FE

平成26年度　春期
基本情報技術者試験
午後　問題

試験時間	13:00 ～ 15:30（2時間30分）

注意事項

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. **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。**
4. 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2～問7	問8	問9 ～ 問13
選択方法	必須	4問選択	必須	1問選択

5. 答案用紙の記入に当たっては，次の指示に従ってください。
   (1) 答案用紙は光学式読取り装置で読み取った上で採点しますので，B又はHBの黒鉛筆で答案用紙の**マークの記入方法**のとおりマークしてください。マークの濃度がうすいなど，**マークの記入方法**のとおり正しくマークされていない場合は読み取れません。特にシャープペンシルを使用する際には，マークの濃度に十分注意してください。訂正の場合は，あとが残らないように消しゴムできれいに消し，消しくずを残さないでください。
   (2) **受験番号欄**に**受験番号**を，**生年月日欄**に**受験票の生年月日**を記入及びマークしてください。答案用紙の**マークの記入方法**のとおり記入及びマークされていない場合は，採点されないことがあります。生年月日欄については，受験票の生年月日を訂正した場合でも，訂正前の生年月日を記入及びマークしてください。
   (3) **選択した問題**については，次の例に従って，**選択欄**の**問題番号**の(選)をマークしてください。答案用紙の**マークの記入方法**のとおりマークされていない場合は，採点されません。問2～問7について5問以上マークした場合は，はじめの4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3，問4，問6，問7，問9を選択した場合の例〕

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

   (4) **解答**は，次の例題にならって，**解答欄**にマークしてください。答案用紙の**マークの記入方法**のとおりマークされていない場合は，採点されません。

〔例題〕　次の［　　］に入れる正しい答えを，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)	(オ)	(カ)	(キ)	(ク)	(ケ)	(コ)

裏表紙の注意事項も，必ず読んでください。

正　誤　表

平成 26 年 4 月 20 日実施

基本情報技術者試験　午後問題　問 13

・72 ページ　上から 12～14 行目

誤	IF(垂直照合(c1, L$2:M$41, 2, 0) = '*', c1,
	IF(垂直照合(c2, L$2:M$41, 2, 0) = '*', c2,
	顧客リスト!G2))
		下線部分を訂正する。

正	IF(垂直照合(c1, L$2～M$41, 2, 0) = '*', c1,
	IF(垂直照合(c2, L$2～M$41, 2, 0) = '*', c2,
	顧客リスト!G2))	

・72 ページ　b に関する解答群

誤	ア　垂直照合(顧客リスト!F2, K$2:K$10, 1, 0) イ　垂直照合(顧客リスト!F2, K$2:K$10, 1, 1) ウ　垂直照合(顧客リスト!F2 + 1, K$2:K$10, 1, 0) エ　垂直照合(顧客リスト!F2 + 1, K$2:K$10, 1, 1) オ　表引き(K$2:K$10, 照合一致(顧客リスト!F2, K$2:K$10, − 1), 1) カ　表引き(K$2:K$10, 照合一致(顧客リスト!F2 + 1, K$2:K$10, 1), 1)	下線部分を訂正する。
正	ア　垂直照合(顧客リスト!F2, K$2～K$10, 1, 0) イ　垂直照合(顧客リスト!F2, K$2～K$10, 1, 1) ウ　垂直照合(顧客リスト!F2 + 1, K$2～K$10, 1, 0) エ　垂直照合(顧客リスト!F2 + 1, K$2～K$10, 1, 1) オ　表引き(K$2～K$10, 照合一致(顧客リスト!F2, K$2～K$10, − 1), 1) カ　表引き(K$2～K$10, 照合一致(顧客リスト!F2 + 1, K$2～K$10, 1), 1)	

〔問題一覧〕

●問 1（必須問題）

問題番号	出題分野	テーマ
問 1	情報セキュリティ	情報資産についてのリスクアセスメント

●問 2～問 7（6 問中 4 問選択）

問題番号	出題分野	テーマ
問 2	ハードウェア	機械語命令
問 3	ソフトウェア	プログラムの並列実行
問 4	ネットワーク	ネットワークにおけるスループットの改善
問 5	ソフトウェア設計	システム統合に伴うソフトウェア設計
問 6	プロジェクトマネジメント	ファンクションポイント法を用いた工数見積り
問 7	経営戦略・企業と法務	システム移行の作業計画

●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	空き領域の管理

●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）	テキストの編集
問 10	ソフトウェア開発（COBOL）	英語の検定テストの結果管理
問 11	ソフトウェア開発（Java）	雑誌記事のオンライン購読サイト
問 12	ソフトウェア開発（アセンブラ）	文字列中の単語の切出し処理
問 13	ソフトウェア開発（表計算）	顧客情報の匿名化

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は必須問題です。必ず解答してください。

問1　情報資産についてのリスクアセスメントに関する次の記述を読んで，設問1～3に答えよ。

Z社は，従業員数が500の中堅SIベンダである。Z社では，プロジェクト開始前に，プロジェクトで扱う情報資産について，図1に示す自社で定めた手順に従って，リスクアセスメントを実施している。このたび，新規に受注したプロジェクトYに対して，リスクアセスメントを実施することになった。

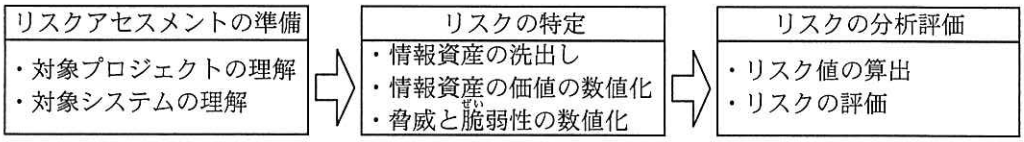


図1　Z社のリスクアセスメントの手順

〔プロジェクトYの説明（抜粋）〕

(1) 顧客が利用する購買システムを開発する。

(2) 開発で利用するテストデータは顧客から提供される。

(3) 顧客のテストデータを格納した顧客のUSBメモリを，プロジェクトメンバが顧客から受け取って自社に持ち帰り，顧客のテストデータを開発用サーバに複写後，USBメモリから削除する。

(4) Z社から顧客の事務所を訪問するのに，電車で1時間30分ほど要する。

(5) 開発用PCでプログラムを開発し，適宜，開発用サーバにアップロードする。

〔Z社の開発環境（抜粋）〕

(1) プログラムの開発には，開発用サーバと開発用PCを利用する。

(2) 開発用サーバは，施錠されたサーバルームに設置されている。

(3) 開発用サーバは，アクセス管理がされており，プロジェクトメンバとシステム管理者だけがアクセスできる。

(4) 開発用PCは，プロジェクト開始時にシステム部から各プロジェクトメンバに貸与され，プロジェクト終了時に返却される。

〔Z社の開発標準（抜粋）〕

(1) 開発時，プロジェクトメンバは顧客のテストデータのうち必要なものだけを，開発用サーバから自分の開発用PCにダウンロードし，不要になったら削除する。

(2) プロジェクト終了時に，プロジェクトマネージャは開発用サーバの顧客のテストデータを削除し，全ての開発用PCから顧客のテストデータが削除されていることを確認する。

〔Z社のリスク値算出方法〕

Z社では，各情報資産のリスク値を，次の式で算出する。

リスク値 ＝ 情報資産の価値 × 脅威 × 脆弱性

ここで，“情報資産の価値”とは情報資産が損なわれたときの影響の大きさを意味し，機密性（以下，Cという），完全性（以下，Iという），可用性（以下，Aという）の観点に対して，影響の大きさをそれぞれ1～3の値で評価する。“脅威”は，発生の可能性の大きさを1～3の値で評価する。“脆弱性”は，脅威が発生した場合に被害が顕在化する度合いの大きさを1～3の値で評価する。ここで，各1～3の値は大きい場合を3，小さい場合を1とする。

C，I，Aごとに算出したリスク値が全て12以下ならばリスクを受容し，そうでないならば追加のリスク対策を実施することにしている。

〔リスクの特定〕

① 情報資産の洗出し

プロジェクトYで扱う情報資産の洗出しを行った。その結果を，表1に示す。

表1 情報資産の洗出し結果

No.	情報資産	作成又は取得	保管場所	廃棄
		⋮		
3	開発用サーバ上の開発中のプログラム	プロジェクトメンバが開発用サーバにアップロードする	開発用サーバ	プロジェクト終了時に削除する
4	顧客のテストデータ	顧客のUSBメモリで受領して，開発用サーバに複写する	顧客のUSBメモリ，開発用サーバ及び開発用PC	
		⋮		

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。

② 情報資産の価値の数値化

表 1 の各情報資産に対して，C，I，A のそれぞれについてその価値を評価した値と評価理由を，表 2 に示す。

表 2 情報資産の価値と評価理由

No.	情報資産	C	I	A	価値の評価理由
⋮					
3	開発用サーバ上の開発中のプログラム	3	3	3	(i) 開発中のプログラムが利用できない場合，プロジェクトの進捗に影響を与える (ii) 社外に漏れた場合，顧客からの信頼を失う (iii) 版管理が行われない場合，不整合によって，プロジェクトの進捗に影響を与える
4	顧客のテストデータ	3	2	1	
⋮					

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。

③ 脅威の数値化

表 2 の情報資産のうち，情報資産 No.4（顧客のテストデータ）について，脅威の内容と脅威の値を，表 3 に示す。

表 3 情報資産 No.4 の脅威の内容と値

No.	脅威 ID	脅威の内容	値
4	T1	顧客のテストデータを格納した顧客の USB メモリを自社に持ち帰る途中で紛失する	3
	T2	開発用サーバが外部から不正アクセスされて顧客のテストデータが盗み出される	1
	T3	ウイルス感染によって顧客のテストデータの破壊又は漏えいが発生する	2
	T4	開発用サーバに複写後，顧客の USB メモリから顧客のテストデータが漏えいする	3
	T5	テスト終了後，不要になった顧客のテストデータが開発用 PC から漏えいする	2
	T6	プロジェクトメンバ又はシステム管理者が顧客のテストデータを開発用サーバから取り出してサーバルームから持ち出す	1
	T7	開発用サーバから顧客のテストデータが滅失する	1

④ 脅威に対する脆弱性の数値化

表3の各脅威に対する脆弱性の低減策と脆弱性の値を，表4に示す。脆弱性の値は，システム，規則又は運用で，二つ以上対策済みなら 1，一つだけなら 2，未対策は3とする。

