平成21年度　春期
基本情報技術者
午後　問題

試験時間	13:00 ～ 15:30（2時間30分）

注意事項

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

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

6. 答案用紙の記入に当たっては，次の指示に従ってください。
   (1) B又はHBの黒鉛筆又はシャープペンシルを使用してください。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，消しくずを残さないでください。
   (2) 答案用紙は光学式読取り装置で処理しますので，答案用紙の**マークの記入方法**のとおりマークしてください。
   (3) **受験番号欄**に，**受験番号**を記入及びマークしてください。正しくマークされていない場合，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。
   (4) **生年月日欄**に，受験票に印字されているとおりの**生年月日**を記入及びマークしてください。正しくマークされていない場合は，採点されないことがあります。
   (5) **選択した問題**については，右の例に従って，**選択欄**の**問題番号**の(選)をマークしてください。マークがない場合は，採点の対象になりません。問1～問7について，6問以上マークした場合は，はじめの5問を採点します。問9～問13について，2問以上マークした場合は，はじめの1問を採点します。

〔問1，問3，問4，問6，問7，問9を選択した場合の例〕

選択欄

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

   (6) 解答は，次の例題にならって，解答欄にマークしてください。

〔例題〕　次の［　　］に入れる正しい答えを，解答群の中から選べ。

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) ● (エ)

裏表紙の注意事項も，必ず読んでください。

〔問題一覧〕

●問 1～問 7（7 問中 5 問選択）

問題番号	出題分野	テーマ
問 1	ハードウェア	画像データの符号化
問 2	ソフトウェア	ソフトウェア製品の品質特性
問 3	データベース	関係データベースの設計と操作
問 4	情報セキュリティ	パケットフィルタリング
問 5	ソフトウェア設計	銀行口座の管理
問 6	プロジェクトマネジメント	スケジュール管理
問 7	経営・関連法規	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	図形の塗替え

●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）	絶対パスへの変換
問 10	ソフトウェア開発（COBOL）	売上分析表の印刷
問 11	ソフトウェア開発（Java）	簡易テキストエディタ
問 12	ソフトウェア開発（アセンブラ）	32 ビットの乗算
問 13	ソフトウェア開発（表計算）	生産計画の作成

共通に使用される擬似言語の記述形式

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

〔宣言，注釈及び処理〕

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

〔演算子と優先順位〕

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

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

〔論理型の定数〕

true，false

問題文中で共通に使用される表記ルール

E-R 図の表記ルールを次に示す。各問題文中に注記がない限り，この表記ルールが適用されているものとする。

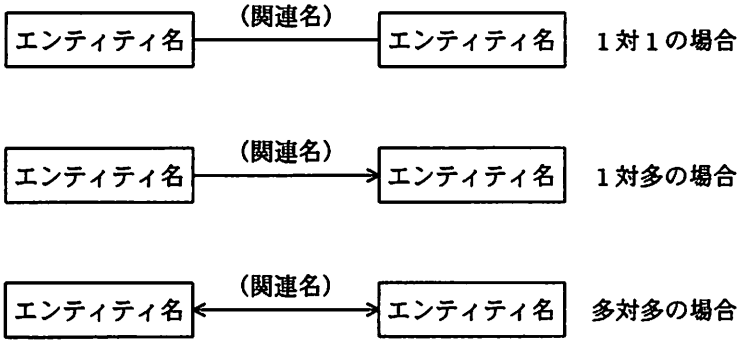


図　エンティティと関連の表記ルール

(1) エンティティを長方形で表す。

(2) 長方形の中にエンティティ名を記入する。

(3) エンティティ間の関連を直線又は矢印で表す。線のわきに関連名を“（関連名）”として記入する。

なお，関連名は省略することもある。

(4) “1対1”の関連は，直線で表す。

“1対多”の関連は，“多”側を指す片方向矢印とする。

“多対多”の関連は，両方向矢印とする。

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，答案用紙の選択欄の(選)をマークして解答してください。

なお，6問以上選択した場合には，はじめの5問について採点します。

問1　画像データの符号化に関する次の記述を読んで，設問1～3に答えよ。

図1は，8×8画素の白と黒だけで色分けされた2値画像の例である。画素を1番上の行の左から右へ，次に2番目の行の左から右へと順に1画素を1ビットで，白を0，黒を1で表すと，図2のように64ビットのビット列で表現することができる。

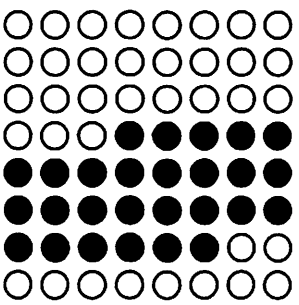


図1　2値画像の例

0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	1	1	1	1	1
1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1
1	1	1	1	1	1	0	0
0	0	0	0	0	0	0	0

図2　画像のビット列表現

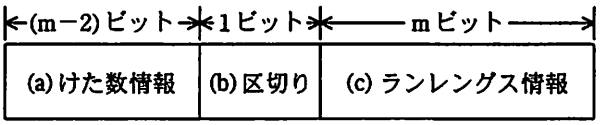
(1) 図2のビット列を，同じ値が連続している部分（以下，ランという）ごとに区切り，各ランをその連続する個数（以下，ランレングスという）で表すことによって，少ないビット数でのビット列表現に書き換えることができる。図2では，左上から数えて0が27個，1が27個，0が10個の順に連続しているので，27，27，10という情報を使った表現に書き換える。これを，ランレングス符号化という。

(2) 1番上の行の左端の画素は白で始まるものとする。ただし，その画素が黒の場合は，先頭に0個の白があるものとして符号化を行う。

(3) ランレングス符号化の方法は，次のとおりである。

① ランレングスをnとし，nを2進数で表現したときのけた数をmとする。ただし，常にm≧2となるように，n＝0の2進数表現を00，n＝1の2進数表現を01とする。

② nビットのランを図3のビット列に書き換える。



(a) nを2進数で表現したときのけた数がmのとき，m−2個の連続する1とする。m＝2のときはこの部分のビット数は0であり，次の(b)から始まる。

(b) 1個の0とする。

(c) nの2進数表現とする。

図3 符号化後のビット列表現

(4) 図2の例では，最初は0が27個連続しているので，n＝27である。27を2進数で表現すると11011（5けた）となり，m＝5である。m−2＝3なので，けた数情報は111である。したがって，この部分の符号化後のビット列表現は111011011となり，9ビットで表現できる。

(5) 表にn，m及び符号化後のビット列を示す。

表 n，m及び符号化後のビット列

n	m	符号化後のビット列
0	2	000
1	2	001
2	2	010
3	2	011
4	3	a
⋮	⋮	⋮
b	4	1101111
⋮	⋮	⋮
27	5	111011011
⋮	⋮	⋮

設問1　表中の［　　］に入れる正しい答えを，解答群の中から選べ。

aに関する解答群

ア　100　　イ　0100　　ウ　10100　　エ　110100

bに関する解答群

ア　14　　イ　15　　ウ　16　　エ　17

設問2　図2の64ビットのビット列をランレングス符号化すると，何ビットで表現できるか。正しい答えを，解答群の中から選べ。

解答群

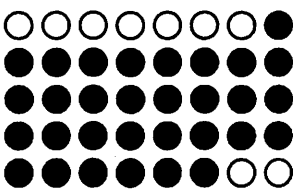
ア　22　　イ　23　　ウ　24　　エ　25

設問3　ランレングス符号化後のビット列が，次のとおりであったとする。このビット列を復号した2値画像として正しい答えを，解答群の中から選べ。

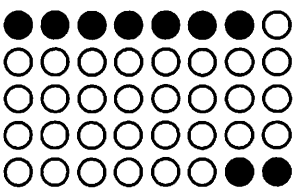
000111011111111011111010

解答群

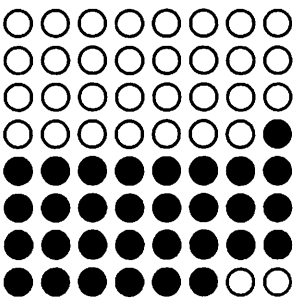
ア



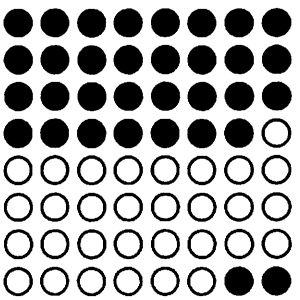
イ



ウ



エ



問2　ソフトウェア製品の品質特性に関する次の記述を読んで，設問に答えよ。

JIS X 0129-1では，ソフトウェア製品の品質について，表1に示す六つの品質特性を定めている。

表1　六つの品質特性（JIS X 0129-1）

品質特性	ソフトウェア製品の能力の概要	品質副特性（一部）
機能性	指定された条件下で利用されるとき，明示的及び暗示的必要性に合致する機能を提供する。	合目的性，正確性，セキュリティ，相互運用性
使用性	指定された条件下で利用するとき，理解，習得，利用でき，利用者にとって魅力的である。	運用性，習得性，魅力性，理解性
信頼性	指定された条件下で利用するとき，指定された達成水準を維持する。	回復性，障害許容性，成熟性
効率性	明示的な条件下で，使用する資源の量に対比して適切な性能を提供する。	時間効率性，資源効率性
保守性	修正のしやすさ	安定性，解析性，試験性，変更性
移植性	ある環境から他の環境に移すことができる。	環境適応性，共存性，設置性，置換性

これらの品質特性のうち，コーディングの段階では，信頼性，効率性，保守性，移植性を考慮することが大切である。

あるソフトウェア開発会社では，開発するソフトウェア製品の品質向上を図るため，品質特性を考慮したプログラム開発の社内標準を制定し，作成したプログラムのコードレビュー体制を確立した。

表2は，最近のコードレビューで新人のプログラム開発担当者が受けた指摘の例である。

表2　新人のプログラム開発担当者が受けた指摘の例

ソースコード	指摘の内容	主な品質特性 (品質副特性)
○実数型: Ave, Count, Total ⋮ ・Ave ← Total ÷ Count ⋮	左の処理を次のように変更すること。 ⋮ Count ＞ 0 ・Ave ← Total ÷ Count ・Ave ← 0 ⋮	a
⋮ r: 1, r ≦ rMax, 1 c: 1, c ≦ cMax, 1 X[r, c] ＞ Sub(r) ＋ 1 ・X[r, c] ← Sub(r) ＋ 1 ⋮	左の処理で，関数 Sub は計算時間は長いが，返却値は引数だけに依存する。 次のように最適化すること。 ⋮ b ⋮	効率性 (時間効率性)
⋮ /* 主記憶の動的取得 */ ・GetMain(Addr, Len) ⋮ /* 主記憶の動的開放 */ ・FreeMain(Addr) ⋮	主記憶の動的取得と開放で，システム標準の関数を使用している。一般に，取得した範囲外や開放済の記憶域を誤って更新するなどの障害は，c 。 次のように，デバッグ機能のある社内で開発した同機能の関数を使用すること。 ⋮ ・X_GetMain(Addr, Len, …) ⋮ ・X_FreeMain(Addr, …) ⋮	保守性 (試験性)
○整数型: P1, P2, Ans ○整数型関数: Fn(P1, P2) ⋮ ・Ans ← Fn(P1, P2) ⋮	このプログラムは複数の機種で汎用的に使われる。機種の違いによって d が異なることがあるので，次のように宣言の記述形式を変更すること。 ○32ビット整数型: P1, P2, Ans ○32ビット整数型関数: Fn(P1, P2) ⋮	e

