 B の最新の取引情報を PC に送る。合計 1,000 レコードであることを除いて、各 PC に送られるデータのレコード長、レコードを送る時間は業務 1 と同じである。

業務 2 タスクはメモリ領域を 6M バイト使用し、一つの CPU で処理した場合、CPU 処理時間は 200 ミリ秒である。

(4) 業務 3

業務 3 を選択した PC は、契約データを処理する。サーバ上の業務 3 を処理するタスク(業務 3 タスク)は、PC から送られる顧客の契約データを処理する。PC とサーバ間のデータは、1 契約当たり 3,600 バイトである。操作時間は 1 契約当たり 30 秒である。

業務 3 タスクはメモリ領域を 10M バイト使用する。しかし、操作時間が 30 秒と長いので CPU 処理時間は無視できる。

サーバは、PC が選択した鍵に合わせて、それに対応する業務タスクを生成する。PC での処理が終了すると、これらのタスクも実行を終了し消滅する。受信タスクについては、サーバ起動時に、証券取引所ごとに、タスクを一つずつ生成する。

最繁忙時には、各業務を実行している PC の数は、業務 1 が 50 台、業務 2 が 30 台、業務 3 が 20 台となる。

このシステムは仮想記憶システムを使っているため、OS 及びタスクの使うメモリ領域すべてが主記憶に割り当てられている必要はない。しかし、OS が必要とするメモリ領域の 80%を、業務タスクが必要とするメモリ領域の 50%を、受信タスクが必要とするメモリ領域の 80%をそれぞれ主記憶に割り当てるとしなないと、処理速度が低下すると予想されている。

設問 1 最繁忙時でも業務処理の速度を低下させないようにするには、少なくとも何 M バイトの主記憶を実装しなければならないか。

設問 2 最繁忙時には、一つの CPU だけでは所定時間内に処理が終了しない。そこで、CPU を追加して所定時間内にすべての処理が終了するようにしたい。サーバが実装しなければならない CPU の数は少なくとも幾つか。ただし、処理速度は CPU の個数に比例するものとする。

設問 3 ネットワークのデータ量及び通信プロトコルについて次の問いに答えよ。

- (1) サーバと PC との通信プロトコルに TCP/IP を使用した理合，LAN に流れるデータ量は最繁時に少なくとも何 M ビット / 秒か。答えは小数第 2 位を四捨五入し，小数第 1 位まで求めよ。  
ここで，LAN にはデータだけが流れるものとし，制御コードなどは無視する。
- (2) 業務 1，2 では，これらの業務を選択している PC に，ほとんど同じデータを送っている。現在，プロトコルに TCP を予定しているが，UDP についても検討することになった。TCP と比較した場合，UDP の特徴として適切なものを解答群からすべて選び記号で答えよ。

解答群

- ア データは固定長方式である。
- イ ブロードキャストが使用でき，同時に複数箇所に送ることができる。
- ウ 受信すると必ず応答パケットを返す。
- エ 信頼性が高い。

問 5 括弧の対応をチェックするプログラムに関する次の記述を読んで、設問 1 ～ 3 に答えよ。

テキストファイル中の括弧の対応をチェックするプログラムのアルゴリズムを考える。対応する括弧が見つからなかった場合は、その括弧の位置を表示する。対応する括弧は同一行になくてもよい。

対応する括弧が見つからない場合には、次の二つのケースがある。

ケース 1：対応する開き括弧が存在しない閉じ括弧が見つかった。対応する開き括弧が見つからないというメッセージと問題が発生した閉じ括弧の位置を表示する。まだ未処理のテキストが残っていれば、残りの部分に対して括弧の対応のチェックを続ける。

ケース 2：すべてを調べ終わった時点で、対応する閉じ括弧が見つからない開き括弧が残った。残っている開き括弧は一つとは限らないので、対応する閉じ括弧が見つからないというメッセージの後に、問題のある開き括弧の位置をすべて表示する。