表4　表3の脅威に対する脆弱性の低減策と値

脅威 ID	脆弱性 ID	脆弱性の低減策	値
T1	Z1	・顧客の USB メモリに顧客のテストデータを保存するときに暗号化してもらう	2
T2	Z2	・脆弱性に対する対策なし	3
T3	Z3	・開発用サーバと開発用 PC にウイルス対策ソフトを導入し，ウイルス定義ファイルを自動更新する ・顧客の USB メモリをウイルスチェックした後に顧客のテストデータを開発用サーバに複写する	1
T4	Z4	・顧客の USB メモリから顧客のテストデータが削除されていることをプロジェクトマネージャが確認する	2
T5	Z5	・開発用 PC から顧客のテストデータが削除されていることをプロジェクトマネージャが確認する	2
T6	Z6	・社員証による入退室管理を行う ・サーバルームに監視カメラを設置する	1
T7	Z7	・脆弱性に対する対策なし	3

〔リスクの分析評価〕

表2～4を基に情報資産 No.4（顧客のテストデータ）のリスクの分析評価を行い，リスク値を算出した結果を，表5に示す。

表5　情報資産 No.4 のリスク値

No.	情報資産の価値			脅威		脆弱性		リスク値			
	C	I	A	脅威 ID	値	脆弱性 ID	値	リスク値 ID	C	I	A
4	3	2	1	T1	3	Z1	2	R1	18	12	6
				T2	1	Z2	3	R2	9	6	3
				T3	2	Z3	1	R3			
				T4	3	Z4	2	R4			
				T5	2	Z5	2	R5			
				T6	1	Z6	1	R6			
				T7	1	Z7	3	R7			

注記　網掛けの部分は表示していない。

プロジェクト Y のプロジェクトマネージャは，リスクの分析評価の結果からリスク対応計画を作成した。その後，リスク対策を実施した。

設問 1　表 2 中の（ii），（iii）は，C，I，A のいずれかの観点から“情報資産の価値”を評価した際の評価理由である。（ii），（iii）に対応する C，I，A の組合せとして適切な答えを，解答群の中から選べ。

解答群

	（ii）	（iii）
ア	A	C
イ	A	I
ウ	C	A
エ	C	I
オ	I	A
カ	I	C

設問 2　情報資産 No.4（顧客のテストデータ）に対するリスクの分析評価の結果，追加のリスク対策が必要になる脅威の数として正しい答えを，解答群の中から選べ。

解答群

ア 1　　イ 2　　ウ 3　　エ 4

設問3　社内でのセキュリティ事故の発生と対策に関する次の記述中の［　　］に入れる適切な答えを，解答群の中から選べ。

プロジェクトYの終了後，新たに発足したプロジェクトXで利用している開発用PCに，プロジェクトYの顧客のテストデータが格納されている，とシステム部に連絡があった。調査した結果，このPCは，プロジェクトYで利用していた開発用PCであり，システム部に返却された後に，システム部からプロジェクトXに貸与されたものであることが判明した。そこで，Z社では，顧客のテストデータの漏えいというリスクに対処するために，［ a ］，［ b ］という対策を追加することにした。

解答群

ア　開発用サーバのアクセスログをシステム部が定期的に確認する

イ　顧客のテストデータを開発用PCにダウンロードして利用する場合は，管理台帳にダウンロード日，削除日，実施者を記入する

ウ　顧客のテストデータを開発用PCに保存する際に，警告メッセージが表示されるようにする

エ　プロジェクトごとに新たに開発用サーバを用意する

オ　プロジェクトメンバが開発用サーバ上の顧客のテストデータにアクセスする権限を参照だけに設定する

カ　返却された開発用PCは，システム部が全データを完全消去する工程を追加する

次の**問2**から**問7**までの6問については，この中から4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，5問以上マークした場合には，はじめの4問について採点します。

問2　機械語命令に関する次の記述を読んで，設問1，2に答えよ。

命令語の形式を，図1に示す。

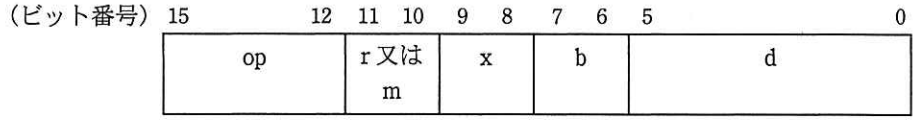


図1　命令語の形式

図1で使用している記号の説明を，表1に示す。数字の末尾にhが付いているものは16進数表記である。

表1　記号の説明

記号	説明
op	命令コード
r	レジスタ番号
m	分岐命令で分岐の判定に使用する値
x	指標レジスタとして使用するレジスタの番号
b	ベースレジスタとして使用するレジスタの番号
d	00h～3Fhで示されるアドレスの変位

(1)　この命令語を実行するコンピュータの1語は16ビットであり，1語長のレジスタを4個（レジスタ番号0～3）と，命令の実行結果によって値が設定される2ビットの条件コードレジスタ（以下，CCという）をもつ。

(2)　実効アドレスは，表2に示す式で算出される。ここで，(x)と(b)は，それぞれxとbで指定されるレジスタに設定されている内容（以下，レジスタの内容という）を示す。

表2　実効アドレスの算出式

x	b	実効アドレス
0	0	d
0	0以外	(b)＋d
0以外	0	(x)＋d
0以外	0以外	(x)＋(b)＋d

(3)　命令コード（一部）を表3に示す。

表3　命令コード（一部）

命令コード	機能	CCの設定
1	実効アドレスに格納されている内容と r で指定されるレジスタの内容の論理和を，r で指定されるレジスタに設定する。	○
2	実効アドレスに格納されている内容と r で指定されるレジスタの内容の論理積を，r で指定されるレジスタに設定する。	○
3	実効アドレスに格納されている内容と r で指定されるレジスタの内容の排他的論理和を，r で指定されるレジスタに設定する。	○
4	m と CC の論理積を求め，結果が 00 でなければ実効アドレスに分岐する。結果が 00 であれば，何もしない。	－

注記　○：論理演算の結果が 0（全てのビットが 0）のときは 2 進数の 10，それ以外のときは2進数の01が設定される。
　　－：実行前の値が保持される。

設問1　レジスタの内容が図2に示す値のとき，次の命令の実効アドレスとして正しい答えを，解答群の中から選べ。

命令：　1983h

レジスタ番号	内容
0	0004h
1	0003h
2	0002h
3	0001h

図2　レジスタの内容

解答群

ア　0001h　　イ　0002h　　ウ　0003h

エ　0004h　　オ　0008h　　カ　000Ah

設問2　次の記述中の［　　］に入れる正しい答えを，解答群の中から選べ。ここで，レジスタと主記憶装置の内容は，図3と図4のとおりとする。

なお，解答は重複して選んでもよい。

レジスタ番号	内容
0	0004h
1	0003h
2	0002h
3	0001h

図3　レジスタの内容

番地	内容
0001h	0001h
0002h	000Fh
0003h	0003h
0004h	0004h
⋮	⋮
0010h	12C0h
0011h	24C0h
0012h	38C2h
0013h	4815h
0014h	4C16h
0015h	18C3h
0016h	28C1h
⋮	⋮

図4　主記憶装置の内容

図 3，4 に示した状態で，主記憶装置に格納されているプログラムを 0010h 番地から実行する。

0011h 番地の命令を実行した直後のレジスタ番号 0 の内容は［ a ］になり，レジスタ番号 1 の内容は［ b ］になる。

0013h 番地の分岐命令では［ c ］。0016h 番地の命令を実行した直後のレジスタ番号 2 の内容は［ d ］になる。

a，b，d に関する解答群

ア　0001h	イ　0002h	ウ　0003h
エ　0004h	オ　0005h	カ　0006h
キ　0007h	ク　0008h	ケ　0009h

c に関する解答群

ア　分岐しない	イ　分岐する

問3　プログラムの並列実行に関する次の記述を読んで，設問1～3に答えよ。

プログラムを並列に実行する方法として，スレッドを使用した並列実行がある。スレッドを使用した並列実行では，プログラムの中で並列実行が可能な部分を抽出して複数の処理に分割し，分割した処理を別々のスレッドとして同時に実行する。

スレッドによる並列実行の例を，図1に示す。プログラムAは，一つのプロセスとして実行され，データ作成，計算処理，結果出力の順に処理が行われる。ここで，計算処理はn個の処理に分割して並列実行が可能であり，分割した処理を異なるスレッドで並列に実行した結果は，スレッドを使わずに計算処理した実行結果と同じであるとする。計算処理では，スレッドの生成と実行を行い，全てのスレッドの終了を同期処理で待つ。各スレッドでは，分割した計算処理（部分計算処理）を行う。

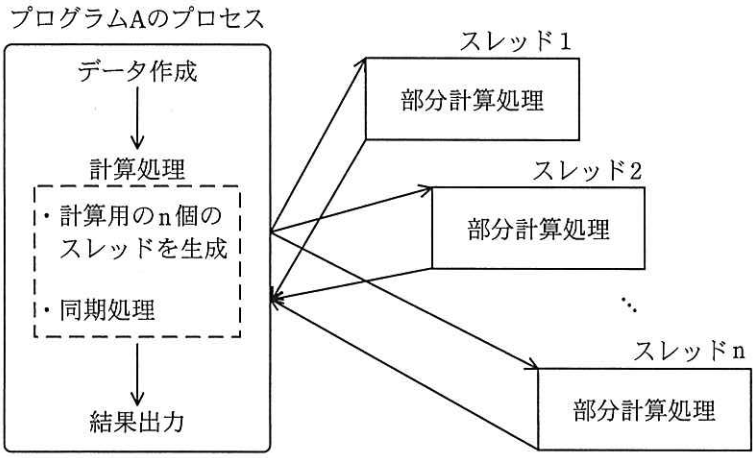


図1　スレッドによる並列実行の例

プログラムAを実行するコンピュータの構成は，性能が同じである複数のCPUと各CPUからアクセス可能な共有メモリで構成されているマルチプロセッサとする。生成された複数のスレッドは，異なるCPUに割り当てられて同時に実行され，各スレッドは共有メモリでデータを共有する。マルチプロセッサにおけるスレッドとCPU，共有メモリの関係を，図2に示す。ここで，データを転送するバスの競合は無いものとする。