設問　表2中の［　　］に入れる正しい答えを，解答群の中から選べ。

a，eに関する解答群

ア　移植性（環境適応性）　　イ　効率性（資源効率性）　　ウ　信頼性（成熟性）

エ　保守性（解析性）　　オ　保守性（変更性）

bに関する解答群

ア

```
r: 1, r ≦ rMax, 1
  c: 1, c ≦ cMax, 1
    ・S1 ← Sub(r) + 1
    X[r, c] > S1
      ・X[r, c] ← S1
```

イ

```
r: 1, r ≦ rMax, 1
  ・S1 ← Sub(r) + 1
  c: 1, c ≦ cMax, 1
    X[r, c] > S1
      ・X[r, c] ← S1
```

ウ

```
r: 1, r ≦ rMax, 1
  c: 1, c ≦ cMax, 1
    ・S1 ← Sub(r) + 1
    ・Xrc ← X[r, c]
    Xrc > S1
      ・Xrc ← S1
```

エ

```
r: 1, r ≦ rMax, 1
  ・S1 ← Sub(r) + 1
  c: 1, c ≦ cMax, 1
    ・Xrc ← X[r, c]
    Xrc > S1
      ・Xrc ← S1
```

cに関する解答群

ア　更新した時点で障害と分かるが，ログを記録する機能のあるOSは少ない

イ　更新した時点で障害と分かるが，ログを記録する機能のあるハードウェアは少ない

ウ　更新内容を後で参照したときに障害となることが多く，原因箇所の特定が困難である

エ　取得可能な主記憶域が残っている間は，障害を検知できない

dに関する解答群

ア　指定できる変数や関数の個数

イ　変数や関数の型宣言で省略した場合のビット数

ウ　リンカで扱える関数のビット数

エ　ローダで扱える関数の個数

問3　関係データベースの設計と操作に関する次の記述を読んで，設問1～4に答えよ。

あるスーパーマーケットでは，野菜，魚，肉の各売場に，来店客が食材名を入力するとその食材を使った料理名と4人分のレシピを表示し，印刷して持ち帰ることもできる端末を設置することにした。このシステムは，料理名，使用する食材，レシピなどを関係データベースで管理し，要件に応じて適切な情報を抽出する。

このシステムで使用する表は図1のとおりである。網掛けした項目は主キーを表す。

料理表

料理ID	料理名	作り方
0001	ビーフカレー	大きめに切った材料を，油を引いたなべで…
0002	ロールキャベツ	みじん切りにした玉ねぎをよくいため，…
⋮	⋮	⋮

商品表

商品ID	商品名	種別
000001	にんじん	野菜
000002	玉ねぎ	野菜
000003	じゃがいも	野菜
⋮	⋮	⋮
001001	牛肉	肉
001002	羊肉	肉
⋮	⋮	⋮

分量表

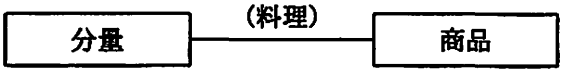
料理ID	商品ID	分量
0001	000001	中1本
0001	000002	中3個
0001	001001	200グラム
⋮	⋮	⋮

図1　表の構成

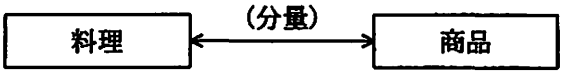
設問1　データベースを設計するに当たって，エンティティの関係を表すE-R図を作成した。図1の構成におけるE-R図を，解答群の中から選べ。

解答群

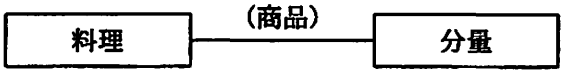
ア



イ



ウ



設問2　図1の表は，料理ごとに作り方，使用する商品（食材）とその分量を記入して作成したレシピの表を正規化して得られたものである。データベース設計における第1正規化に基づいて実施した処置を，解答群の中から選べ。

解答群

ア　実行性能の向上を図り，料理表と分量表に分けた。

イ　主キーを一意にするため，料理名に料理IDを割り振った。

ウ　商品とその分量という繰返しの要素を排除した。

エ　情報の独立性を高めるため，商品名に商品IDを割り振った。

設問3　野菜売場の端末で来店客が好みの野菜を一つ選択したときに，その野菜を使用する料理名をすべて抽出して表示したい。来店客が“じゃがいも”を選択した場合，次のSQL文の［　　］に入れる正しい答えを，解答群の中から選べ。

```
SELECT 料理表.料理名 FROM 料理表, 分量表, 商品表
   WHERE [                                    ]
```

解答群

ア
```
料理表.料理ID = (SELECT 分量表.料理ID FROM 分量表, 商品表
                    WHERE 商品表.商品名 = 'じゃがいも')
```

イ
```
料理表.料理ID = 分量表.料理ID AND 分量表.商品ID = 商品表.商品ID
                    AND 商品表.商品名 = 'じゃがいも'
```

ウ
```
料理表.料理ID = 分量表.料理ID AND 分量表.商品ID = 商品表.商品ID
                    OR 商品表.商品名 = 'じゃがいも'
```

エ
```
料理表.料理ID IN (SELECT 分量表.料理ID FROM 分量表, 商品表
                    WHERE 商品表.商品名 = 'じゃがいも')
```

設問4　設問3のSQL文では，当日に在庫がない商品を材料に使用する料理名まで表示してしまう。このため，図2に示すとおり，開店時に在庫がある場合は1，ない場合は0を設定する項目“在庫”を追加し，使用する食材すべてについて在庫がある料理名だけを表示できるように商品表を変更する。また，同時にSQL文の可読性を向上させるため，表の結合にはJOIN句を用いる。次のSQL文の［　　］に入れる正しい答えを，解答群の中から選べ。

商品表

商品ID	商品名	種別	在庫
000001	にんじん	野菜	1
000002	玉ねぎ	野菜	1
000003	じゃがいも	野菜	1
⋮	⋮	⋮	⋮
001001	牛肉	肉	1
001002	羊肉	肉	0
⋮	⋮	⋮	⋮

図2　変更した商品表

```
SELECT 料理表.料理名 FROM 料理表
    JOIN 分量表 ON 料理表.料理ID = 分量表.料理ID
    JOIN 商品表 ON 分量表.商品ID = 商品表.商品ID
    WHERE [                                    ]
```

解答群

ア　商品表.在庫 != 0 AND 商品表.商品名 = 'じゃがいも'

イ　商品表.在庫 != 0 OR 商品表.商品名 = 'じゃがいも'

ウ
```
料理表.料理ID IN
    (SELECT 分量表.料理ID FROM 分量表
        JOIN 商品表 ON 分量表.商品ID = 商品表.商品ID
        WHERE 商品表.在庫 = 0)
    AND 商品表.商品名 = 'じゃがいも'
```

エ
```
料理表.料理ID NOT IN
    (SELECT 分量表.料理ID FROM 分量表
        JOIN 商品表 ON 分量表.商品ID = 商品表.商品ID
        WHERE 商品表.在庫 = 0)
    AND 商品表.商品名 = 'じゃがいも'
```

問4　パケットフィルタリングに関する次の記述を読んで，設問1，2に答えよ。

X社では，図に示すネットワークを構築し，インターネットへのWebサイトの公開と電子メール（以下，メールという）の送受信を行っている。

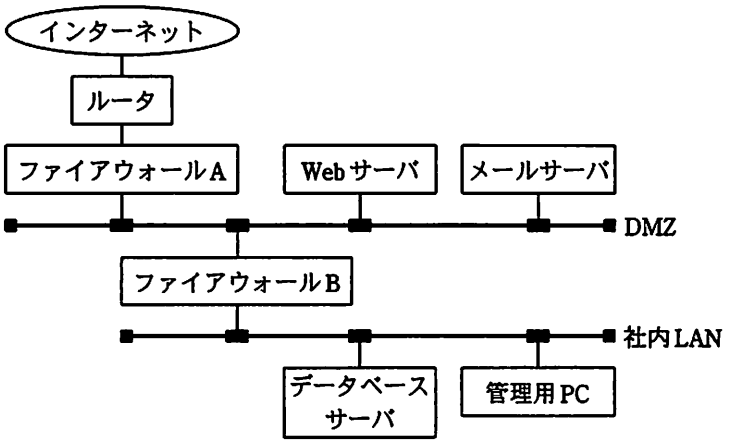


図　X社のネットワーク構成

X社のネットワークは二つのファイアウォールによって，DMZ及び社内LANの二つのセグメントに分けられている。Webサーバ，メールサーバ及びデータベースサーバ（以下，DBサーバという）は，それぞれ次の役割を果たしている。

(1) Webサーバ

Webサイトとして，自社の情報をインターネットに公開する。Webサーバ上では，社外との取引情報を処理するプログラムが動作する。このプログラムが利用するデータはDBサーバ上に格納される。

(2) メールサーバ

社外とのメールの送受信を行う。また，取引先に対してメールを自動配信するプログラムが動作する。メール配信のためのデータはDBサーバ上に格納される。

(3) DBサーバ

Webサーバ及びメールサーバで利用するデータを格納する。

社内LANに接続された管理用PCからは，SSHを使った各サーバへのログイン操作と，メールサーバを介した外部とのメール送受信が可能である。管理用PCから自社Webサーバの参照はできるが，社外Webサイトの利用は許可されていない。

ネットワーク上で使われるプロトコルとポート番号を表1に示す。

表1　プロトコルとポート番号

サービス	プロトコル	ポート番号
Web	HTTP	80
メール転送	SMTP	25
セキュアシェル（遠隔ログイン）	SSH	22
メール受信	POP3	110
DBアクセス	DB専用	1999

設問1　次の記述中の［　　］に入れる正しい答えを，解答群の中から選べ。解答は重複して選んでもよい。

インターネットとDMZをつなぐファイアウォールAのパケットフィルタリングの設定を表2に示す。また，DMZと社内LANをつなぐファイアウォールBのパケットフィルタリングの設定を表3に示す。

フィルタリングの設定ルールは，送信元のIPアドレス，あて先のIPアドレス及び接続先ポート番号を指定して通信の許可／拒否を制御する。設定は上の行のルールから調べて，最初に条件が合致した行の動作を実行する。また，応答パケットについては動的フィルタリング機能によって自動的に許可されるので設定は不要なものとする。

表2　ファイアウォールAのフィルタリングの設定

条件			動作
送信元	あて先	ポート番号	
任意	Webサーバ	80	許可
任意	メールサーバ	25	許可
a	任意	b	許可
任意	任意	任意	拒否

表3　ファイアウォールBのフィルタリングの設定

条件			動作
送信元	あて先	ポート番号	
Webサーバ	DBサーバ	1999	許可
メールサーバ	DBサーバ	1999	許可
管理用PC	c	d	許可
管理用PC	メールサーバ	22	許可
管理用PC	メールサーバ	25	許可
管理用PC	Webサーバ	80	許可
管理用PC	Webサーバ	22	許可
任意	任意	任意	拒否

a，cに関する解答群

ア　DBサーバ　　イ　Webサーバ　　ウ　管理用PC

エ　メールサーバ　　オ　任意

b，dに関する解答群

ア　22　　イ　25　　ウ　80　　エ　110　　オ　1999

設問2　X社のネットワークでは，ファイアウォールによるパケットフィルタリングによって，インターネット接続に伴うセキュリティ上のリスクを低減しているが，パケットフィルタリングは，すべての脅威に対する防御とはならない。パケットフィルタリングによって防ぐことができるセキュリティ上のリスクとして，正しい答えを解答群の中から二つ選べ。