図 1 のテキストファイルに対して、図 2 のように表示する。

```
(1+2)
abc)
((def)gx))
(((h)
ij)(k
(lml)
```

図 1 テキストファイルの内容

```
対応する開き括弧がありません。
2 行目 4 文字目

対応する開き括弧がありません。
3 行目 10 文字目

対応する閉じ括弧がありません。
5 行目 4 文字目
4 行目 1 文字目
```

図 2 表示結果

ここでは括弧の位置を覚えておくために、整数のスタックを用いる。スタック操作には、push、pop とスタックが空かどうかを調べる empty 及びスタックの一番上のデータの値を返す peek が使えるものとする。

スタック操作以外に、関数 nextch、kind を使うことができる。nextch はファイルから次の 1 文字を読み込む関数で、入力文字を 1 文字読み込んで返すと同時に、読み込んだ文字が何行目の何文字目かを、変数 line と pos に設定する。ただし、ファイルの終わりでは変数 EOF を“真”に設定し、空白を返す。整数 EOF の初期値は“偽”である。また、kind は 1 引数の関数で、文字を引数で受け取り、文字が括弧かどうかを整数値で返す。その関数値は、次のとおりである。

```
kind(c) = 0 c が括弧以外の文字
          1 c が開き括弧
          2 c が閉じ括弧
```

この考察に基づいて書いたアルゴリズムは次のようになった。

〔アルゴリズム 1〕

```
スタックを空に初期化
ch  nextch()
while (not EOF)
  k  kind(ch)
  if (  )
    
    
  elseif (  )
    if (  )
      dummy  pop()
      dummy  pop()
    else
      メッセージを表示 (A)
      何行目(line)と何文字目(pos)を表示
    endif
  endif
  ch  nextch()
endwhile
if (not empty())
  メッセージを表示 (B)
  while (not empty())
    L  pop()
    P  pop()
    何行目(L)と何文字目(P)を表示
  endwhile
endif
```

括弧が 2 種以上あるときには、括弧の種類ごとに対応を調べる必要がある。括弧は(...[...](...))のように完全な入れ子構造になっていることが要求されており、(...[...])のようなものは許されない。

ここでは、3 種類の括弧 { } [] を対象とするようにアルゴリズム 1 を修正した。この場合はスタックに括弧の種類、行番号及び文字位置の三つのデータを積む必要がある。まず、括弧の種類を識別するために、括弧を表のとおりコード化する。kind は、括弧の場合には表のコードを、括弧以外の文字の場合には 0 を関数値として返すように修正した。

表 括弧のコード

(	)	{	}	[	]
1	2	3	4	5	6

閉じ括弧が出できた場合に、異なる種類の開き括弧がスタックの一番上にあったときは、この閉じ括弧に対応する開き括弧がないというメッセージとこの閉じ括弧の位置を表示して、スタック上の開き括弧はそのままにして処理を続けることにする。

これに従うと、アルゴリズムは次のようになる。

〔アルゴリズム 2〕

```
スタックを空に初期化
ch  nextch()
while (not EOF)
    k  kind(ch)
    if (k > 0)
        if (  )
            スタックに情報を積む
        elseif(  and  )
            dummy  pop()
            dummy  pop()
            dummy  pop()
        else
            メッセージを表示 (A)
            何行目(line)と何文字目(pos)を表示
        endif
    endif
    ch  nextch()
endwhile
if (not empty())
    メッセージを表示 (B)
    while (not empty())
        dummy  pop()
        L  pop()
        p  pop()
        何行目(L)と何文字目(p)を表示
    endwhile
endif
```

設問 1 アルゴリズム 1 の  ~  に入れる適切な字句を答えよ。

設問 2 アルゴリズム 1, 2 の(A), (B)の位置で表示するメッセージを答えよ。

設問 3 アルゴリズム 2 の  ~  で行う処理内容を解答群から選び, 記号で答えよ。また, その処理内容を具体的な式で示せ。必要なら余りを求める演算には“ % ”を使うこと。条件式は左から順に評価を行い, 途中で値が確定した場合は, 残りの評価は行わないものとする。