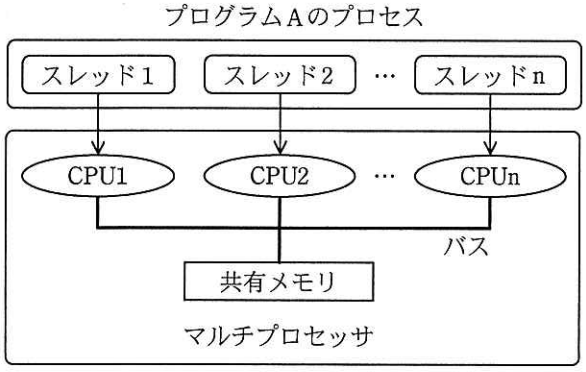


図2　マルチプロセッサにおけるスレッド実行

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。

マルチプロセッサによる並列実行で得られる理想的な高速化率 E は，次の式で求められる。

$$E = \frac{1}{1 - r + \left(\frac{r}{n}\right)}$$

n：CPUの数（$n \geqq 1$）

r：対象とする処理のうち，並列実行が可能な部分の処理時間の割合（$0 \leqq r \leqq 1$）

図1のプログラムAにおいて，データ作成，計算処理，結果出力の処理時間の割合が7：90：3の場合，単一のCPUで実行したときと比べた高速化率を5以上にするには，CPUが最低［　a　］個必要である。ここで，スレッドの生成処理などの並列実行に伴うオーバヘッドは考慮しない。

aに関する解答群

ア　5	イ　6	ウ　7
エ　8	オ　9	カ　10

設問2　次の記述中の［　　］に入れる正しい答えを，解答群の中から選べ。

プログラムの一部を複数の処理に分割して並列に実行するためには，プログラムの中から並列実行が可能な部分を抽出する必要がある。並列実行が可能なループの例を，図3に示す。図3は，i のループに関してループを四つに分割し，分割したそれぞれのループの処理をスレッドとして並列実行する場合である。ここで，配列のデータはスレッド間で共有され，変数 i はスレッドごとに確保されるものとする。

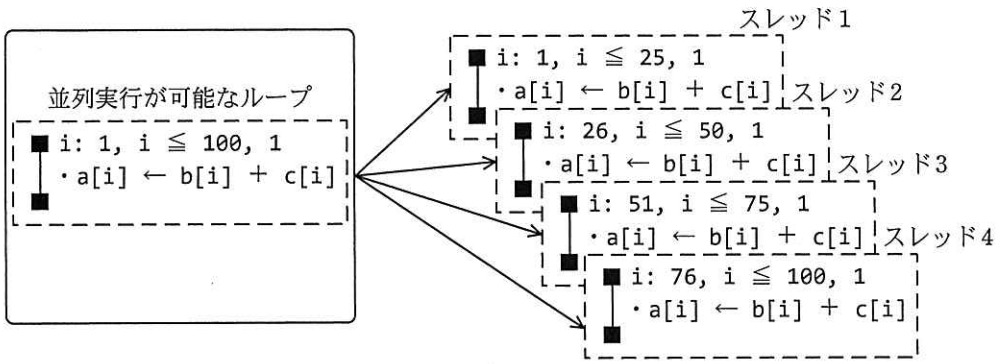


図3　並列実行が可能なループの例

プログラムの中から並列実行が可能な部分を抽出する場合，並列に実行してもデータの更新と参照の順序が変化しないことを保証する必要がある。図4に示すプログラム1～3を，i のループに関して複数のループに分割し，分割したそれぞれのループの処理を並列に実行する場合の並列実行可能性について考える。ここで，配列は十分に大きいものとする。

プログラム1は，ループの中で［　b　］，並列実行できない。プログラム2は，ループの中で［　c　］，並列実行できない。プログラム3は，m の値が不明の場合には並列実行できないが，［　d　］であることが保証されていれば並列実行は可能である。

プログラム 1

```
i: 2, i ≦ n, 1
・a[i] ← a[i−1] + b[i]
```

プログラム 2

```
i: 1, i ≦ n, 1
・a[i] ← a[i+1] + b[i]
```

プログラム 3

```
i: 1, i ≦ n, 1
・a[i] ← a[i+m] + b[i]
```

図4　ループのプログラム1～3

b, cに関する解答群

ア　更新した値が次の繰返しで参照されるので

イ　更新した値が次の繰返しで再び更新されるので

ウ　参照した値が次の繰返しで更新されるので

エ　参照した値が次の繰返しで再び参照されるので

dに関する解答群

ア　m ≧ 0　　イ　m ≧ n　　ウ　m ≦ 0

エ　m ≦ n

設問3　図5に示すプログラム4は，配列 a で更新する要素を示すインデックスの値が配列 ip で間接的に決定される。この配列 a のような更新を含むプログラムは，配列 ip の値によっては並列実行できない場合があるので注意が必要である。プログラム4を，図5のように i のループに関して複数のループに分割し，分割したそれぞれのループの処理をスレッドで並列実行するとき，並列実行可能な ip[i](i＝1, 2, …, 20)の値として適切な答えを，解答群の中から選べ。

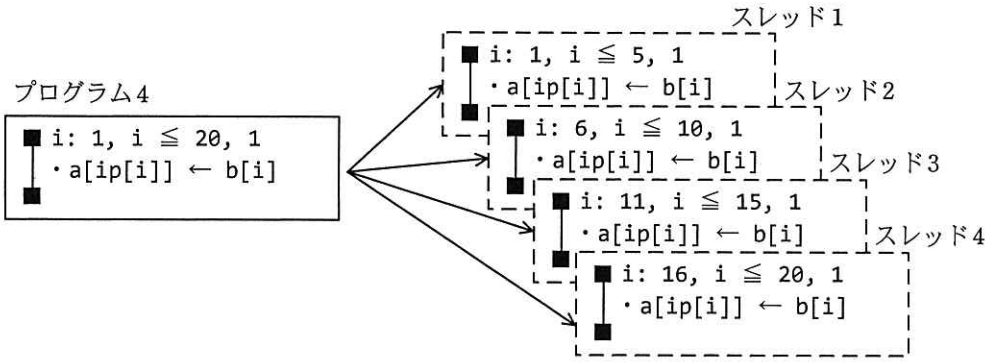


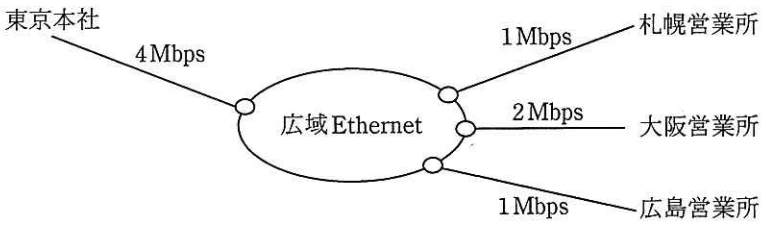
図5　プログラム4の並列実行

解答群

	ip																			
i	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
ア	1	2	3	4	5	1	2	3	4	5	1	2	3	4	5	1	2	3	4	5
イ	2	17	14	10	7	19	16	13	9	6	20	15	12	8	5	18	1	11	3	4
ウ	10	5	8	13	4	8	3	17	2	3	15	13	15	13	1	6	11	18	16	3
エ	20	17	14	10	7	19	16	13	9	6	18	15	12	8	5	17	14	11	7	4

問4　ネットワークにおけるスループットの改善に関する次の記述を読んで，設問 1，2 に答えよ。

X 社は，東京に本社を，札幌，大阪，広島に営業所をもっている。X 社は，広域 Ethernet を利用した企業ネットワークを構築済みである。X 社のネットワーク構成を図 1 に示す。ここで，本社と営業所との間のファイル転送時間に注目し，本社内及び営業所内のファイル転送時間は考慮しない。



bps：ビット／秒，数字は契約した回線速度
○：アクセスポイント

図 1　X 社のネットワーク構成

〔X 社のファイル転送〕

X 社では，各業務システムにおいて，東京本社内にあるファイルサーバ上のファイルを各営業所内の利用者 PC（以下，PC という）から利用している。各営業所からファイルサーバには広域 Ethernet 経由でアクセスしている。①広域 Ethernet はデータリンク層で接続するサービスであり，各営業所内の LAN は広域 Ethernet にアクセスポイントで接続している。

ファイルサーバと PC との間でファイル転送を行う際には，図 2 に示すように，PC からファイルサーバに対して読出し要求を行う。ファイルサーバは，ファイルを固定サイズのブロック（以下，データブロックという）に区切って，PC へ送信することを繰り返す。2 回目以降は，PC からの読出し要求に確認応答が含まれており，ファイルサーバは，PC から確認応答が届くまで次のデータブロックを送信することができない。PC が読出し要求をファイルサーバへ送出し始めてから，ファイルサーバの送信したデータブロックが届き始めるまでを応答時間と呼ぶ。この応答時間は，札幌

又は広島営業所と東京本社との間が 45 ミリ秒，大阪営業所と東京本社との間が 24 ミリ秒であった。ここで，通信障害による再送はないものとする。

このネットワーク構成では，広島営業所内の PC と東京本社のファイルサーバ間で，1M バイトのファイルをブロック長が 4k バイトのデータブロックで転送するには，［ a ］秒掛かる。

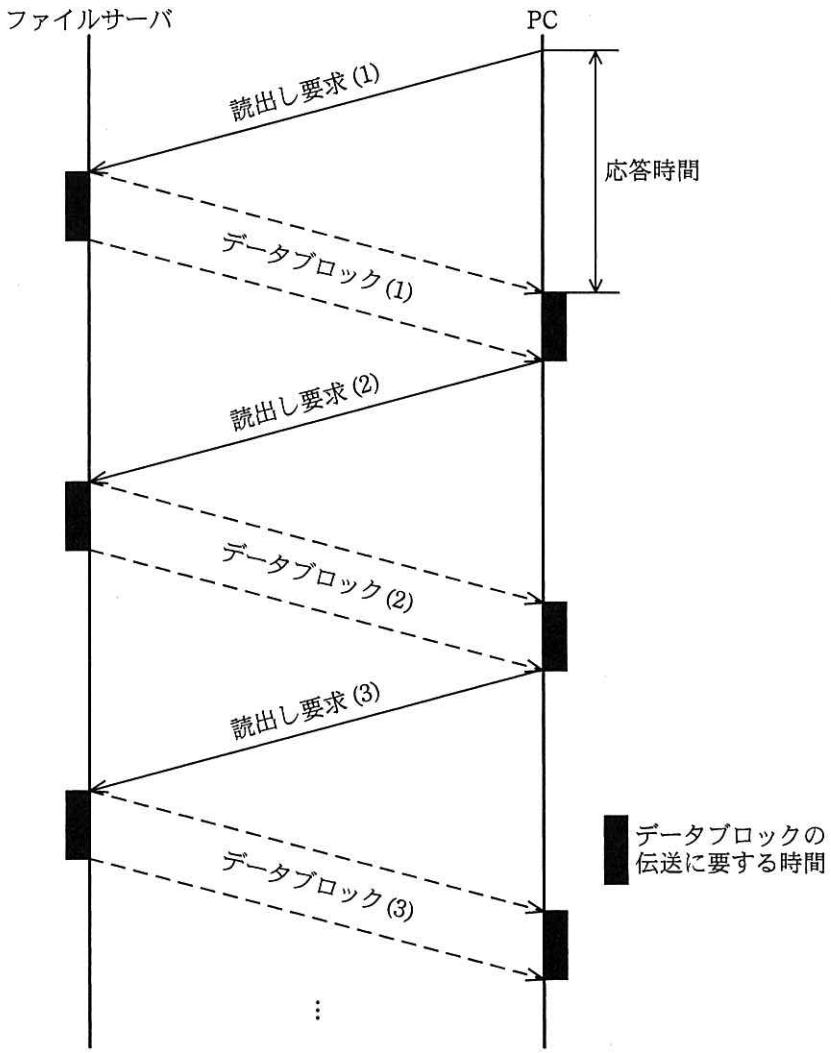
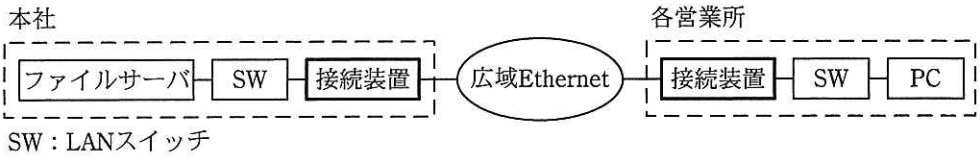