解答群

ア　Webサイトとやり取りされるデータの盗聴や改ざん

イ　WebサイトへのSQLインジェクション攻撃

ウ　インターネットからDMZ内のサーバへの許可されていないポートでの接続

エ　インターネットから社内LANへの不正アクセスによる攻撃

オ　メールによる社内からのファイル流出

問5　銀行口座の管理に関する次の記述を読んで，設問に答えよ。

ある銀行の個人顧客口座の口座管理手数料，現金自動預払機（以下，ATM という）での預払い及びATM 使用手数料に関する口座情報を更新するプログラムである。

口座は，口座番号で識別される。口座には，所定の口座管理手数料が毎月掛かり，その月の月末処理で残高から引き落とされる。この口座管理手数料は，その口座の前月末日時点の残高が10万円以上であれば，当月分は掛からない。

口座に対してのATM からの預入れ及び引出しの操作には，1回当たり所定のATM 使用手数料が掛かり，操作時に，その口座の残高から自動的に引き落とされる。このATM 使用手数料も，その口座の前月末日時点の残高が10万円以上であれば，当月分は掛からない。

なお，同一口座に対しては，預入れ及び引出しの操作が完了するまで，ほかの操作は行えないものとする。

データベースには，所定の口座管理手数料及び所定のATM 使用手数料と，口座ごとの残高及び手数料マークが格納されている。

手数料マークは，前月末日時点の残高が10万円未満であった場合に“真”，そうでない場合に“偽”となる。

プログラムを構成する各モジュール間の関連を図に示す。

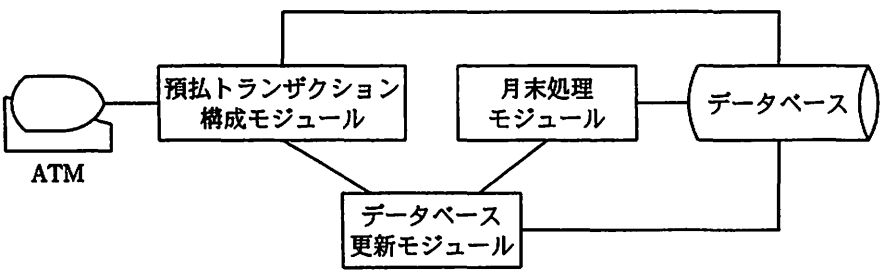


図　プログラムを構成する各モジュール間の関連

〔預払トランザクション構成モジュールの説明〕

このモジュールには，次の各処理が用意されており，口座ごとに，預払い１件を単位として一つのトランザクションを構成する。

処理Ａ：ATM での預払操作に対応して，データベースを更新する命令を作成し，データベース更新モジュールに送る。

処理Ｂ：残高不足となる場合は，利用者にメッセージで伝え，預払処理を行わない。

処理Ｃ：データベースを更新する命令１件に対して，所定の ATM 使用手数料を口座から引き落とす命令を作成し，データベース更新モジュールに送る。

〔データベース更新モジュールの説明〕

到着した順番にトランザクションを処理し，データベース上の対応する口座の情報を更新する。

〔月末処理モジュールの説明〕

このモジュールには，次の各処理が用意されており，月末処理のため，口座ごとに一つのトランザクションを構成し，必要な処理を行う。月末処理は，月末日の ATM からの操作受付終了後に行う。

処理Ｄ：翌月のために手数料マークを設定する命令を作成し，データベース更新モジュールに送る。

処理Ｅ：所定の口座管理手数料を残高から引き落とした新たな残高を計算する。新たな残高を書き戻す命令を作成して，データベース更新モジュールに送る。計算結果の残高が負の値になることもある。

設問　次の各モジュールに関する記述中の［　　］に入れる正しい答えを，解答群の中から選べ。

預払トランザクション構成モジュールでは，それぞれの実行条件に従って処理A～C及びデータベースへのアクセスが実行される。

その順番は，次のとおりである。

［ a ］→［ b ］→［ c ］→処理C

月末処理モジュールの処理Dでの手数料マークの設定に最低限必要なデータは，［ d ］である。また，処理Eは実行される場合とされない場合がある。処理Eの実行判定に必要なデータと，処理Eの実行に最低限必要なデータは，［ e ］である。

ここで，月末処理モジュールの処理D，処理Eの両方が実行される場合，その順番は，［ f ］である。

a～cに関する解答群

ア　処理A　　イ　処理B

ウ　データベースの検索　　エ　データベースの項目内容変更

オ　データベースのレコード削除

d，eに関する解答群

ア　所定の口座管理手数料

イ　所定の口座管理手数料，その口座の残高

ウ　所定の口座管理手数料，その口座の残高，その口座の手数料マークの値

エ　その口座の残高

オ　その口座の手数料マークの値

fに関する解答群

ア　処理Dの次に処理E　　イ　処理Eの次に処理D　　ウ　任意

問6　スケジュール管理に関する次の記述を読んで，設問1～3に答えよ。

A社は，2年前に基幹業務システムをB社に委託して再構築し，業務効率の向上に成功した。経営改革を更に進めるため，蓄積された経営データを活用した経営管理システムを開発することになり，情報システム部内で第Ⅰ期プロジェクトを立ち上げた。

〔第Ⅰ期プロジェクトの概要〕

経営管理システムは，経営データ分析サブシステムM1と経営幹部向けナビゲーションサブシステムM2とで構成される。M1はデータ集計用プログラムと分析プログラムとから成る。M2は，M1による分析結果を経営幹部向けに編集し，イントラネットを通してブラウザで閲覧できるようにする。これらのサブシステムの要件定義は，A社の情報システム部が行い，設計からテストまでの開発作業は，経営管理システムの開発経験も豊富であるB社に委託した。B社は，A社からシステムの概略機能について説明を受けた後に概算見積りを行い，開発期間を3か月として請負契約を結んだ。

なお，開発はすべてB社内で行うことにした。

〔システム開発の遂行状況〕

情報システム部による要件定義は予定どおりの日程で完了し，要件定義書がB社に渡された。B社は，自社内での開発作業を開始し，開発期間中は両社合同のプロジェクト進捗会議を毎週1回行った。

開発開始から2か月が経過した時点の進捗会議で，B社から“M1の開発作業に遅れが出ているが，開発メンバを増やして納期に間に合わせる”という報告があった。しかし，2週間後にB社から“開発作業の遅れを取り戻せないので，納期をとりあえず2週間延ばしてほしい”との申出があり，A社は仕方なく了承した。その後もスケジュールの遅れは続き，当初の予定から1か月遅れて開発が完了した。B社からは，“M1が予想以上に複雑であり，更にM2の操作性を自社判断で一層高めたことによって，開発規模が当初見積りの1.5倍になってしまったことが遅延理由である”との報告があった。

A社側での受入検査の結果は，品質を含めて良好で，特に，M2は要件定義時の仕様よりも使い勝手がとても良く，経営幹部の評価も高かった。

設問1　B社は，今回のシステム開発での遅延理由を二つ挙げている。契約時の見積値とは違ってしまった原因の説明として，表1中の［　　］に入れる最も適切な答えを，解答群の中から選べ。

表1　B社の回答と原因の説明

	〔システム開発の遂行状況〕中のB社の回答	原因の説明
1	M1が予想以上に複雑であったこと	a
2	M2の操作性を一層高めたこと	b

解答群

ア　開発担当者の技術力が計画時の予測よりも低かったので，実際の開発期間が契約時の見積値を超えてしまった。

イ　設計作業の途中に，A社との調整を行わないで勝手に機能を広げてしまったので，開発規模が増大し，実際の開発期間が契約時の見積値を超えてしまった。

ウ　要件定義書を受け取った時点で契約時の概算見積りを見直さなかったので，契約時の概算見積りによる計画のまま開発が進められて，実際の開発期間が契約時の見積値を超えてしまった。

エ　要件定義をすべてA社側で行った上に，開発作業をすべてB社内で行ったので，A社の業務の理解に計画よりも多くの時間を要した。また，仕様の解釈に誤解が生じて設計作業の手戻りも発生し，実際の開発期間が契約時の見積値を超えてしまった。

設問2　両社合同によるプロジェクト進捗会議を毎週実施したにもかかわらず，今回のような開発の遅れが生じたことを互いに反省し，進捗会議でのリスク管理について対策会議を行った。この結果，次のような改善策の実施を決めた。［　　］に入れる最も適切な答えを，解答群の中から選べ。

これまでの進捗会議では，主にサブシステム単位での作業状況を，B社から報告していた。開発上で何らかの問題が発生した場合又はそのおそれがある場合には，問題点の内容，プロジェクトへの影響度及び対策案について両社で相談してきた。

今後は，より定量的なデータによって開発状況の実態を把握し，プロジェクトに悪影響を与えることがないよう未然に防止する。もし発生した場合でも，その

影響を最小限に抑えられるようにする。

定量的なデータとしては，各開発工程での作業成果物の生産データ（プログラム規模など），品質データ（レビュー結果，テスト結果など）及び進捗データ（プログラムの完成度，プロジェクト進捗度，計画と実績の差異分析）を使用する。これらのデータを分析することによって，例えば，今回の開発での［ c ］といった現象を回避する。

また，開発途中での仕様，スケジュール，開発体制に関する変更管理（変更提案，変更に対する審議・承認など）を進捗会議の場で行う。これによって，例えば，今回の開発での［ d ］といった現象を回避する。

解答群

ア　開発メンバを増強する

イ　概算見積りを行った時点よりも要求機能が複雑であったM1の納期遅延

ウ　操作性を一層高めたことによるM2の納期遅延

エ　納期を2週間延ばした後の更なる遅れの発生

〔第Ⅱ期プロジェクトの概要〕

経営管理システムの完成から1年後，A社の経営幹部から新たな経営データ分析機能の要求があり，情報システム部に第Ⅱ期プロジェクトを立ち上げた。このプロジェクトでは，経営データ分析サブシステムM3の新規開発とナビゲーションサブシステムM2の改造を行う。システム開発はB社が再び請け負い，開発作業は前回の開発担当者2名で行う。B社では，前回の反省から，開発でのスケジュール見積精度の向上を図るために，3点見積法を使用したスケジュールリスク分析を行うことにした。

3点見積法とは，仕事の作業期間（ここでは日数）を，最頻値，悲観値（悲観的に最も長い期間を見積もる），及び楽観値（楽観的に最も短い期間を見積もる）の3種の値を用いて推定する方法である。3点見積法による作業期間の平均，分散，標準偏差，開発全体の標準偏差の計算式は，次のように定義されている。

$$\text{平均}=\frac{\text{悲観値}+4\times\text{最頻値}+\text{楽観値}}{6} \qquad \text{分散}=\left(\frac{\text{悲観値}-\text{楽観値}}{6}\right)^2$$

$$\text{標準偏差}=\sqrt{\text{分散}} \qquad \text{開発全体の標準偏差}=\sqrt{\text{分散の合計}}$$

B社が行った3点見積法によるスケジュールリスク分析を表2に示す。

表2　3点見積法によるスケジュールリスク分析

作業名	作業日数			平均作業日数	分散	標準偏差
	悲観値	最頻値	楽観値			
M3の新規開発	20	12	8	12.7	4	2
M2の改造	16	12	10	12.3	1	1
開発全体					5	2.2

注　網掛けの部分は表示していない。

なお，プロジェクトの作業日数の確率分布は，正規分布（平均μ，分散σ^2）に近似できると仮定する。作業日数が$\mu \pm 1\sigma$の範囲に収まる確率は0.68であり，$\mu \pm 2\sigma$の範囲に収まる確率は0.95である。