解答群

- (a) k が括弧以外のコードか
- (b) k が開き括弧のコードか
- (c) k が閉じ括弧のコードか
- (d) スタックの一番上のデータが k に対応する開き括弧か
- (e) スタックの一番上のデータが k に対応する閉じ括弧か
- (f) スタックが空か
- (g) スタックが空でないか

問 6 作業状況を管理するシステムに関する次の記述を読んで、設問 1 ～ 4 に答えよ。

SI ベンダの A 社では、アプリケーション開発作業の進捗状況を管理するため、担当者が毎日提出する作業日報のデータを基に、作業実績集計表を作成している。このたび、プロジェクト管理者の作業負荷軽減と後続プロジェクトに対する実績データの提供を目的として、作業状況管理システムを開発することになった。

A 社では、既存の手書き帳票を参考にして作業日報入力画面（図 1）と、作業実績集計表（図 2）を設計し、担当者が入力する内容と、作業実績集計表の出力要件を次のように決定した。

〔担当者の入力内容〕

- ・担当者は、退社前に、その日の作業日報を作業日報入力画面（図 1）から入力する。このとき、当該作業が完了した場合は、作業完了サインに 1 を入力する。
- ・担当者が入力する項目は、アプリケーション設計書の作成やソースプログラムの編集作業・テスト作業などの直接作業と、会議や総務申請処理などの間接作業に分けて、作業コード単位に入力する。
- ・作業時間は、作業コードごとに 0.1 時間（＝6 分）刻みで入力する。

〔作業実績集計表の出力要件〕

- ・当システムは、担当者が入力する項目のうち、直接作業に関するデータだけを集計し、週 1 回の工程会議でプロジェクト管理者が参照するための作業実績集計表（図 2）を出力する。
- ・作業実績集計表は、作業コード別、担当者コード別に集計する。
- ・作業実績集計表の所要時間は、作業日報入力画面から入力された全作業時間について、作業コード、担当者コードごとに加算して求める。
- ・作業実績集計表の見積時間は、プロジェクト企画時に作業コード別、担当者コード別に作業時間を見積ったものである。

次に、このシステムで用いるデータベースを設計するために、作業状況管理に関する E-R 図（図 3）を作成した。



作業日報入力画面					
担当者コード：123456		担当者名：試験 宣太		作業年月日：2001/04/15	
項番	作業コード	作業名称	作業時間	作業完了サイン	備考
1	3359	PGM0020 内部設計書作成	2.6	1	
2	3360	PGM0020 ソース編集	4.5		
3	3361	PGM0020 テスト仕様書作成	2.5		
4	3461	PGM0030 テスト仕様書作成	1.0		
5	9010	進捗会議	1.0		
6					
日計			11.6		
残業時間 ( 3.6 )					

図1 作業日報入力画面

作業実績集計表						
項番	作業コード	作業名称	見積時間	担当者コード	担当者名	所要時間
1	3359	PGM0020 内部設計書作成	16.0	123456	試験 宣太	18.3
2	3360	PGM0020 ソース編集	16.0	123456	試験 宣太	11.5
3	3361	PGM0020 テスト仕様書作成	8.0	123456	試験 宣太	3.5
4	3459	PGM0030 内部設計書作成	18.0	789012	祖父斗 植亜	11.2
5	3461	PGM0030 テスト仕様書作成	4.0	123456	試験 宣太	1.0
6	3461	PGM0030 テスト仕様書作成	4.0	789012	祖父斗 植亜	1.5
⋮	⋮	⋮	⋮	⋮	⋮	⋮