図 2　ファイルサーバと PC 間の通信シーケンス

〔接続装置の検討〕

図 3 に示すように，ファイル転送の高速化のために，広域 Ethernet への接続装置の導入を検討することになった。接続装置は，本社と各営業所にそれぞれ設置され，

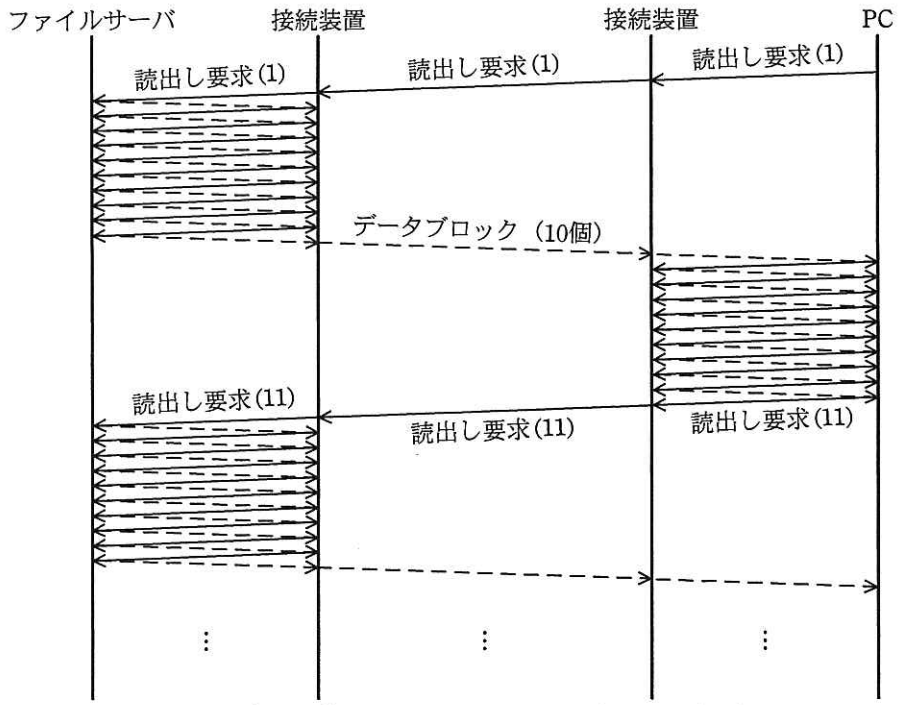
広域 Ethernet を挟んで対向して使用される。



SW：LANスイッチ

図３　本社と営業所間における接続装置の設置

この接続装置は，PC に代わって読出し要求をファイルサーバへ送ることができ，ファイルサーバは連続してデータブロックを送信できるようになる。この接続装置を利用した場合のファイル転送の通信シーケンスを図４に示す。



注記　接続装置は，データブロック 10 個をまとめて送ることができる。

図４　接続装置を利用したファイル転送の通信シーケンス

接続装置を導入すると，札幌営業所内の PC と東京本社のファイルサーバとの間で，1M バイトのファイルをブロック長が 4k バイトのデータブロックで転送するには，［ b ］秒掛かる。ここで，ファイルサーバと接続装置との間及び PC と接続装置との間の，読出し要求とデータブロックの転送時間は考慮しない。

〔アクセス回線速度の検討〕

X社では，業務の都合から，大阪営業所内のPCにおいて1Mバイトのファイルをブロック長が4kバイトのデータブロックで，8秒以内に転送する必要がある。接続装置を導入しない場合，大阪営業所のアクセス回線速度は最低でも，［ c ］Mbpsで広域Ethernetの契約をする必要があることが分かった。

設問1　X社のネットワーク構成において，本文中の下線①の説明文として適切な答えを，解答群の中から選べ。

解答群

ア　広域 Ethernet のアクセスポイントへは，各営業所及び本社からネットワーク層の機能をもつ装置を経由して接続しなければならない。

イ　サーバや PC など，全ての接続機器のネットワーク層のアドレス設定を同一にする必要がある。

ウ　社内の業務システムにおいて，様々な通信プロトコルをネットワーク層で利用することができる。

エ　X社では，TCP/IP以外のプロトコルを使うことができない。

設問2　本文中の［　　］に入れる正しい答えを，解答群の中から選べ。ここで，1Mバイト＝1,000kバイトとし，小数第3位を四捨五入する。

aの解答群

ア　8.00　　イ　10.00　　ウ　11.25　　エ　19.25

bの解答群

ア　3.14　　イ　5.50　　ウ　9.13　　エ　13.40

cの解答群

ア　3　　イ　4　　ウ　8　　エ　10

問5　システム統合に伴うソフトウェア設計に関する次の記述を読んで，設問 1，2 に答えよ。

C 社は大型の電子工具を製造する中堅メーカである。このたび，競争力の強化を図るために，小型の電子工具を製造する D 社を吸収合併することにした。

これに伴い，両社の基幹システムを統合することとなった。早期にシステムを稼働させるために，C 社の基幹システム（以下，C 社システムという）に，D 社の基幹システム（以下，D 社システムという）の機能を追加するというシステム統合の方針を決めた。

なお，C 社は受注生産方式，D 社は見込生産方式と，異なる生産方式を採っているが，D 社の吸収合併後も，C 社製品，D 社製品それぞれの生産方式は変更しない。

〔システム統合の方針〕

(1)　C 社と D 社の両方に存在するシステムについては，C 社システムに，D 社システムの固有機能を追加する。この固有機能の追加では，D 社システムの既存のプログラムをできるだけ再利用する。

(2)　D 社システムを構成する各システム間で行っているデータの連携は，統合後のシステムにおいても，システム統合前の仕様を踏襲する。

(3)　D 社システムだけにある在庫管理システムはそのまま利用し，在庫管理システムと，他システムの連携に関しては，新規にインタフェースを開発する。

〔C 社システムに関する説明〕

図 1 に C 社システムの構成を示す。

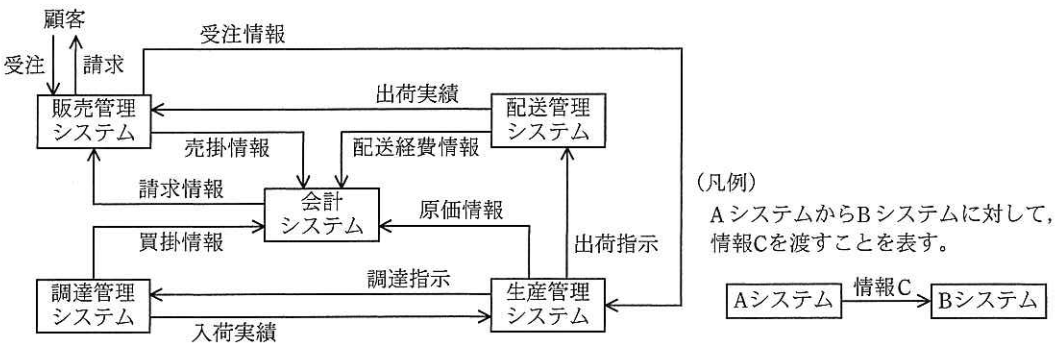


図 1　C 社システムの構成

(1) 販売管理システム

① 顧客からEDI（電子データ交換）で受注する。受注の都度，受注情報を生産管理システムに渡す。

② 出荷実績に基づき，日次で売上計上し，売掛情報を会計システムに渡す。

③ 月初に，顧客に対してEDIで前月の納品分を請求する。

(2) 生産管理システム

① 受注情報に基づき，週次で製品ごとの生産計画を作成する。生産計画作成時点で，製造に必要な部品の調達指示を調達管理システムに渡す。また，生産計画に基づき，製品の出荷指示を配送管理システムに渡す。