設問3　3点見積法によるスケジュール見積り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。

M2及びM3の開発作業全体の平均作業日数は，［ e ］日になる。また，確率0.95で作業が完了する日数を，スケジュールリスクを考慮して見積もったとき，その最長作業日数は，平均作業日数に［ f ］日を加えた値である。

eに関する解答群

ア　24　　イ　25　　ウ　27　　エ　29

fに関する解答群

ア　2　　イ　2.2　　ウ　4　　エ　4.4

問7　需要予測に関する次の記述を読んで，設問1～3に答えよ。

精密機器を製造，販売しているC社では，製品在庫の削減と，欠品に伴う販売機会損失の低減を目的として，製品の需要予測システムを構築することになった。

担当となったD君は，C社製品の需要の特性及び需要予測システムへの要望を，社内の関係部署からヒアリングすることにした。

〔ヒアリングの結果〕

D君が実施したヒアリングの結果は，次のとおりである。

(1) 製造工程の事情によって，製品の生産台数は月単位で計画し，需要の増減に対応しているので，需要数量の予測も月単位で提供されることが望まれる。

(2) 月単位でみた需要の増減のパターンが，毎年繰り返される製品がある。

(3) 製品の世代交代などの影響で，需要が徐々に増えていく製品と，徐々に減っていく製品とがある。

(4) 上記の(2)，(3)以外の製品は，小幅な需要の増減が不規則に発生している。

(5) 製品のモデルチェンジのサイクルは5年から10年の間であり，新製品を除いて時系列分析に必要な期間の需要実績データは保管されている。

設問1　需要予測システムに関する次の記述中の［　　］に入れる最も適切な答えを，解答群の中から選べ。

D君は実施したヒアリングの結果から，C社製品の需要予測には，過去の需要推移から将来を予測する手法である時系列分析が適用できると考えた。需要予測のために必要と考えたシステム機能は，次のとおりである。

① 製品ごとの需要実績データに含まれる，需要の［ a ］傾向変動を把握して，予測値に反映させる機能

② 製品ごとの需要実績データに含まれる，需要の［ b ］季節変動を把握して，予測値に反映させる機能

③ 製品ごとの需要実績データに含まれる，［ c ］である不規則変動の影響を取り除き，予測値を平準化させる機能

a，bに関する解答群

ア　増加と減少のパターンが１年ごとに繰り返される

イ　増加と減少のパターンが５か月ごとに繰り返される

ウ　増加と減少のパターンが不定期に繰り返される

エ　増加又は減少が間欠的に発生する

オ　長期的な増加又は減少が継続する

cに関する解答群

ア　傾向変動や季節変動では説明できない部分

イ　人為的な影響による需要の増減部分

ウ　定常的に発生する需要の部分

エ　天候や気象の影響による需要の増減部分

設問２　Ｄ君は，システム機能が要望を満たしているか確認するため，図に示すＣ社製品の過去３年間の需要実績を基に，製品ごとに適用する予測手法の検討を行った。製品Ｘ，製品Ｙ，製品Ｚの需要実績に適用すべき予測手法として適切な組合せを，解答群の中から選べ。

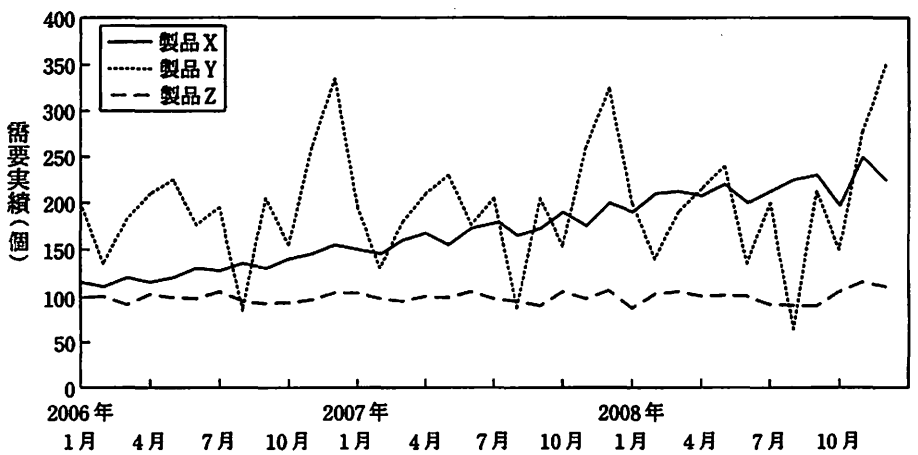


図　Ｃ社製品の過去３年間の需要実績

解答群

	製品X	製品Y	製品Z
ア	季節変動と不規則変動	傾向変動と不規則変動	不規則変動だけ
イ	季節変動と不規則変動	不規則変動だけ	傾向変動と不規則変動
ウ	傾向変動と不規則変動	季節変動と不規則変動	不規則変動だけ
エ	傾向変動と不規則変動	不規則変動だけ	季節変動と不規則変動
オ	不規則変動だけ	季節変動と不規則変動	傾向変動と不規則変動

設問3 D君は，製品Zの需要変動に対し，移動平均法と指数平滑法の二つの予測手法の適合度を，実際のデータを使って調べることにした。その結果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。

表　製品Zの過去7か月の需要実績

年	2008年						
月	6月	7月	8月	9月	10月	11月	12月
需要実績	100	95	93	92	105	115	110

〔予測手法の製品Zへの適用の流れと結果〕

① D君は，2008年11月までの実績データを用いて2008年12月の予測値を求め，実際の2008年12月の実績値との適合性をチェックしてみた。

② 次の式で計算される移動平均法を，過去3か月の製品Zの需要実績について適用してみた。その結果，移動平均法（過去3か月の需要実績を使用する場合）による2008年12月の予測値は，［ d ］になった。

・2008年12月の予測値（移動平均法）

＝(2008年9月から11月までの各月の需要実績の総和)÷3

③ 次の式で計算できる指数平滑法を用いて，指数aを0.6として，製品Zの2008年12月の予測値を求めると，［ e ］となった。

・2008年12月の予測値（指数平滑法）

＝a×(2008年11月の需要実績)＋(1－a)×(2008年11月の予測値)

なお，指数平滑法で求めた2008年11月の予測値は100であった。

d，eに関する解答群

ア 98	イ 101	ウ 104
エ 106	オ 109	カ 110

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，設問に答えよ。

〔プログラムの説明〕

64（8×8）画素からなる表示領域がある。この表示領域中の一つの画素を指定して，その画素を含む同じ色の領域を，指定した色で塗り替えるプログラムである。

ある一つの画素について，その画素の上下左右の4方向に隣接した画素の中に同じ色のものがあれば，それらの画素は同じ色の領域内にあるものと判定する。このようにして領域内にあると判定された隣接する各画素について，同様の判定を繰り返し，指定した色で塗り替えていく。

(1) 大きさ10×10の2次元配列Image（各添字の範囲は0～9）を用意する。各画素の色は，配列Imageの一部（各添字の範囲は1～8）に保持する。

(2) 色は，黒（■），灰（▒），白（□）の3色で，それぞれを値1，2，3で表す。

(3) 塗り替えたい領域中の開始点を示す一つの画素を，変数VSとHSで指定する。VSとHSは，その画素に対応する配列Imageの要素の縦と横方向の添字である。

(4) 塗り替えたい色を，変数NCで指定する。

(5) プログラムは，Image[VS，HS]から現在の画素の色を取得し，その画素を含む同じ色の領域を，色NCで塗り替える。

(6) 大域変数Image，NC，VS，HSには，正しい値が設定されているものとする。

(7) 配列VPos，HPosの添字は，1から始まる。

(8) 図1は，VS＝5，HS＝3として，Image[5，3]を塗り替えたい領域内の開始点に指定し，NC＝1として，塗り替える色を黒（■）に指定した場合の，プログラムの実行例である。

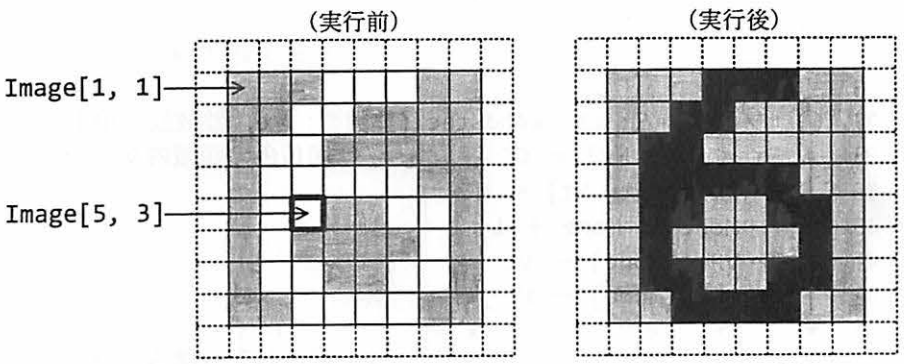


図1　プログラムの実行例（NC＝1，VS＝5，HS＝3の場合）

〔プログラム〕

```
(行番号)
 1  ○符号なし8ビット整数型: Image[10, 10]      /* 画素の色情報 */
 2  ○整数型: VS, HS                          /* 開始点はImage[VS,HS] */
 3  ○符号なし8ビット整数型: CC, NC           /* 色CCの領域を色NCで塗替え */
 4  ○整数型: More                            /* 処理待ちの画素の数 */
 5  ○整数型: VPos[64], HPos[64]              /* 処理待ちの画素の位置情報 */
 6
 7  ○プログラム: Main
 8  ○整数型: V, H                        /* 縦(V)と横(H)方向の添字用 */
 9  ○符号なし8ビット整数型: Wall         /* 表示領域の外周に格納する値 */
10
11  ・CC ← Image[VS, HS]                 /* 開始点の現在の色を取得 */
12  ▲ CC = NC
13  |   ・Return                         /* 処理を終了 */
14  ▼
15  ・ Wall ← 0
16  ■ V: 1, V ≦ 8, 1                     /* Vを1～8として外周に値を設定 */
17  |   ・Image[V, 0] ← Wall
18  |   ・Image[V, 9] ← Wall
19  ■
20  ■ H: 1, H ≦ 8, 1                     /* Hを1～8として外周に値を設定 */
21  |   ・Image[0, H] ← Wall
22  |   ・Image[9, H] ← Wall
23  ■
24  ・More ← 0
25  ・CheckAndStack(VS, HS)   /* 開始点を処理待ちの画素として登録 */
26  ■ More > 0                /* More>0の間，次の処理を繰り返す。 */
27  |   ・V ← VPos[More]
28  |   ・H ← HPos[More]
29  |   ・More ← More − 1
30  |   ・CheckAndStack(V − 1, H)
31  |   ・CheckAndStack(V, H − 1)
32  |   ・CheckAndStack(V + 1, H)
33  |   ・CheckAndStack(V, H + 1)
34  ■
35  ・Return                             /* 処理を終了 */
36
37  ○副プログラム: CheckAndStack(整数型: Vt, 整数型: Ht)
38  ▲ Image[Vt, Ht] = CC                 /* 同じ色の領域内か？ */
39  |   ・Image[Vt, Ht] ← NC
40  |   ・More ← More + 1
41  |   ・VPos[More] ← Vt
42  |   ・HPos[More] ← Ht
43  ▼
44  ・Return                             /* 呼出し元へ戻る。 */
```

設問　次の記述中の［　　］に入れる正しい答えを，解答群の中から選べ。

このプログラムに関する先輩技術者（以下，先輩という）と新人技術者（以下，新人という）の会話である。

先輩：プログラムの動作を理解するには，処理の流れを追跡してみることが大切です。まず，画素の色がどのような順序で塗り替えられていくか，図1のデータが与えられたものとして，最初の五つが塗り替えられる順序を1，2，…，5で示してみてください。