図2 作業実績集計表

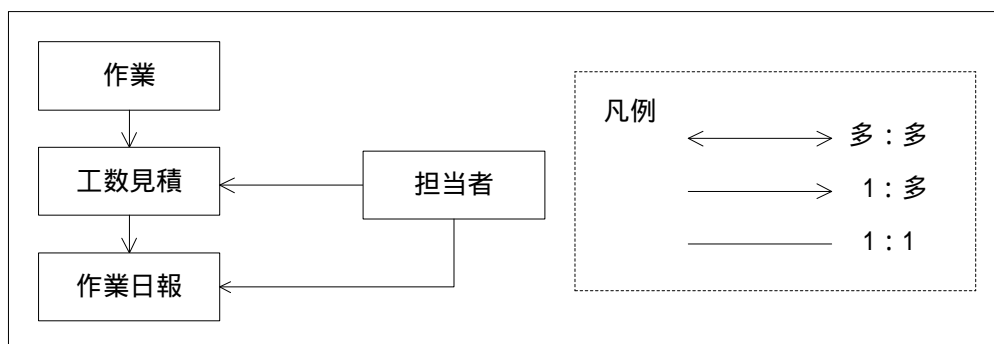


図3 作業状況管理に関するE-R図

設問 1 図 3 に示す E-R 図では、直接作業に関するデータと間接作業に関するデータを別々に集計することができないので、図 4 に示すように“作業区分表”を追加した。

作業区分表は、属性として次のような項目と値をもつ。

作業区分コード = 1 のとき、作業区分名称 = 直接作業  
作業区分コード = 0 のとき、作業区分名称 = 間接作業

(1) 凡例に従って、図 3 に作業区分を追加して E-R 図を完成させよ。

(2) 図 4 中の  ~  に入れる適切な字句を答えよ。識別キー項目（主キー項目）には下線を付けよ。

作業表 ( 作業コード, , 作業名称 )  
工数見積表 ( 担当者コード, 作業コード, 見積時間 )  
作業日報表 ( 担当者コード, 作業コード, 作業年月日, 作業時間, 作業完了サイン, 備考 )  
担当者表 ( 担当者コード, 担当者名 )  
作業区分表 ( ,  )

図 4 作業管理に関するテーブルと属性

設問 2 作業コード別、担当者コード別に作業時間を集計し、作業実績集計表を出力するためのユーザビューを作成する。次の SQL 文中の  ~  に入れる適切な字句を答えよ。

作業実績表  
( 作業コード, 作業名称, 見積時間, 担当者コード, 担当者名, 所要時間 )  
 D.作業コード, D.作業名称, C.見積時間, A.担当者コード,  
A.担当者名,   
FROM 担当者表 A, 作業日報表 B, 工数見積表 C, 作業表 D  
  
GROUP BY

設問 3 設問 2 で定義したユーザビューを使用して、作業コード別、担当者コード別の作業実績集計表を、作業コード順（昇順）、担当者コード順（昇順）に並べ替えて出力する。次の SQL 分中の  ~  に入れる適切な字句を答えよ。

なお、SQL 文法上の省略時解釈（デフォルト）は、使用しないものとする。

```
 *  
  
 作業コード  , 担当者コード 
```

設問 4 設問 2 で定義したユーザビューを使用して、所要時間が見積時間を超過している作業の一覧を作成するための SQL 文を作成した。このとき WHERE 句を次のようにコーティングしたところ、エラーになった。エラーになる理由を説明した次の記述中の  ~  に入れる適切な字句を答えよ。ただし、 には設問 2 の  と同じ字句が入る。

```
WHERE 所要時間 > 見積時間
```

エラーになる理由：

に対する参照処理は、 定義を用いて、 に対する参照処理に変換されて処理される。

したがって、

```
WHERE 所要時間 > 見積時間
```

は、

```
WHERE  > 見積時間
```

と、置き換えられることになる。

これは、WHERE 句としては正しくない。なぜなら、WHERE 句中の条件に  を指定することができないからである。