② その他，生産工程の作業管理を行い，発生した作業実績（原価情報）を週次で会計システムに渡す。

(3) 調達管理システム

① 調達指示に基づき，部品を発注する。部品の納品（入荷）時に，入荷実績を生産管理システムに渡す。

② 買掛情報を月次で会計システムに渡す。

(4) 配送管理システム

① 出荷指示に基づき，週次で配送計画を作成する。製品の出荷後は，出荷実績を日次で販売管理システムに渡す。

② 配送経費情報を月次で会計システムに渡す。

(5) 会計システム

① 顧客別の請求情報を月次で販売管理システムに渡す。

② 原価計算，売掛・買掛管理，一般会計処理などを行う。

〔D社システムに関する説明〕

図2にD社システムの構成の一部を示す。ここでは，D社システムのうち，システム統合に関係するものだけを記載している。

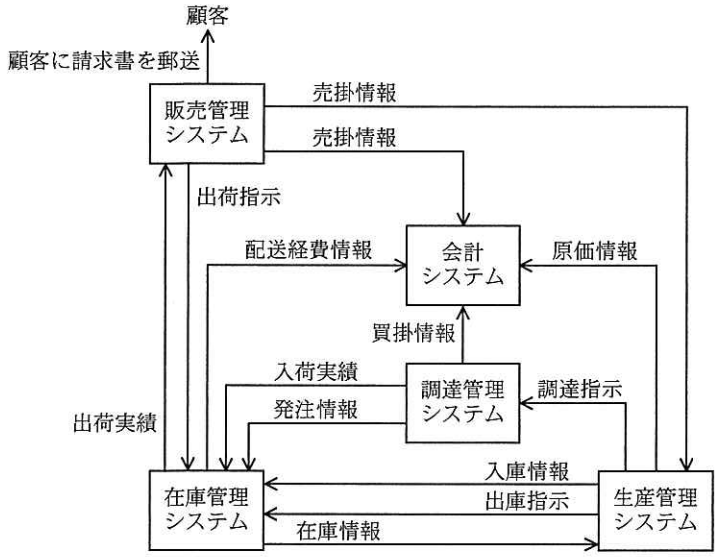


図2　D社システムの構成（一部）

(1)　販売管理システム

①　電話，メール及びファックスで注文を受け付け，D社の社員が端末から受注情報を入力する。この時点で出荷指示を在庫管理システムに渡す。

②　月初に，顧客別に前月納品分の請求書を一括で発行し，顧客に郵送する。

(2)　在庫管理システム

①　製品及び部品の在庫を管理し，入出庫の都度，在庫情報を更新する。

②　出荷指示に基づいて在庫を引き当てる。

③　出荷実績を日次で販売管理システムに渡す。

④　配送経費情報を月次で会計システムに渡す。

(3)　生産管理システム

①　販売管理システムの売掛情報と，在庫管理システムの在庫情報を参照し，製品ごとの生産計画を日次で作成する。

②　生産計画に基づき，部品の出庫指示を在庫管理システムに，調達指示を調達管理システムに渡す。また，製品の計画時点の入庫情報を在庫管理システムに渡す。

③ 製品の完成時の入庫情報を在庫管理システムに渡す。

(4) 調達管理システム

① 調達指示に基づき，部品を発注し，発注情報を在庫管理システムに渡す。

② 部品の納品（入荷）時に，入荷実績を在庫管理システムに渡す。

設問1 表1は，システム統合に当たりC社システムに追加開発が必要となる機能を整理した表である。表中の［　　］に入れる正しい答えを，解答群の中から選べ。

表1 追加開発が必要となる機能

対象のシステム	追加開発が必要となる機能
販売管理システム	・端末から受注情報を入力する機能 ・［　　］
［ a ］システム	・［ b ］

注記 網掛けの部分は表示していない。

aに関する解答群

ア 会計　　イ 生産管理　　ウ 調達管理　　エ 配送管理

bに関する解答群

ア 在庫情報と売掛情報から，製品ごとの生産計画を日次で作成する機能

イ 作業実績（原価情報）から，月次で原価計算をする機能

ウ 受注情報から，月別の販売予測を作成する機能

エ 出荷指示の情報から，週次の配送計画を作成する機能

オ 請求情報から，月別，顧客別に請求書を作成する機能

設問2　システム統合の完了後，C社が経営課題の一つとして抱えていた納期短縮を目的として，C社の製品及び部品も，一定量の在庫を保有してリードタイムを短縮することにした。

これに伴い，システム間の情報の渡し方を見直して，在庫管理システムが他システムから在庫の更新要求を受け取るシステム間連携のインタフェースを新たに開発する。

〔在庫管理に関する説明〕

(1)　在庫は，製品と部品のどちらも，同一体系で一意となる品番コードで実在庫と有効在庫を管理する。

(2)　実在庫は，実際に保有する製品及び部品の在庫の数量である。

(3)　製品の有効在庫は，実在庫に翌日までに完成する製品数を加算し，受注に対して引当て済みの製品数を減算した数量である。

(4)　部品の有効在庫は，実在庫に調達中（未納品）の発注数を加算し，生産計画に対して引当て済みの部品数を減算した数量である。

(5)　受注時は，有効在庫から受注数を引き当てる。

(6)　生産計画時に，翌日までに生産予定の製品数を有効在庫として加算する。また，同時に生産に必要となる部品を有効在庫から引き当てる。

(7)　部品が不足し，調達先へ発注したときは，発注分を有効在庫として加算する。部品が調達先から納品されたら実在庫として加算する。

(8)　部品は，生産工程に投入されたら実在庫を減算する。

(9)　製品の完成時に，製品の実在庫として加算する。

(10)　製品の出荷時に，製品の実在庫を減算する。

表2は，在庫管理システムが他システムから受け取る在庫更新のインタフェース仕様の説明であり，表3は，他システムと在庫管理システムのシステム間連携について，在庫を更新するタイミングとインタフェースの設定内容を整理した表である。表3の［　　］に入れる正しい答えを，解答群の中から選べ。

表２　在庫更新のインタフェース仕様の説明

設定項目	説明
品番コード	更新を行う在庫品の品番コードを設定する
在庫区分	“有効在庫”か“実在庫”のいずれかを設定する
入出庫区分	“入庫”か“出庫”のいずれかを設定する
数量	在庫を増減させる数量

表３　在庫を更新するタイミングとインタフェースの設定内容

在庫を更新するタイミング		インタフェースの設定内容			
		品番コード	在庫区分	入出庫区分	数量
生産管理システム	生産計画時	更新を行う在庫品の品番コード	c		
	生産時				
	製品完成時				
調達管理システム	部品発注時				
	部品納品時		d		
販売管理システム	受注時		e		
配送管理システム	製品出荷時		f		

注記　網掛けの部分は表示していない。

c～fに関する解答群

ア	実在庫	出庫	受注数
イ	実在庫	出庫	出荷数
ウ	実在庫	入庫	生産予定数
エ	実在庫	入庫	納品数
オ	有効在庫	出庫	完成数
カ	有効在庫	出庫	受注数
キ	有効在庫	入庫	生産予定数
ク	有効在庫	入庫	納品数

問6　ファンクションポイント法を用いた工数見積りに関する次の記述を読んで，設問1，2に答えよ。

A社では，名簿データ管理システム開発プロジェクトの計画を策定中であり，アプリケーション設計チームで作成した図1に示す処理フロー図を基に，工数見積りを行う段階に入った。工数見積りの結果がコスト見積り時の根拠となるので，正確性の確保はもとより，プロジェクトにおける各種の制約条件を考慮したものとする必要がある。

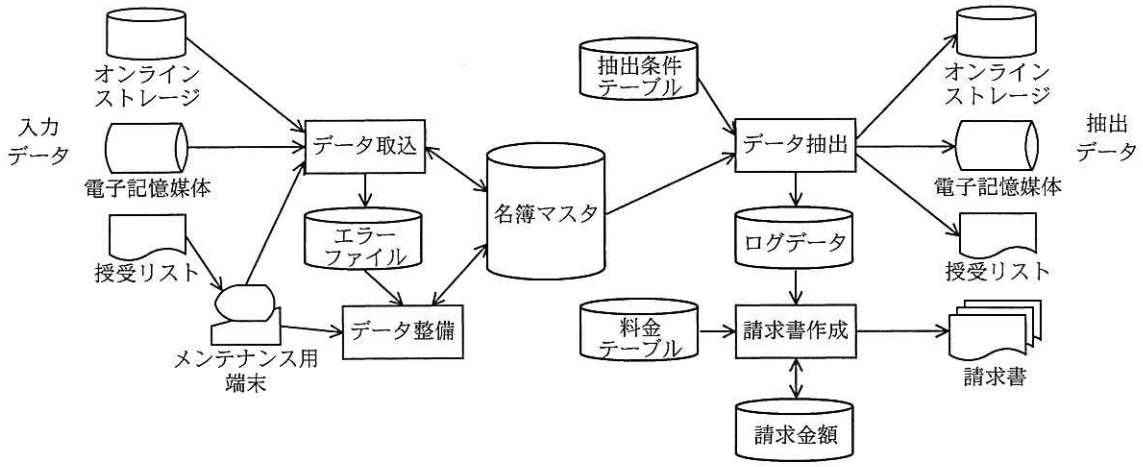


図1　名簿データ管理システムの処理フロー図

〔処理フロー図の説明〕

名簿データ管理システムは，名簿の提供者から受領する名簿データを取り込み，データを整備し，名簿データの利用者からの要求に応じて，データを抽出する。また，利用者から，抽出件数に応じた利用料金を徴収するための請求書を作成する。システムの処理ごとに，その説明を示す。

(1)　データ取込処理では，入力データを基にレコードを編集し，名簿マスタを更新する。入力インタフェースとして，“オンライン接続（オンラインストレージサービスの利用）”，“電子記憶媒体授受”及び“授受リストを基にしたメンテナンス用端末からの入力”の3種類が用意されている。名簿マスタの更新ができなかったデータは，エラーファイルに出力する。

(2) データ整備処理では，名簿マスタ更新時のエラーファイルを基に，メンテナンス用端末を使用して，レコードを編集し，名簿マスタを更新する。

(3) データ抽出処理では，名簿マスタ及び抽出条件テーブルを参照し，目的のレコードを抽出して出力する。出力インタフェースとして，“オンライン接続（オンラインストレージサービスの利用）”，“電子記憶媒体授受”及び“授受リスト”の3種類が用意されている。また，利用者ごとのデータ抽出処理の実行と抽出件数に関するログデータを出力する。

(4) 請求書作成処理では，データ抽出処理で出力されたログデータ及び使用した出力インタフェースとデータ抽出件数ごとに設定された料金テーブルを参照して，請求金額を算出する。算出された請求金額を，請求レコード用フォーマットに編集して，請求書を出力する。

〔工数見積方法の説明〕

(1) 工数見積りには，A社でアレンジした簡易ファンクションポイント法を用いる。簡易ファンクションポイント法は，ソフトウェアがもつ機能及び入出力インタフェースの数を基に，そのソフトウェアの開発規模を推定する手法である。図1で示す各処理の内容を，機能及び入出力インタフェースごとに分類し，それぞれをファンクションとして定義する。図1の処理フロー図から見積対象の処理ごとに，各ファンクションの個数を求める。

(2) 求められた各ファンクションの個数を基に，処理ごとの開発規模を求める。名簿データ管理システムのファンクションとそれに対応する開発規模を表1に示す。

表1　ファンクション一覧

項番	ファンクション名	ファンクションに対応する開発規模（kステップ）
1	ファイル読込	0.1
2	ファイル書込	0.3
3	帳票作成	0.6
4	端末画面	1.0
5	レコード編集	0.5

注記　開発規模の単位　1kステップ＝1,000ステップ

(3) 見積対象処理ごとに算出された開発規模と表 2 に示す開発生産性の計画値から，工数を算出する。

表 2　開発生産性の計画値

開発生産性計画値	0.5（k ステップ／人月）

設問 1　プロジェクト管理グループに属する社員 B は，プロジェクトマネージャの指導の下，工数見積方法に従い，図 1 の処理フロー図に示された各処理の開発に要する工数を算出して，表 3 にまとめた。表 3 中の［　　］に入れる正しい答えを，解答群の中から選べ。