新人：はい。追跡してみます。結果は，［　a　］のようになりました。

先輩：そうですね。では，プログラムを検討していきましょう。まず，元の画素数は8×8ですが，配列Imageの大きさは10×10で，行番号15～23で最外周の配列要素に値0を設定しています。これは何のための処理でしょうか。

新人：はい，［　b　］からです。しかし，表示領域の要素数64に対して，32の配列要素に値を設定するのは，随分無駄な処理のように思えます。

先輩：では，一般にm×n画素からなる表示領域を考えたとき，行番号15～23で値を設定する要素数は，どういう式で表せますか。

新人：はい，［　c　］となります。ということは，表示領域の画素数m×nが大きくなるほど，この要素数は相対的に小さくなっていくのですね。

先輩：次に，最外周の配列要素に設定する値について考えてみましょう。行番号15では，変数Wallに値0を設定しています。与えられた仕様では，画素の色を表す値は1～3の範囲だからこれでよいのですが，もしも，符号なし8ビットで表せる値0～255がすべて色の指定に使えるとした場合，行番号15をどう変更したらよいですか。例を一つ挙げてみてください。

新人：行番号15を［　d　］と変更するのも一つの方法です。

先輩：次は，行番号12～14について考えてみます。これは何の処理ですか。

新人：現在の色と同じ色で塗り替えようとしているなら，以降の処理をせずに呼出し元に戻ります。塗り替えても，結局元どおりなので無駄ですから。

先輩：それも理由の一つではあるけれど。では，このプログラムから行番号12～14を取り去ってみましょう。そして，図2の①～③に示す，1画素，2画素，3画素からなる三つの白色の領域の例について，同じ色で塗り替えようとし

た場合のプログラムの動作を追跡してみてください。変数 VS，HS には，太枠で示した画素を指定するものとします。

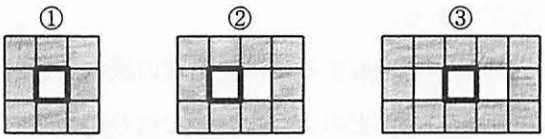


図2　1画素，2画素，3画素からなる領域

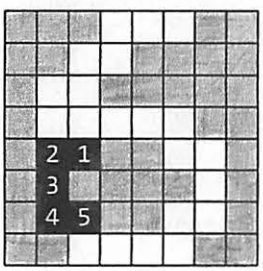
新人：では，追跡してみます。結果は，次の表のとおりです。この結果から，行番号12～14は省略できない必要な処理であることが分かりました。

表　追跡の結果

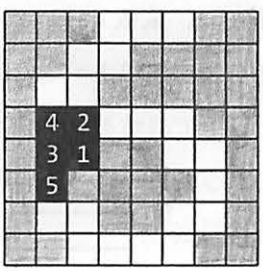
テストケース	プログラムの動作
図2の①（1画素からなる領域）	e
図2の②（2画素からなる領域）	f
図2の③（3画素からなる領域）	変数 More の値が増加していき配列の添字の上限を超える。

aに関する解答群

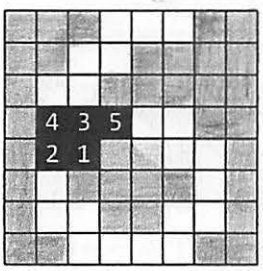
ア



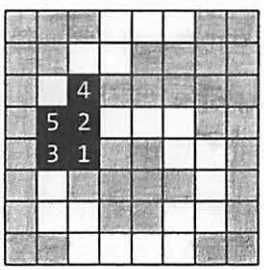
イ



ウ



エ



bに関する解答群

ア　これらの要素が参照されることはないが，添字から表示領域内かどうかが分かる

イ　これらの要素も色 NC での塗替えの対象とすることで，処理を簡素化できる

ウ　配列 Image の各添字の範囲チェックを省略できる

エ　配列 VPos と HPos の各添字の範囲チェックを省略できる

cに関する解答群

ア　$2(m+1)(n+1)$　　イ　$2(m+n)$

ウ　$2(m+n+2)$　　エ　$2(m+n-2)$

dに関する解答群

ア　・Wall ← CC　　イ　・Wall ← NC

ウ　・Wall ← 256 − CC　　エ　・Wall ← 255 − NC

e, fに関する解答群

ア　仕様どおりに同じ色で塗り替えて正しく終了する。

イ　塗替えは一度も実行せずに終了する。

ウ　塗り替えるべき領域に含まれない周辺の画素まで塗り替えて終了する。

エ　変数 More の値が増加していき配列の添字の上限を超える。

オ　変数 More の値は一定値以下であるが処理が無限にループする。

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，答案用紙の選択欄の(選)をマークして解答してください。

なお，2問以上選択した場合には，はじめの1問について採点します。

問9　次のCプログラムの説明及びプログラムを読んで，設問1，2に答えよ。

C

〔プログラムの説明〕

与えられたパスを絶対パスに変換する関数convertである。

階層構造をもつファイルシステムにおいて，ファイルやディレクトリを特定する文字列をパスという。ルートディレクトリを基準としたパスを絶対パスと呼び，“/”から始まり，各階層を“/”で区切っていく。与えられたパスがディレクトリのとき，最後の“/”はあってもなくてもよい。例えば，図のディレクトリeの絶対パスは“/a/d/e”又は“/a/d/e/”で示す。

一方，カレントディレクトリを基準としたパスを相対パスと呼び，相対パスを指定するときに階層を一つ上にたどる場合は“..”を用いる。例えば，図においてディレクトリcをカレントディレクトリにした場合，ファイルfile1.txtの相対パスは“../file1.txt”，ディレクトリeの相対パスは“../../d/e”又は“../../d/e/”となる。また，カレントディレクトリ自身は“.”又は“./”で示す。

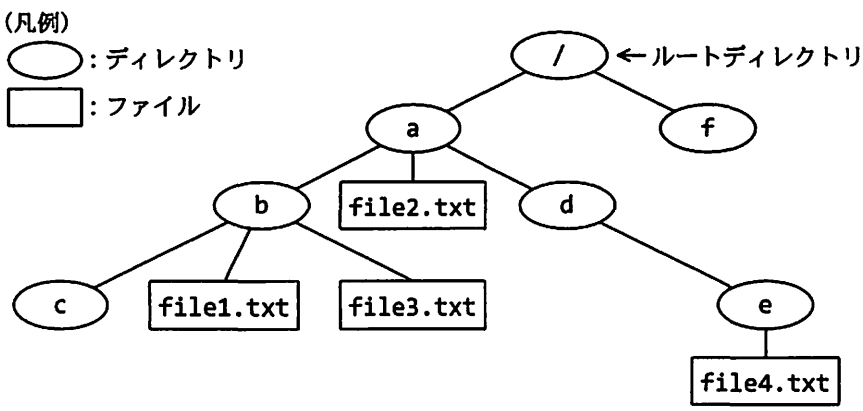


図　階層構造をもつファイルシステムの例

(1) 関数の仕様は，次のとおりである。

```
void convert(const char *path,
             const char *base,
             char *result);
```

引数：path　変換前のパス

base　カレントディレクトリの絶対パス

result　変換後の絶対パス

機能：path が相対パス表記であれば，base を基準にした絶対パス表記に変換し，result に格納する。path が絶対パス表記であれば，result には base に関係なく path をそのまま格納する。

返却値：なし。

ただし，result が参照する領域は，変換後の文字列を格納するのに十分であるとする。また，冗長なパス又はパスとして認識できない文字列が引数として与えられることはないものとする。

(2) ファイルシステム上に，指定されたディレクトリやファイルが実際に存在するかどうかのチェックは行わない。

(3) 変換例を表1に示す。

表1　変換例

path	base	result
../../b/c/	/a/d/e/	/a/b/c/
b/file1.txt	/a/	/a/b/file1.txt
c/	/a/b/	/a/b/c/
file1.txt	/a/b/	/a/b/file1.txt
./	/a/b/c/	/a/b/c/
/a/d/e/file4.txt	/a/b/c/	/a/d/e/file4.txt

(4) 次のライブラリ関数を用いる。

```
unsigned int strlen(const char *s);
```

機能：文字列 s の長さを計算する。

返却値：終端を示すナル文字に先行する文字の個数を返す。

```
int strcmp(const char *s1, const char *s2);
```

機能：文字列 s1 と文字列 s2 を比較する。

返却値：s1 と s2 が同一文字列の場合は 0，それ以外の場合は 0 以外を返す。

```
int strncmp(const char *s1, const char *s2, int n);
```

機能：文字列 s1 と文字列 s2 を先頭から n 文字，又はナル文字までを比較する。

返却値：比較した n 文字が同一の場合は 0 を，それ以外の場合（比較が途中で終了した場合も含む）は 0 以外を返す。

```
char *strcpy(char *s1, const char *s2);
```

機能：文字列 s1 に文字列 s2 をナル文字まで複写する。

返却値：s1

```
char *strncpy(char *s1, const char *s2, int n);
```

機能：文字列 s1 に文字列 s2 を n 文字複写する。s2 の長さが n 以上の場合は n 文字目までを複写し，n 未満の場合は残りをナル文字で埋める。

返却値：s1

〔プログラム〕

```
#include <string.h>

void convert(const char*, const char*, char*);

void convert(const char *path, const char *base, char *result){

   const char *pp, *bp;
   char *rp;
   int length;

   /* path が絶対パス表記の場合 */
   if(*path == '/'){
      [          a          ];
      return;
   }
```

```
    /* path がカレントディレクトリの場合 */
    if(!strcmp(path, ".") || !strcmp(path, "./")){
        [ b ];
        return;
    }

    length = strlen(base);
    bp = base + length;    /* bp は文字列 base の終端を指す。 */
    if(*(bp - 1) == '/')
        --bp;

    /* path の先頭部にある".."又は"../"を解析することで,
       base のパス表記のうち，どこまで result と共通になるかを調べる。 */
    for(pp = path; *pp != '\0' && *pp == '.';){
        if(!strncmp(pp, "../", 3)){
            pp += 3;
            while(bp > base && *--bp != '/');
        }else if(!strncmp(pp, "./", 2)){
            pp += 2;
        }else if(!strncmp(pp, "..\0", 3)){
            pp += 2;
            while(bp > base && *--bp != '/');
        }else{
            break;
        }
    }
    /* base のパス表記と共通な部分を result に複写する。 */
    length = [ c ];
    strncpy(result, base, length);

    rp = [ d ];
    *rp++ = '/';

    /* path の文字列のうち，先頭部分の"./"や"../"を除いた残りの
       部分(pp が指す文字列)を，result の文字列に追加する。 */
    strcpy(rp, pp);
    return;
}
```

設問1　プログラム中の［　　］に入れる正しい答えを，解答群の中から選べ。

a，bに関する解答群

ア　strcpy(base, path)	イ　strcpy(base, result)
ウ　strcpy(path, base)	エ　strcpy(path, result)
オ　strcpy(result, base)	カ　strcpy(result, path)

cに関する解答群

ア bp - base　　イ bp - path

ウ pp - base　　エ pp - path

dに関する解答群

ア base + length　　イ bp + length

ウ path + length　　エ pp + length

オ result + length

設問2　表2の引数列で関数convertを呼んだときのプログラムの動作について，表2中の［　　］に入れる正しい答えを，解答群の中から選べ。ただし，プログラム中の［ a ］～［ d ］には正しい答えが入っているものとする。