表 3　処理ごとのファンクション数と開発規模一覧

ファンクション名 / 処理名	ファイル読込	ファイル書込	帳票作成	端末画面	レコード編集	処理別開発規模合計	処理別工数合計（人月）
データ取込	3	2	0	1	1		4.8
	0.3	0.6	0.0	1.0	0.5	2.4	
データ整備	2	1	0	1	1		b
	0.2	0.3	0.0	1.0	0.5	2.0	
データ抽出	2	a	1	0	0		
	0.2		0.6	0.0	0.0		
請求書作成	3	1	1	0	1		3.4
	0.3	0.3	0.6	0.0	0.5	1.7	
ファンクション別開発規模合計	1.0		1.2	2.0	1.5		

注記　表中の処理ごとに，上段がファンクション数，下段がファンクション数を基に算出された開発規模（単位：k ステップ）を表す。網掛けの部分は表示していない。

a に関する解答群

ア　1　　イ　2　　ウ　3　　エ　4

b に関する解答群

ア　1.0　　イ　2.0　　ウ　2.5　　エ　4.0

設問2　設問1で求めた工数を基に算出した開発コストは，プロジェクトで想定している開発コストの計画値を上回った。社員 B はプロジェクトマネージャから，開発工数を削減する方法の検討を行うよう指示を受けた。

社員 B は，処理別工程ごとの工数を表 4 にまとめた。表 3 及び表 4 を検証して，工数削減方法の検討を進めた。検討を行った作業に関する次の記述中の［　　］に入れる適切な答えを，解答群の中から選べ。ここで，工程ごとの工数は，処理ごとの工数に工程比率を乗じて算出したものである。

表 4　処理別工程ごとの工数一覧

工程名	外部設計	詳細設計	製造	総合試験	運用試験	処理別工数合計（人月）
工程比率 / 処理名	15%	25%	40%	15%	5%	
データ取込	0.72	1.20	1.92	0.72	0.24	4.8
データ整備	0.60		1.60	0.60	0.20	
データ抽出	0.51	0.85	1.36		0.17	
請求書作成	0.51	0.85	1.36	0.51	0.17	3.4

注記　網掛け部分は表示していない。

全ての処理に利用され，ファンクション別開発規模合計が最も大きいファンクションに対して対策を講じることが，工数削減の効果が最も大きいと想定し，表3から対象ファンクションを［ c ］と判断した。

工数削減施策として，“対象ファンクションを共通部品として作成し，各処理に適用することによって，工数を削減する施策”を検討した。共通部品化によって，全工程の工数合計の 80%に相当する［ d ］工程における工数の一部を削減することができる。

例えば，データ抽出処理では，［ d ］工程の工数のうち［ e ］人月が［ c ］の工数なので，本施策によって工数削減を見込むことができる。

cに関する解答群

ア　端末画面　　イ　帳票作成　　ウ　ファイル書込

エ　ファイル読込　　オ　レコード編集

dに関する解答群

ア　外部設計

イ　外部設計，詳細設計

ウ　外部設計，詳細設計，製造

エ　外部設計，詳細設計，製造，総合試験

オ　外部設計，詳細設計，製造，総合試験，運用試験

eに関する解答群

ア　1.088　　イ　1.44　　ウ　2.176　　エ　2.72

問7　システム移行の作業計画に関する次の記述を読んで，設問1～3に答えよ。

A社の物流部では，物流費管理システムのバージョンアップを予定している。物流費管理システムは，ソフトウェアパッケージ（以下，パッケージソフトという）とA社用に開発されたアプリケーションソフトウェア（以下，開発ソフトという）から構成されている。このたび，パッケージソフト及び開発ソフトのバージョンアップ対応版が準備できたので，物流費管理システムのバージョンアップに関する作業計画を策定することになった。表1は，バージョンアップに必要な作業の一覧である。

表1　バージョンアップに必要な作業の一覧

作業	作業内容	作業時間（時間）	必要要員数（人）	先行作業
No1	物流費管理システムのサービス停止	1	1	—
No2	物流費管理システムのバックアップ	1	4	No1
No3	パッケージソフトデータの抽出	1	3	No2
No4	開発ソフトデータの抽出	1	3	No2
No5	パッケージソフトデータの変換	1	3	No3
No6	開発ソフトデータの変換	1	3	No4
No7	パッケージソフトの新バージョンの導入	2	4	No3
No8	開発ソフトの新バージョンの導入	1	2	No4
No9	パッケージソフトの新バージョンと開発ソフトの新バージョンの単体動作及び連結動作の確認	3	6	No7，No8
No10	変換後のパッケージソフトデータのロード	2	2	No5，No9
No11	変換後の開発ソフトデータのロード	1	2	No6，No9
No12	変換後のパッケージソフトデータ及び開発ソフトデータの確認	1	3	No10，No11
No13	物流費管理システム全体の動作確認及び判定	3	6	No9，No12

設問1　バージョンアップのスケジュール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。

物流部では，表1を基にアローダイアグラムを作成することにした。図1は，作成途中のアローダイアグラムである。

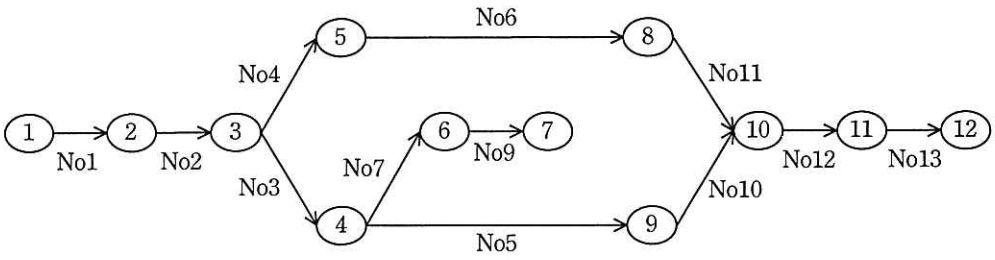


図1　作成途中のアローダイアグラム

図 1 の作業途中のアローダイアグラムに不足しているのは，ノード［　a　］への作業 No8 を表す矢線と，ノード 7 から［　b　］へのダミー作業を表す矢線である。

バージョンアップに必要な一連の作業の最短所要時間を計算すると，［　c　］時間となる。

a に関する解答群

ア　5 から 6　　イ　5 から 7　　ウ　6 から 8　　エ　6 から 9

b に関する解答群

ア　8　　イ　9　　ウ　10

エ　8 と 9　　オ　8 と 10　　カ　9 と 10

c に関する解答群

ア　9　　イ　10　　ウ　13　　エ　14　　オ　19

設問 2　物流部では，バージョンアップ作業に必要となる要員計画を策定することにした。要員計画に関する次の記述及び表中の［　　］に入れる適切な答えを，解答群の中から選べ。

物流部では，6 時に作業 No1 を開始し，その後の作業は先行作業が全て完了した時点ですぐに開始できるように作業を配置した。表 2 に，作成した要員計画表の一部を示す。表 2 の“作業”の欄は，該当時刻に開始する作業を表している。

同じ開始時刻に複数の作業が表記されている場合は，それらの作業を並行して行うことを意味している。“必要人数”の欄は，その時間帯に必要となる要員数であり，同じ時間帯に複数の作業を行う場合は，それぞれの作業の要員数の合計である。

表2　要員計画表の一部

<table>
<tr><th>開始時刻</th><th colspan="4">作業</th><th>必要人数</th></tr>
<tr><td>6時</td><td>No1</td><td></td><td></td><td></td><td>1</td></tr>
<tr><td>7時</td><td>No2</td><td></td><td></td><td></td><td>4</td></tr>
<tr><td>8時</td><td>No3</td><td></td><td>No4</td><td></td><td>6</td></tr>
<tr><td>9時</td><td>No5</td><td rowspan="2">No7 [1)]</td><td>No6</td><td>No8</td><td>12</td></tr>
<tr><td>10時</td><td></td><td></td><td></td><td>4</td></tr>
<tr><td>11時</td><td rowspan="6"></td><td rowspan="6"></td><td rowspan="6"></td><td rowspan="6"></td><td>6</td></tr>
<tr><td>12時</td><td>6</td></tr>
<tr><td>13時</td><td rowspan="3">d</td></tr>
<tr><td>14時</td></tr>
<tr><td>15時</td></tr>
<tr><td>16時</td><td>3</td></tr>
<tr><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td></td><td></td><td></td><td></td><td></td><td></td></tr>
</table>

注記　網掛けの部分は表示していない。

注 [1)]　No7 のように作業が 1 時間よりも長く掛かる場合は，途中の罫線（けい）を省略している。

9～10 時の必要人数は 12 人であるが，［　e　］に開始するなど作業の時間帯を工夫することによって，全体の作業完了時刻を遅らせることなく，その時間帯の必要人数を減らすことができる。

物流部では，全体の作業完了時刻を遅らせることなく，かつ，時間帯ごとの必要人数を最少とするように，作業 No5，No6 及び No11 の開始時刻を見直した。その結果，一連の作業を行うのに必要な最少人数は［　f　］人となった。ここで，要員は全ての作業が行えるものとする。

dに関する解答群

ア

6
2
2

イ

6
2
4

ウ

6
4
2

エ

8
2
2

オ

8
4
2

eに関する解答群

ア　No5を10時　　イ　No5を14時　　ウ　No6を7時　　エ　No6を8時

fに関する解答群

ア　4　　イ　6　　ウ　7　　エ　8　　オ　9　　カ　10

設問3　物流部では，全体の作業完了時刻を早めたいと考えている。全体の作業完了時刻を30分早められる記述として正しい答えを，解答群の中から二つ選べ。

解答群

ア　No2とNo9の作業時間を15分ずつ短縮する。

イ　No3とNo4の作業時間を15分ずつ短縮する。

ウ　No5とNo6の作業時間を15分ずつ短縮する。

エ　No10とNo11の作業時間を15分ずつ短縮する。

オ　No12とNo13の作業時間を15分ずつ短縮する。

次の**問8**は必須問題です。必ず解答してください。

問8　次のアルゴリズムの説明を読んで，設問1～3に答えよ。

セルを1列に連続して並べた領域がある。この領域中のセルについて，割当てと解放の処理を行う。

各セルには，セル位置を指定するための連続する整数が対応している。領域のセル数や，対応する整数の範囲には，特に制限がない。

各セルは，“空き”又は“割当済み”のいずれかの状態にある。現在，領域中のどのセルが“空き”の状態にあるかという情報を，空きリストとして保持している。

関数 Alloc(始点，終点) は，引数で指定した始点から終点までの連続した“空き”セルを“割当済み”として，空きリストから取り除く。関数 Free(始点，終点) は，引数で指定した始点から終点までの連続した“割当済み”セルを“空き”として，空きリストに戻す。