表2　引数列

path	base	result
../../../../d/	/a/b/c/	e
d/	/a/b/c	f
d	/a/b/c/	g

eに関する解答群

ア ../d/　　イ ../../../../d/

ウ /　　エ /../d/

オ /d/　　カ d/

f，gに関する解答群

ア /a/b/c/d　　イ /a/b/c/d/

ウ /a/b/cd　　エ /a/b/cd/

オ d　　カ d/

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〕

売上ファイルを読み込み，販売店コード別に売上集計を行い，売上分析表を印刷する副プログラムである。

(1) 売上ファイルSALES-Fは順ファイルである。

(2) 売上ファイルのレコード様式を図1に示す。

販売店コード 10けた	商品コード 10けた	顧客コード 10けた	売上金額 10けた

図1　売上ファイルSALES-Fのレコード様式

① 売上金額はゼロより大きいものとする。

② すべてのレコードの売上金額を合計しても10けたを超えないものとする。

③ 売上ファイルのデータに誤りはないものとする。

(3) 販売店コードで集計して印刷した売上分析表の例を図2に示す。

コード	売上金額	累積比率%	1++++ … +70++++++++80++++++++90+++++++100
TKY-01	13,345,000	51.0	***** …
OSK-01	6,350,000	75.4	***** … ********
TKY-02	2,330,000	84.3	***** … **********
FUK-01	1,984,000	91.9	***** … **********
NGY-01	1,453,000	97.4	***** … **********
SAP-01	528,000	99.5	***** … **********
HIR-01	130,000	100.0	***** … **********
TOTAL	26,120,000		

図2　売上分析表の例

① 売上分析表の見出しは印刷済とする。

② 売上分析表は，集計された売上金額の多い順に整列し，印刷する。

③ 累積比率は，当該販売店から上位順位店すべての売上金額の累積を売上金額の総合計で割った比率（%）の小数第2位を切り捨てたものである。

④ 文字“*”の並びは，この累積比率の整数部分の値mをm個の“*”で示したものである。

〔プログラム〕

```
(行番号)
     1  DATA DIVISION.
     2  FILE SECTION.
     3  FD   SALES-F.
     4  01   SALES-R.
     5       05  S-SHOP-CD          PIC X(10).
     6       05  S-PRODUCT-CD       PIC X(10).
     7       05  S-CUSTOMER-CD      PIC X(10).
     8       05  S-SALES            PIC 9(10).
     9  FD   TEMP-F.
    10  01   TEMP-R.
    11       05  TEMP-CD            PIC X(10).
    12       05  FILLER             PIC X(20).
    13       05  TEMP-SALES         PIC 9(10).
    14  FD   PRINT-F.
    15  01   PRINT-R.
    16       05  P-CD               PIC X(10).
    17       05  FILLER             PIC X.
    18       05  P-SALES            PIC Z,ZZZ,ZZZ,ZZ9.
    19       05  FILLER             PIC X(5).
    20       05  P-RATIO            PIC ZZ9.9.
    21       05  FILLER             PIC X(2).
    22       05  P-ASTER            PIC X(100).
    23  SD   SORT-F.
    24  01   SORT-R.
    25       05  SORT-CD            PIC X(10).
    26       05  FILLER             PIC X(20).
    27       05  SORT-SALES         PIC 9(10).
    28  WORKING-STORAGE SECTION.
    29  01   EOF-SW                 PIC 9.
    30  01   GROSS-SUM              PIC 9(10).
    31  01   INTERIM-SUM            PIC 9(10).
    32  01   W-SORT-CD              PIC X(10).
    33  01   W-RATIO                PIC 999.
    34  PROCEDURE DIVISION.
    35  MAIN-CTL.
    36       SORT SORT-F [          a          ]
    37            INPUT  PROCEDURE   SORT-1-INPUT
    38            GIVING TEMP-F.
    39  *
    40       SORT SORT-F [          b          ]
    41            INPUT  PROCEDURE   SORT-2-INPUT
    42            OUTPUT PROCEDURE   SORT-2-OUTPUT.
    43  EXIT PROGRAM.
```

```
44 *
45  SORT-1-INPUT.
46      OPEN INPUT SALES-F.
47      MOVE 0 TO EOF-SW.
48      PERFORM UNTIL EOF-SW = 1
49         READ SALES-F AT END MOVE 1 TO EOF-SW
50              NOT AT END
51                  MOVE SALES-R TO SORT-R
52                  RELEASE SORT-R
53         END-READ
54      END-PERFORM.
55      CLOSE SALES-F.
56 *
57  SORT-2-INPUT.
58      OPEN INPUT TEMP-F.
59      MOVE 0 TO GROSS-SUM.
60      MOVE 0 TO EOF-SW.
61      PERFORM UNTIL EOF-SW = 1
62         READ TEMP-F AT END MOVE 1 TO EOF-SW
63              NOT AT END
64                  IF GROSS-SUM = 0 OR
65                     W-SORT-CD NOT = TEMP-CD THEN
66                     IF GROSS-SUM NOT = 0 THEN
67                        RELEASE SORT-R
68                     END-IF
69                     MOVE TEMP-CD  TO W-SORT-CD
70                                      SORT-CD
71                     [          c          ]
72                  ELSE
73                     [          d          ]
74                  END-IF
75                  [          e          ]
76         END-READ
77      END-PERFORM.
78      IF GROSS-SUM NOT = 0 THEN
79         RELEASE SORT-R
80      END-IF.
81      CLOSE TEMP-F.
82 *
83  SORT-2-OUTPUT.
84      OPEN OUTPUT PRINT-F.
85      MOVE 0 TO INTERIM-SUM.
86      MOVE 0 TO EOF-SW.
87      PERFORM UNTIL EOF-SW = 1
88         RETURN SORT-F AT END MOVE 1 TO EOF-SW
89                NOT AT END
90                    ADD SORT-SALES TO INTERIM-SUM
91                    PERFORM DETAIL-PRINT-PROC
92         END-RETURN
93      END-PERFORM.
```

```
 94       PERFORM FOOTER-PRINT-PROC.
 95       CLOSE PRINT-F.
 96 *
 97   DETAIL-PRINT-PROC.
 98       MOVE SPACE TO PRINT-R.
 99       MOVE SORT-CD    TO P-CD.
100       MOVE SORT-SALES TO P-SALES.
101       COMPUTE P-RATIO W-RATIO = INTERIM-SUM * 100 / GROSS-SUM.
102       IF W-RATIO > 0 THEN
103          MOVE ALL "*" TO P-ASTER(1:W-RATIO)
104       END-IF.
105       WRITE PRINT-R AFTER 1.
106   FOOTER-PRINT-PROC.
107       MOVE SPACE TO PRINT-R.
108       MOVE "TOTAL"   TO P-CD.
109       MOVE GROSS-SUM TO P-SALES.
110       WRITE PRINT-R AFTER 1.
```

COBOL

設問1　プログラム中の［　　］に入れる正しい答えを，解答群の中から選べ。

a，bに関する解答群

ア　ASCENDING　KEY SORT-CD

イ　ASCENDING　KEY SORT-SALES

ウ　DESCENDING KEY SORT-SALES

エ　DESCENDING KEY TEMP-CD

c～eに関する解答群

ア　ADD　TEMP-SALES TO GROSS-SUM

イ　ADD　TEMP-SALES TO INTERIM-SUM

ウ　ADD　TEMP-SALES TO SORT-SALES

エ　MOVE TEMP-SALES TO GROSS-SUM

オ　MOVE TEMP-SALES TO INTERIM-SUM

カ　MOVE TEMP-SALES TO SORT-SALES

設問2　この副プログラムを，商品コード，顧客コードでの集計にも使えるように変更する。パラメタの値が，1のときは販売店コード，2のときは商品コード，3のときは顧客コードによる集計となるようにする。表中の［　　］に入れる正しい答えを，解答群の中から選べ。

表　プログラムの変更内容

処置	変更内容
行番号8と9の間に追加	01　SALES-REC-R. 　　05　［　　f　　］. 　　05　FILLER　　PIC 9(10).
行番号34を変更	LINKAGE SECTION. 01　SUM-CD PIC 9. PROCEDURE DIVISION USING SUM-CD.
［　g　］に追加	MOVE S-CD(SUM-CD) TO SORT-CD

fに関する解答群

ア　S-CD　　PIC X(10)

イ　S-CD　　PIC X(10) OCCURS 3

ウ　S-CD　　PIC X(10) OCCURS 3 INDEXED BY SUM-CD

エ　S-CD　　PIC X(20)

オ　S-CD　　PIC X(30)

gに関する解答群

ア　行番号50と51の間

イ　行番号51と52の間

ウ　行番号66と67の間

エ　行番号78と79の間

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〕

ギャップバッファを利用した簡易テキストエディタである。

ギャップバッファとは，編集対象となる文字列の編集箇所に空き領域を作り，そこに文字を挿入する機構をもつバッファのことである。一般に，テキストの編集時には，文字の挿入，削除などの変更は局所的に集中することが多く，編集箇所にあらかじめ空き領域を作っておくと効率よく変更を行うことができるという利点がある。ギャップバッファ内の空き領域をギャップといい，ギャップバッファ内の文字列をテキストという。ここで，ギャップは，ギャップバッファ内に一つしか存在しない。ただし，文字の挿入によって，ギャップがない状態になる場合がある。

テキストの表示イメージとギャップバッファを図1に示す（網掛けの部分がギャップである）。

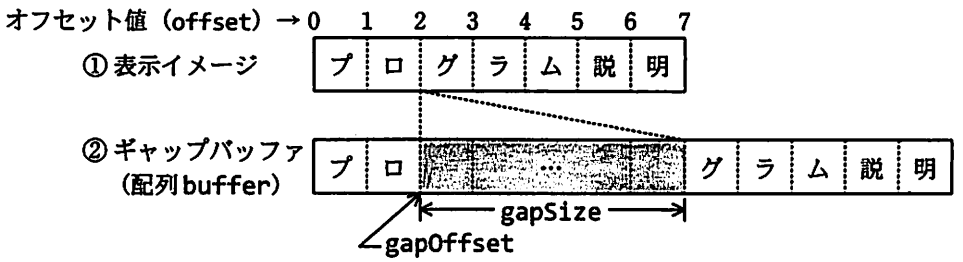


図1　テキストの表示イメージとギャップバッファ

クラス GapBuffer はギャップバッファの機能を実現する。このクラスの使用者は，テキストを仮想的に連続する文字列として扱うことができ，ギャップの位置やサイズを意識せずに文字の挿入や削除の処理を行う。このとき，図1①のように各文字はオフセット値（offset）で指定し，テキストの最初の文字はオフセット値0，テキストの最後の文字はオフセット値“テキストの文字数－1”で指定する。

図1②は，配列 buffer 内におけるテキストの物理的な表現の例である。フィールド gapOffset と gapSize は，それぞれギャップの先頭位置とギャップのサイズを表す。

図1②の状態から“プログラム”と“説明”の間に“の”を挿入する手順を図2に示す。

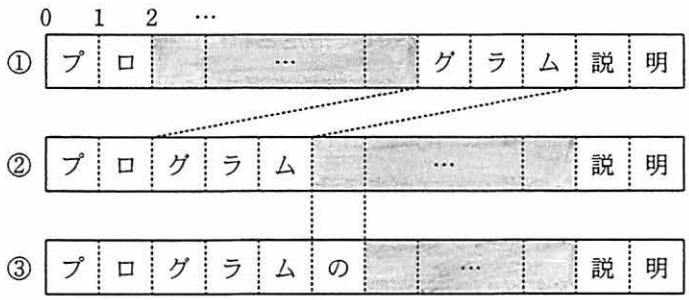


図2　挿入する手順

まず，“プロ”の直後に“グラム”を移動することで，ギャップを“プログラム”と“説明”の間に移動する（図2②）。ギャップの先頭に“の”を格納する（図2③）。この操作によって，ギャップのサイズは1文字分小さくなり，ギャップの先頭位置は1文字分後ろへずれる。