〔空きリストの説明〕

空きリストの形式を，次に示す。

$\{\{始点_1，終点_1\}，\{始点_2，終点_2\}，\cdots，\{始点_N，終点_N\}\}$

$\{始点_i，終点_i\}$（$始点_i \leqq 終点_i$）は，一つの連続した“空き”セルの先頭位置と終端位置の組（以下，組という）で，$始点_1 < 始点_2 < \cdots < 始点_N$ である。

割当て・解放の処理と空きリストの状態の例を，次の (1) ～ (3) に示す。ここで，セル □ 中の数字は，セル位置を表す。また，□ は“空き”を，▒ は“割当済み”を，それぞれ表す。

(1)　領域の初期状態は，全セルが空いている。空きリストは {{ -∞，+∞ }} で表す。

…	0	1	2	3	4	5	6	7	8	9	…

初期状態の空きリスト：{{ -∞，+∞ }}

ここで，記号“-∞”は，領域中のどのセル位置の値よりも小さい整数を表し，記号“+∞”は，領域中のどのセル位置の値よりも大きい整数を表すものとする。また，セル位置 -∞ ～ +∞ のうち，領域外の部分には“空き”セルが並んでいるもの

とする。

(2) 関数 Alloc で“割当済み”としたセルは，空きリストから取り除く。例えば，(1) の初期状態から，Alloc(1, 2) と Alloc(6, 8) を実行すると，次のようになる。

…	0	1	2	3	4	5	6	7	8	9	…

実行後の空きリスト：{ { -∞, 0 }, { 3, 5 }, { 9, +∞ } }

実行後，空きリスト中の組の個数は 3 となる。

(3) 関数 Free で解放したセルは，空きリストに戻す。例えば，(2) の実行後の状態から，Free(6, 7) を実行すると，次のようになる。

…	0	1	2	3	4	5	6	7	8	9	…

実行後の空きリスト：{ { -∞, 0 }, { 3, 7 }, { 9, +∞ } }

実行後，解放された“空き”セルの組 { 6, 7 } は，実行前の“空き”セルの組 { 3, 5 } とつながって一つの連続した“空き”セルの組 { 3, 7 } となるので，空きリスト中の組の個数は 3 となる。

〔関数 Alloc の説明〕

関数 Alloc(始点$_p$, 終点$_p$) の処理手順は，次のとおりである。

なお，引数の値は，$-\infty <$ 始点$_p \leqq$ 終点$_p < +\infty$ を満たしているものとする。

(1) 空きリスト中に，始点$_i \leqq$ 始点$_p$ かつ 終点$_p \leqq$ 終点$_i$ を満たす組 { 始点$_i$, 終点$_i$ } が存在すれば (2) へ進む。存在しなければ，“一部又は全体が割当済み”を表示して，処理を終了する。

(2) 割当てが可能であるので，表 1 に従って，引数の状況に対応した空きリストの更新処理を実行して，処理を終了する。

表1　関数 Alloc の空きリスト更新処理

引数の状況	空きリストの更新処理
始点$_i$ ＝ 始点$_p$ かつ 終点$_p$ ＝ 終点$_i$	組 { 始点$_i$，終点$_i$ } を取り除く。
始点$_i$ ＝ 始点$_p$ かつ 終点$_p$ ＜ 終点$_i$	組 { 始点$_i$，終点$_i$ } を組　　で置き換える。
始点$_i$ ＜ 始点$_p$ かつ 終点$_p$ ＝ 終点$_i$	組 { 始点$_i$，終点$_i$ } を組　　で置き換える。
始点$_i$ ＜ 始点$_p$ かつ 終点$_p$ ＜ 終点$_i$	組 { 始点$_i$，終点$_i$ } を二つの組 { 始点$_i$，始点$_p$ −1 } と { 終点$_p$ +1，終点$_i$ } で置き換える。

注記　網掛けの部分は表示していない。

〔関数 Free の説明〕

関数 Free(始点$_p$，終点$_p$) の処理手順は，次のとおりである。

なお，引数の値は，$-\infty <$ 始点$_p \leqq$ 終点$_p < +\infty$ を満たしているものとする。

(1)　空きリスト中に，終点$_i <$ 始点$_p$ かつ 終点$_p <$ 始点$_{i+1}$ を満たす連続する二つの組 { 始点$_i$，終点$_i$ } と { 始点$_{i+1}$，終点$_{i+1}$ } が存在すれば (2) へ進む。存在しなければ，“一部又は全体が割当済みでない”を表示して，処理を終了する。

(2)　解放が可能であるので，表 2 に従って，引数の状況に対応した空きリストの更新処理を実行して，処理を終了する。

表2　関数 Free の空きリスト更新処理

引数の状況	空きリストの更新処理
終点$_i$ ＝ 始点$_p$ −1 かつ 終点$_p$ +1 ＝ 始点$_{i+1}$	二つの組 { 始点$_i$，終点$_i$ } と { 始点$_{i+1}$，終点$_{i+1}$ } を一つの組 [a] で置き換える。
終点$_i$ ＝ 始点$_p$ −1 かつ 終点$_p$ +1 ＜ 始点$_{i+1}$	組 { 始点$_i$，終点$_i$ } を組 { 始点$_i$，終点$_p$ } で置き換える。
終点$_i$ ＜ 始点$_p$ −1 かつ 終点$_p$ +1 ＝ 始点$_{i+1}$	組 { 始点$_{i+1}$，終点$_{i+1}$ } を組 { 始点$_p$，終点$_{i+1}$ } で置き換える。
終点$_i$ ＜ 始点$_p$ −1 かつ 終点$_p$ +1 ＜ 始点$_{i+1}$	組 { 始点$_i$，終点$_i$ } の直後に組 [b] を挿入する。

設問1　本文中の［　　］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　{始点$_i$，終点$_{i+1}$}　　イ　{始点$_i$，終点$_p$}

ウ　{始点$_p$，終点$_{i+1}$}　　エ　{始点$_p$，終点$_p$}

設問2　次のプログラム中の［　　］に入れる正しい答えを，解答群の中から選べ。

関数Allocの説明に基づいて，プログラムを作成した。

空きリスト中の現在の組の個数は大域整数型変数 N に格納されている。空きリスト{{始点$_1$，終点$_1$}，{始点$_2$，終点$_2$}，…，{始点$_N$，終点$_N$}}については，始点$_i$（i：1，2，…，N）の値は大域整数型配列 **始点** の要素 **始点**[i] に，終点$_i$（i：1，2，…，N）の値は大域整数型配列 **終点** の要素 **終点**[i] に，それぞれ格納されている。これらの配列は，十分に大きいものとする。

〔プログラム〕

○関数: Alloc(整数型: 始点P, 整数型: 終点P)
○整数型: I, L

```
• I ← 1
■ 終点P ＞ 終点[I]
  • I ← I+1
■
▲ 始点[I] ≦ 始点P
  ▲ (始点[I] ＝ 始点P) and (終点P ＝ 終点[I])
    ■ L: I+1, L ≦ N, 1
      • 始点[L−1] ← 始点[L]
      • 終点[L−1] ← 終点[L]
    ■
    • N ← N−1
  ▲ (始点[I] ＝ 始点P) and (終点P ＜ 終点[I])
    • [      c      ]
  ▲ (始点[I] ＜ 始点P) and (終点P ＝ 終点[I])
    • [      d      ]
  ▲ (始点[I] ＜ 始点P) and (終点P ＜ 終点[I])
    ■ L: [      e      ]
      • 始点[L+1] ← 始点[L]
      • 終点[L+1] ← 終点[L]
    ■
    • 始点[I+1] ← 終点P+1
    • 終点[I+1] ← 終点[I]
    • 終点[I] ← 始点P−1
    • N ← N+1
  • print("一部又は全体が割当済み")   /* "  " 内の文字列を表示 */
```

c, dに関する解答群

ア　始点[I] ← 始点P−1　　イ　始点[I] ← 終点P+1
ウ　終点[I] ← 始点P−1　　エ　終点[I] ← 終点P+1

eに関する解答群

ア　I+1, L ＜ N, 1　　イ　I+1, L ≦ N, 1
ウ　N, L ≧ I+1, −1　　エ　N, L ＞ I+1, −1

設問3　次の記述中の［　　］に入れる適切な答えを，解答群の中から選べ。

このアルゴリズムでは，空きリスト｛｛始点$_1$，終点$_1$｝，｛始点$_2$，終点$_2$｝，…，｛始点$_N$，終点$_N$｝｝の始点$_1$に値 -∞ を，終点$_N$に値 +∞ をそれぞれ設定している。このような設定をすることの利点の一つに，［ f ］という特徴が挙げられる。

また，このアルゴリズムでは，空きリスト中の組の個数が変化する。領域中のセル数が E 個であるとする。このとき，空きリスト中の組の個数は，最大で［ g ］となる。また，E 個の全てのセルが“割当済み”となったとき，空きリスト中の組の個数は，［ h ］となる。ここで，整数同士の除算では，商の小数点以下を切り捨てる。

fに関する解答群

ア　空きリストが空（組の個数が 0）にならない

イ　関数 Free の実行時に空きリスト中の組の個数が 2 以上であることが保証される

ウ　始点$_1$又は終点$_N$の値が変わらない限り領域中に“空き”セルが残っている

エ　領域中の一つの連続した“空き”セルが幾ら長くても一つの組で表せる

g，hに関する解答群

ア　1	イ　2	ウ　E÷2＋1
エ　(E＋1)÷2＋1	オ　E＋1	カ　E＋2

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

なお，2問以上マークした場合には，はじめの1問について採点します。

問9　次のCプログラムの説明及びプログラムを読んで，設問1，2に答えよ。

C

〔プログラムの説明〕

関数 format_text は，印字したときに単語が行末で切れないようにテキストを編集してファイルに出力するプログラムである。

(1) テキストに含まれる文字は，次のものである。

① 英字　A～Z，a～z

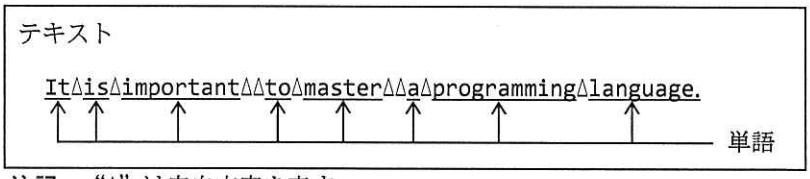
② 数字　0～9

③ 記号　!"#%&'()*+,-./:;<=>?[]^_{|}~

④ 空白文字

⑤ 改行文字

(2) 単語は，空白文字及び改行文字を含まない文字列であり，空白文字又は改行文字で区切られている。単語の文字数は20以下とする。図1に，テキストに含まれる単語の例を示す。