次に，図2③の状態から“プログラムの”の“グ”を削除する手順を図3に示す。

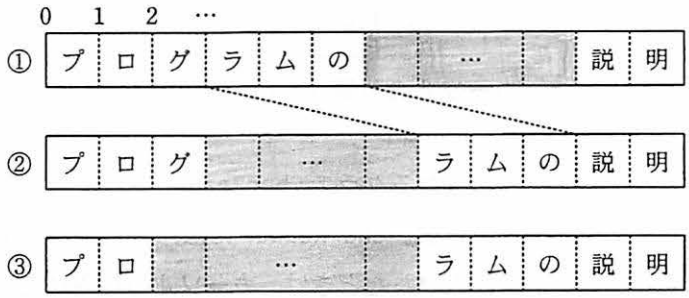


図3　削除する手順

まず，“ラムの”を“説明”の直前に移動することで，ギャップを“プログ”と“ラムの説明”の間に移動する（図3②）。ギャップの先頭位置を1文字分前にずらし，ギャップのサイズを1文字分増やす（図3③）。

クラス`GapBuffer`は，次のコンストラクタ及びメソッドをもつ。

(1) コンストラクタは引数`initialText`で指定された文字列をテキストの初期値としてギャップバッファを生成する。ギャップは，テキストの前に作られる。

(2) メソッド`insert`は，メソッド`confirmGap`を呼んで引数`offset`の位置にギャップを移動した上で，引数`offset`で指定された位置に，引数`ch`で指定された文字を挿入する。

(3) メソッド`delete`は，引数`offset`で指定された位置にある文字を削除する。テキストがない場合は，何もしない。

(4) メソッド charAt は，引数 offset で指定された位置の文字を返す。

(5) メソッド length は，テキストの文字数を返す。

(6) メソッド confirmGap は，配列 buffer において引数 newGapOffset で与えられた位置に 1 文字以上のギャップがあるようにする。必要であればギャップを指定された位置に移動し，ギャップがない場合は，新たにギャップを作る。

なお，このクラスの各メソッドが呼び出されるとき，引数は正しいものとする。

クラス Editor は，GapBuffer を利用して，簡易テキストエディタを実現する。メソッド main に与えられた最初の引数をテキストの初期値とする。カーソルは，文字の間にあり，カーソルの位置はオフセット値で表し，フィールド cursor に保持される。例えば，図 4 のようにカーソルが“プロ”と“グラム”の間にあるとき，cursor の値は 2 である。カーソルがテキストの末尾にあるとき，cursor の値はテキストの文字数と同じであり，テキストがないとき，cursor の値は 0 である。

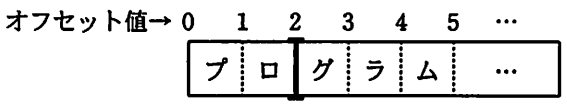


図4　カーソルの位置（cursor の値は 2）

文字の入力は，外部で与えられるクラス CharReader で行う。メソッド get は，ターミナルから 1 文字ずつ読み込む。CharReader には，次の制御文字が定義されている。制御文字以外の文字が入力されたときは，カーソルの位置に文字を挿入し，カーソルを 1 文字分進める。

① CharReader.MOVE_FORWARD

カーソルを 1 文字分進める。カーソルがテキストの末尾にあるときは，無視する。

② CharReader.MOVE_BACKWARD

カーソルを 1 文字分戻す。カーソルがテキストの先頭にあるときは，無視する。

③ CharReader.DELETE

カーソルの直後の 1 文字を削除する。カーソルがテキストの末尾にあるときは，無視する。

④ CharReader.EOF

エディタを終了する。

ギャップバッファの内容の表示は，外部で与えられるクラス Display で行う。メソッド output は，GapBuffer のメソッド charAt を呼び出してギャップバッファのテキストを取得する。

〔プログラム1〕

```
class GapBuffer {
   private static final int INITIAL_GAP_SIZE = 128;
   private char[] buffer;
   private int gapOffset = 0;
   private int gapSize = INITIAL_GAP_SIZE;

   GapBuffer(String initialText) {
      buffer = new char[initialText.length() + gapSize];
      System.arraycopy(initialText.toCharArray(), 0,
                       buffer, gapSize, initialText.length());
   }

   void insert(int offset, char ch) {
      confirmGap(offset);
      buffer[gapOffset++] = ch;
      [   a   ];
   }

   void delete(int offset) {
      if (length() == 0)
         return;
      confirmGap(offset + 1);
      gapOffset--;
      gapSize++;
   }

   char charAt(int offset) {
      if (offset >= gapOffset)
         offset += [   b   ];
      return buffer[offset];
   }

   int length() { return [   c   ]; }

   private void confirmGap(int newGapOffset) {
      if (gapSize == 0) {
         char[] temp = new char[buffer.length + INITIAL_GAP_SIZE];
         System.arraycopy(buffer, 0, temp, 0, buffer.length);
         gapOffset = buffer.length;
         gapSize = INITIAL_GAP_SIZE;
         buffer = temp;
      }
      if (newGapOffset < gapOffset) {
         System.arraycopy(buffer, newGapOffset, buffer,
                newGapOffset + gapSize, gapOffset - newGapOffset);
      } else {
         System.arraycopy(buffer, gapOffset + gapSize,
                buffer, gapOffset, newGapOffset - gapOffset);
      }
```

```
      gapOffset = newGapOffset;
    }
  }
```

〔プログラム2〕

```
class Editor {
    private [ d ] buf;
    private int cursor = 0;

    private Editor(String text) {
        buf = new GapBuffer(text);
    }

    private void run() {
        Display.output(buf, cursor);
        char ch;
        while ((ch = CharReader.get()) != CharReader.EOF) {
            switch (ch) {
            case CharReader.MOVE_FORWARD:
                moveCursor(1);
                break;
            case CharReader.MOVE_BACKWARD:
                moveCursor(-1);
                break;
            case CharReader.DELETE:
                if (cursor < buf.length()) {
                    buf.delete([ e ]);
                }
                break;
            default:
                buf.insert([ f ], ch);
                break;
            }
            Display.output(buf, cursor);
        }
    }

    private void moveCursor(int n) {
        int newCursor = cursor + n;
        if (newCursor >= 0 && newCursor <= buf.length()) {
            cursor = newCursor;
        }
    }

    public static void main(String[] args) {
        Editor editor = new Editor(args[0]);
        editor.run();
    }
}
```

設問　プログラム中の［　　］に入れる正しい答えを，解答群の中から選べ。

aに関する解答群

ア　gapSize--　　イ　gapSize++

ウ　gapSize - 1　　エ　gapSize + 1

bに関する解答群

ア　buffer.length　　イ　gapOffset

ウ　gapSize　　エ　length()

cに関する解答群

ア　buffer.length

イ　buffer.length - gapOffset

ウ　buffer.length - gapSize

エ　buffer.length - INITIAL_GAP_SIZE

dに関する解答群

ア　CharReader　　イ　Display　　ウ　Editor

エ　GapBuffer　　オ　Object　　カ　StringBuffer

e，fに関する解答群

ア　--cursor　　イ　++cursor　　ウ　cursor

エ　cursor--　　オ　cursor++

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。

〔プログラムの説明〕

32ビットの乗算を行う副プログラムMULSである。

(1) MULSは，32ビットの被乗数と16ビットの乗数を受け取り，32ビットの積を返す。数値はすべて符号なし整数とし，積のオーバフローは考慮しない。

被乗数と積は，それぞれ連続する2語に上位16ビット，下位16ビットの順に格納される。それぞれの上位語のアドレスは，GR1とGR3に設定して渡される。

乗数はGR2に直接設定して渡される。

(2) 副プログラムMULS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〕

```
MULS    START               ; 32ビット×16ビット → 32ビット
        RPUSH
        LAD     GR6,0       ; 積　上位語の初期化
        LAD     GR7,0       ; 積　下位語の初期化
        LD      GR4,0,GR1   ; 被乗数　上位語の取出し
        LD      GR5,1,GR1   ; 被乗数　下位語の取出し
LP      SRL     GR2,1       ; 乗数を1ビット右にシフト
        [      a      ]
        JZE     FIN
        JUMP    NEXT        ; 加算処理をスキップ
ADD32   ADDL    GR6,GR4     ; 32ビット＋32ビット → 32ビット
        ADDL    GR7,GR5
        [      b      ]
        JUMP    NEXT
ADJ1    ADDL    GR6,=1      ; けた上げ処理
NEXT    SLL     GR4,1       ; 被乗数(32ビット)を1ビット左にシフト
        [      c      ]
        JOV     ADJ2
        JUMP    LP
ADJ2    OR      GR4,=1
        JUMP    LP
FIN     ST      GR6,0,GR3   ; 乗算結果の格納
        ST      GR7,1,GR3
        RPOP
        RET
        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。

aに関する解答群

ア	JMI	ADD32	イ	JMI	LP
ウ	JOV	ADD32	エ	JOV	LP
オ	JPL	ADD32	カ	JPL	LP

bに関する解答群

ア	JMI	ADJ1	イ	JMI	ADJ2
ウ	JOV	ADJ1	エ	JOV	ADJ2
オ	JPL	ADJ1	カ	JPL	ADJ2

cに関する解答群

ア	JNZ	LP	イ	JPL	LP
ウ	JZE	LP	エ	SLL	GR5,1
オ	SRA	GR5,1	カ	SRL	GR5,1

設問2　MULSを利用して，32ビット同士の乗算を行う副プログラムMULを作成した。プログラム2中の［　　］に入れる正しい答えを，解答群の中から選べ。

(1) MULは，被乗数と乗数を受け取り，積を返す。数値はすべて32ビットの符号なし整数とし，積のオーバフローは考慮しない。

被乗数，乗数，積は，それぞれ連続する2語に上位16ビット，下位16ビットの順に格納される。それぞれの上位語のアドレスは，GR1，GR2，GR3に設定して渡される。

(2) 副プログラムMUL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〕

```
MUL     START               ; 32ビット×32ビット → 32ビット
        RPUSH
        PUSH  0,GR3
        PUSH  0,GR2
        LD    GR2,1,GR2     ; 乗数下位語を取り出してGR2に設定
        CALL  MULS          ; 被乗数×乗数下位語 → 積(A)
        POP   GR2
        [      d      ]
        LAD   GR3,SV        ; 結果の格納先として作業領域を設定
        CALL  MULS          ; 被乗数×乗数上位語 → 積(B)
        [      e      ]
        POP   GR3
        ADDL  GR6,0,GR3     ; 積(A)の上位語と積(B)の下位語を加算
        ST    GR6,0,GR3
        RPOP
        RET
SV      DS    2
        END
```

dに関する解答群

ア　LD　GR1,0,GR1　　イ　LD　GR1,0,GR2

ウ　LD　GR1,1,GR1　　エ　LD　GR2,0,GR1

オ　LD　GR2,0,GR2　　カ　LD　GR2,1,GR1

eに関する解答群

ア	LD	GR6,0,GR1	イ	LD	GR6,0,GR2
ウ	LD	GR6,0,GR3	エ	LD	GR6,1,GR1
オ	LD	GR6,1,GR2	カ	LD	GR6,1,GR3

問13　次の表計算及びワークシートの説明を読んで，設問に答えよ。

〔表計算の説明〕

E社は，3種類の商品M1～M3を自社で生産し，3か所の販売拠点P1～P3で販売している。販売する商品は，前月までに生産したものである。

(1) E社では，各販売拠点で2月に立てた4～9月の半年分の商品ごとの月別販売計画を基に，次の①及び②に従って3～8月の半年分の月別生産計画を作成している。

① 過去の経験から，販売計画量以上に売れる月があるので，各月の在庫量は，翌月の販売計画量に20％の余裕を加えた数量とする。

② 商品ごとに1か月間に生産できる最大の数量（以下，最大月産量という）が決まっているので，前月にすべてを生産できない場合がある。その場合，不足分は前々月までに生産するように計画する。

(2) 各販売拠点で2月に立てた4～9月の半年分の商品ごとの月別販売計画は，表計算ソフトのワークシート“販売計画”に1行ずつ入力する。そのワークシートの例を図1に示す。

	A	B	C	D	E	F	G	H
1	拠点名	商品名	4月計画量	5月計画量	6月計画量	7月計画量	8月計画量	9月計画量
2	P1	M1	1,000	1,100	2,300	1,200	1,500	1,200
3	P1	M2	500	1,500	800	50	40	20
4	P1	M3	500	600	500	400	500	1,200
5	P2	M1	2,000	2,000	2,000	2,000	2,000	2,000
6	P2	M2	400	1,000	400	30	40	60
7	P2	M3	300	400	500	600	700	800
8	P3	M1	1,000	1,500	1,200	1,600	1,400	1,800
9	P3	M2	600	1,000	800	70	50	40
10	P3	M3	200	500	1,000	500	800	1,000

図1　月別販売計画を記録したワークシート“販売計画”の例

(3) 図1のワークシート“販売計画”を基に，商品M2について3～8月の半年分の月別生産計画を作成したワークシート“生産計画”の例を図2に示す。

	A	B	C	D	E	F	G	H	I	J
1	商品名	最大月産量		3月	4月	5月	6月	7月	8月	9月
2	M2	3,000	販売計画量		1,500	3,500	2,000	150	130	120
3			必要在庫量	1,800	4,200	2,400	180	156	144	
4			繰越在庫量	0	300	700	400	250	120	
5			必要生産量	1,800	3,900	1,700	0	0	24	
6			繰上生産量	0	900	0	0	0	0	0
7			計画生産量	2,700	3,000	1,700	0	0	24	

注　網掛けの部分には，データは入らない。

図2　商品M2の月別生産計画作成後のワークシート“生産計画”の例

(4)　一つの商品についての月別生産計画の作成は，次の手順で行う。

①　4～9月の期間について，E社全体での月ごとの販売計画量を求める。

②　3～8月の期間について，月ごとに新たに必要となる商品の数量（以下，必要生産量という）を，最大月産量を考慮せずに求める。

③　3～8月の各月における必要生産量を確保するために，最大月産量を考慮して，月ごとに生産する商品の数量（以下，計画生産量という）を求める。

(5)　各月の必要生産量の求め方は，次のとおりである。

必要生産量は，当月の必要在庫量から繰越在庫量を差し引いた数量である。ただし，繰越在庫量が必要在庫量以上ある場合は，商品の生産は必要ないので0となる。

①　必要在庫量は最低限もつ必要がある在庫の量で，翌月の販売計画量の120％とする。

②　繰越在庫量とは，前月の繰越在庫量と必要生産量の合計から，当月の販売計画量を差し引いた数量である。

なお，3月の繰越在庫量は0である。

(6)　各月の計画生産量の求め方は，次のとおりである。

計画生産量は，当月の必要生産量と翌月の繰上生産量の合計である。ただし，この数量が最大月産量を超える場合は，計画生産量は最大月産量とする。

繰上生産量とは，当月の必要生産量と翌月の繰上生産量の合計から，当月の計画生産量を差し引いた数量である。

(7)　ワークシート“生産計画”で用いる関数を表に示す。

表　ワークシート“生産計画”で用いる関数

書式	説明
照合合計(照合値,照合範囲,対応範囲)	照合範囲のセルにおいて，照合値と等しい値をもつセルをすべて探し出す。そして，照合値と等しい値をもつセルの相対位置と同じ位置にある対応範囲のセルの値（数値）を合計して返す。
切上げ(実数値)	実数値の小数点以下を切り上げた整数を返す。

〔ワークシート：生産計画〕

ワークシート“生産計画”の作成方法は，次のとおりである。

(1) セルA2に，商品名としてM1～M3のいずれかを入力する。

(2) セルB2に，セルA2の商品の最大月産量を入力する。

(3) セルE2～J2に，4～9月の販売計画量を計算するための式を入力する。これを実行するために，セルE2に次の式を入力し，セルF2～J2に複写する。複数のワークシート間でデータを参照する場合には“ワークシート名!セル”又は“ワークシート名!セル範囲”という形式で指定する。

照合合計($A2,販売計画![a],販売計画![b])

(4) セルD3～I3に，3～8月の必要在庫量（小数点以下切上げ）を計算するための式を入力する。これを実行するために，セルD3に次の式を入力し，セルE3～I3に複写する。

[c]

(5) セルD4に，3月の繰越在庫量として0を入力する。セルE4～I4に，4～8月の繰越在庫量を計算するための式を入力する。これを実行するために，セルE4に次の式を入力し，セルF4～I4に複写する。

[d]

(6) セルD5～I5に，3～8月の必要生産量を計算するための式を入力する。これを実行するために，セルD5に次の式を入力し，セルE5～I5に複写する。

IF([e],0,D3－D4)

(7) セルJ6に，9月の繰上生産量として0を入力する。セルD6～I6に，3～8月の繰上生産量を計算するための式を入力する。これを実行するために，セルI6に次の式を入力し，セルD6～H6に複写する。

[f]

表計算

(8) セルD7～I7に，3～8月の計画生産量を計算するための式を入力する。これを実行するために，セルI7に次の式を入力し，セルD7～H7に複写する。

IF(g ,I5＋J6,$B2)

設問 ワークシート“生産計画”の説明中の［　　］に入れる正しい答えを，解答群の中から選べ。

a，bに関する解答群

ア　B2～B10	イ　C2～C10	ウ　H2～H10
エ　$B2～$B10	オ　$C2～$C10	カ　$H2～$H10

cに関する解答群

ア　切上げ(E2)＊0.2	イ　切上げ(E2)＊1.2
ウ　切上げ(E2＊0.2)	エ　切上げ(E2＊1.2)

dに関する解答群

ア　D3＋D4－D5	イ　D3＋D4－E2	ウ　D3＋D5－D4
エ　D3＋D5－E2	オ　D4＋D5－D3	カ　D4＋D5－E2

eに関する解答群

ア　D3≦D4	イ　D3≧D4	ウ　D3＜0
エ　D3≧0	オ　D4＜0	カ　D4≧0

fに関する解答群

ア　I5＋J6　　イ　I5＋J6－I7　　ウ　I7＋J6　　エ　I7＋J6－I5

gに関する解答群

ア　I5＋J6≦$B2	イ　I5＋J6＞$B2
ウ　I5≦$B2	エ　I5＞$B2
オ　J6≦$B2	カ　J6＞$B2

■Javaプログラムで使用するAPIの説明

java.lang

public final class System

有用なクラスフィールドやクラスメソッドが定義されている。

メソッド

public static void arraycopy(Object src, int srcPos, Object dest,
int destPos, int length)

指定された複写元の配列の指定位置から，指定された複写先配列の指定位置に配列要素を指定された個数分複写する。引数 src と引数 dest が同じ配列を参照している場合は，複写元と複写先の領域の重なりを考慮し，複写によるデータの破壊が起こらないようにする。

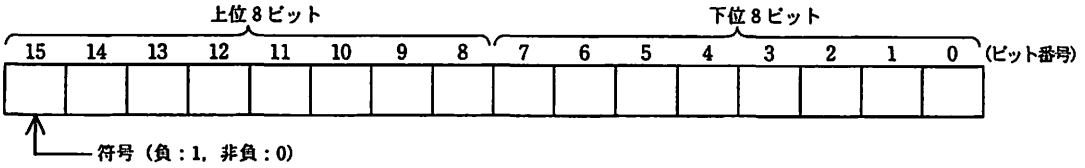
引数： src — 複写元配列
srcPos — 複写元配列内の開始位置
dest — 複写先配列
destPos — 複写先配列内の開始位置
length — 複写される配列要素の個数

■アセンブラ言語の仕様

1. システム COMETⅡの仕様

1.1 ハードウェアの仕様

(1) 1語は16ビットで，そのビット構成は，次のとおりである。



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

(3) 数値は，16ビットの2進数で表現する。負数は，2の補数で表現する。

(4) 制御方式は逐次制御で，命令語は1語長又は2語長である。

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

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

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

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

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

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

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

1.2 命令

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

命　令	書き方		命　令　の　説　明	FRの設定
	命令コード	オペランド		

(1) ロード，ストア，ロードアドレス命令

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

(2) 算術，論理演算命令

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

(3) 比較演算命令

算術比較 ComPare Arithmetic	CPA	r1,r2	(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
		r,adr [,x]		
論理比較 ComPare Logical	CPL	r1,r2		
		r,adr [,x]		

比較結果	FR の値 SF	FR の値 ZF
(r1) ＞ (r2)	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)		

(4) シフト演算命令

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

(5) 分岐命令

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

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

(6) スタック操作命令

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

(7) コール，リターン命令

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

(8) その他

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

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

1.3 文字の符号表

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

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

2. アセンブラ言語 CASLⅡの仕様

2.1 言語の仕様

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

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

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

2.2 命令の種類

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

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

2.3 アセンブラ命令

アセンブラ命令は，アセンブラの制御などを行う。

(1)

START	[実行開始番地]

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

(2) | END | |

END 命令は，プログラムの終わりを定義する。

(3) | DS | 語数 |

DS 命令は，指定した語数の領域を確保する。

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

(4) | DC | 定数［,定数］… |

DC 命令は，定数で指定したデータを（連続する）語に格納する。

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

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

2.4 マクロ命令

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

(1) | IN | 入力領域,入力文字長領域 |

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

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

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

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

(2) | OUT | 出力領域,出力文字長領域 |

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

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

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

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

(3) RPUSH

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

(4) RPOP

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

2.5 機械語命令

機械語命令のオペランドは，次の形式で記述する。

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

2.6 その他

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

(2) 生成された命令語，領域は，主記憶上で連続した領域を占める。

3. プログラム実行の手引

3.1 OS

プログラムの実行に関して，次の取決めがある。

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

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

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

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

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

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

3.2 未定義事項

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

表計算ソフトの機能・用語

表計算ソフトの機能，用語などは，原則として次による。

1. ワークシート

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

2. セル

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

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

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

(4) データが入力されていないセルを，空白セルという。

3. セルへの入力

(1) セルに数値，文字列，計算式を入力できる。

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

(3) セルA1に数値5を入力するときは，“セルA1に5を入力”と表す。

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

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

4. セルの内容の表示

(1) セルに数値を入力すると，右詰めで表示される。

(2) セルに文字列を入力すると，左詰めで表示される。

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

(4) セルの内容の表示については，左詰め，中央揃え，右詰めに変更できる。

5. 計算式

(1) 計算式には，数学で用いられる数式が利用できる。

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

(3) 算術演算子による計算の優先順位は，数学での優先順位と同じである。

6. 再計算

(1) セルに計算式を入力すると，直ちに計算結果を表示する。

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

7. 関数

(1) 計算式には次の表で定義する関数を利用することができる。

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

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

8. セルの複写

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

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

9. 絶対参照

(1) 計算式を複写しても参照したセルが変わらない参照を絶対参照といい，記号$を用いて$A$1などと表す。例えば，セルB1に計算式$A$1+5を入力した場合，セルB1をセルC4に複写してもセルC4の計算式はA1+5のままである。

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

10. マクロ

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

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

11. その他

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

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

退室可能時間	13:40 ～ 15:20

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