注記　"△"は空白文字を表す。

図1　テキストに含まれる単語の例

(3) 改行文字を除き，1行分として印字できる文字数（以下，最大文字数という）をプログラムの引数で与えるが，その値は40以上とする。単語の途中で最大文字数を超える場合は，単語の直前に改行文字を出力し，その単語が次の行の先頭に印字されるようにする。

(4) テキスト中の空白文字及び改行文字は，そのまま出力する。ただし，連続する

空白文字を出力している途中で最大文字数を超える場合は改行文字を出力し，最大文字数を超える分は次の行の先頭から印字されるようにする。

(5) 関数 format_text 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。

in_file	編集前のテキスト（以下，入力テキストという）が格納されているファイル名
out_file	編集後のテキスト（以下，出力テキストという）を格納するファイル名
width	最大文字数

(6) プログラムを実行したときは，図2に示すようになる。

（入力テキスト）

```
TheΔΔΔΔInformationΔΔΔTechnologyΔΔΔEngineers↓
ΔΔΔExaminationΔΔΔwasΔΔfirstΔΔadministeredΔΔinΔΔ1969.ΔInΔ1970,↓
ΔitΔbecameΔaΔnationalΔexamination.Δ↓
ΔΔΔΔΔΔΔSinceΔitsΔcommencement,ΔtheΔexaminationΔhasΔplayedΔΔΔ↓
ΔΔΔΔanΔimportantΔroleΔΔΔinΔtheΔdevelopmentΔofΔITΔΔΔΔengineers.Δ↓
```

（出力テキスト）

```
TheΔΔΔΔInformationΔΔΔTechnologyΔΔΔEngineers↓
ΔΔΔExaminationΔΔΔwasΔΔfirstΔΔadministeredΔΔinΔΔ↓
1969.ΔInΔ1970,↓
ΔitΔbecameΔaΔnationalΔexamination.Δ↓
ΔΔΔΔΔΔΔSinceΔitsΔcommencement,ΔtheΔexaminationΔhas↓
ΔplayedΔΔΔ↓
ΔΔΔΔanΔimportantΔroleΔΔΔinΔtheΔdevelopmentΔofΔITΔΔ↓
ΔΔengineers.Δ↓
---------+---------+---------+---------+---------+
1        10        20        30        40        50
印字したときの文字位置
```

注記 “Δ”は空白文字を，“↓”は改行文字を表す。

図2 最大文字数を50とした場合の例

〔プログラム〕

(行番号)

```
 1 #include <stdio.h>

 2 #define WLEN_MAX 20  /* 単語の最大長 */

 3 void format_text(char *, char *, int);

 4 void format_text(char *in_file, char *out_file, int width) {
 5     FILE *ifp, *ofp;
 6     int  ch,          /* 入力した文字 */
 7          lpos = 0,    /* 出力処理をしている行での出力済み文字数 */
 8          sp = 0;      /* strに格納されている文字数 */
 9     char str[WLEN_MAX + 2];     /* 単語を含む出力用文字列 */

10     ifp = fopen(in_file, "r");
11     ofp = fopen(out_file, "w");

12     while ((ch = fgetc(ifp)) != EOF) {
13         if (ch == '\n') {
14             str[sp++] = ch;
15             str[sp]   = '\0';
16             fputs(str, ofp);
17             [   a   ];
18             sp = 0;
19         } else if (ch == ' ') {
20             lpos += sp;
21             if (lpos >= width) {
22                 str[sp++] = [   b   ];
23                 lpos = 0;
24             }
25             str[sp] = '\0';
26             fputs(str, ofp);
27             fputc(ch, ofp);
28             lpos++;
29             [   c   ];
30         } else {
31             if ([   d   ]) {
32                 fputc('\n', ofp);
33                 lpos = 0;
34             }
35             str[sp++] = ch;
36         }
37     }
38     str[sp] = '\0';
39     fputs(str, ofp);

40     fclose(ifp);
41     fclose(ofp);
42 }
```

設問1　プログラム中の［　　］に入れる正しい答えを，解答群の中から選べ。

aに関する解答群

ア　lpos = 0　　イ　lpos = sp　　ウ　lpos = width

エ　lpos++　　オ　lpos += sp

bに関する解答群

ア　'\0'　　イ　'\n'　　ウ　' '　　エ　ch

cに関する解答群

ア　sp = 0　　イ　sp = lpos　　ウ　sp++

エ　str[lpos] = ch　　オ　str[sp++] = ch

dに関する解答群

ア　lpos > 0　　イ　lpos >= width

ウ　lpos > sp　　エ　sp > 0

オ　(lpos + sp) > 0　　カ　(lpos + sp) >= width

設問2　次の記述中の［　　］に入れる正しい答えを，解答群の中から選べ。

テキストの編集処理のうち，行頭の空白文字に関する処理を変更することになった。これに合わせて関数 format_text に，処理を追加する。ここで，プログラム中の［ a ］～［ d ］には正しい答えが入っているものとする。

(1)　テキストの編集処理の変更内容は，次のとおりである。

①　入力テキスト中の1文字以上の連続する空白文字列で，印字したときに行頭に来る部分は出力しない。

②　最初に出力する単語の場合，又は行の先頭に来る単語の場合で直前に出力した単語の最後の文字が“.”であったときは，単語の直前に空白文字を一つ出力する。

(2)　処理を変更したプログラムを実行したときは，図3に示すようになる。

（入力テキスト）

```
TheΔΔΔΔInformationΔΔΔTechnologyΔΔΔEngineers↓
ΔΔΔExaminationΔΔΔwasΔΔfirstΔΔadministeredΔΔinΔΔ1969.ΔInΔ1970,↓
ΔitΔbecameΔaΔnationalΔexamination.Δ↓
ΔΔΔΔΔΔΔSinceΔitsΔcommencement,ΔtheΔexaminationΔhasΔplayedΔΔΔ↓
ΔΔΔΔanΔimportantΔroleΔΔΔinΔtheΔdevelopmentΔofΔITΔΔΔΔengineers.Δ↓
```

（出力テキスト）

```
ΔTheΔΔΔΔInformationΔΔΔTechnologyΔΔΔEngineers↓
ExaminationΔΔΔwasΔΔfirstΔΔadministeredΔΔinΔΔ1969.Δ↓
ΔInΔ1970,↓
itΔbecameΔaΔnationalΔexamination.Δ↓
ΔSinceΔitsΔcommencement,ΔtheΔexaminationΔhasΔ↓
playedΔΔΔ↓
anΔimportantΔroleΔΔΔinΔtheΔdevelopmentΔofΔITΔΔΔΔ↓
engineers.Δ↓
---------+---------+---------+---------+---------+
1        10        20        30        40        50
印字したときの文字位置
```

注記　“Δ”は空白文字を，“↓”は改行文字を表す。

図３　処理を変更し，最大文字数を 50 とした場合の例

(3)　処理の変更に対応するために，プログラムを表１のとおりに変更する。

表１　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	`int lch = '.';`
行番号 26 と 27 の間に追加	`if (  e  ) {`
行番号 28 と 29 の間に追加	`}`

行番号 34 と 35 の間に追加	`if ((lpos == 0) && (sp == 0) && (lch == '.')) {`
	`  fputc(' ', ofp);`
	`  lpos++;`
	`}`
	`  f  ;`

eに関する解答群

ア `lpos == 0`	イ `lpos == sp`	ウ `lpos != 0`
エ `lpos != sp`	オ `sp == 0`	カ `sp != 0`

fに関する解答群

ア `lch = ' '`	イ `lch = '.'`	ウ `lch = ch`
エ `str[sp++] = ' '`	オ `str[sp++] = lch`	

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〕

X社では，従業員の英語力向上のために，定期的に英語の検定テストを実施している。従業員は誰でも希望すれば受検することができ，直近の5回分の検定結果が得点マスタファイルに記録される。このプログラムは，1回分の検定結果が記録された採点ファイルを読み込み，検定結果を得点マスタファイルに反映する。

(1)　得点マスタファイルは，図1に示すレコード様式の順ファイルで，従業員の得点履歴を管理する。

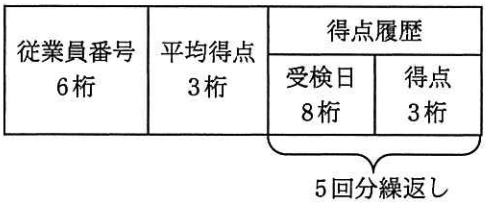


図1　得点マスタファイルのレコード様式

①　全ての従業員に対するレコードが，従業員番号の昇順に格納されている。

②　平均得点には，得点履歴に記録された得点の平均点が小数点以下切捨てで格納される。

③　得点履歴には，直近5回分の受検日と得点が，受検日の降順に格納される。受検回数が5回に満たない場合，残りの得点履歴の受検日及び得点にはゼロが格納される。

④　受検日には，年，月，日が，それぞれ4桁，2桁，2桁の西暦で格納される。

⑤　得点には，当該受検日の得点が格納される。得点は000～100である。

(2)　採点ファイルは，図2に示すレコード様式の順ファイルで，英語の検定テストの都度，作成される。

様式1

受検日 8桁	未使用域 1桁

様式2

従業員番号 6桁	得点 3桁

図2　採点ファイルのレコード様式

① ファイルの先頭レコードには，様式1のレコードが1レコードだけ格納される。受検日には，年，月，日が，それぞれ4桁，2桁，2桁の西暦で格納される。

② 2レコード目以降には，様式2のレコードが，従業員番号の昇順に格納される。なお，受検は希望者だけなので，この回に受検しなかった従業員のレコードは存在しない。

〔プログラム〕

（行番号）

```
 1   DATA DIVISION.
 2   FILE SECTION.
 3   FD MST-FILE.
 4   01 MST-REC.
 5      02 MST-ENUM      PIC 9(6).
 6      02 MST-AVG       PIC 9(3).
 7      02 MST-HST.
 8         03 MST-ELM    OCCURS 5.
 9            04 MST-DATE   PIC 9(8).
10            04 MST-SCORE  PIC 9(3).
11   FD SCR-FILE.
12   01 FIRST-REC.
13      02 SCR-DATE      PIC 9(8).
14      02 FILLER        PIC X(1).
15   01 SCR-REC.
16      02 SCR-ENUM      PIC 9(6).
17      02 SCR-SCORE     PIC 9(3).
18   WORKING-STORAGE SECTION.
19   77 SCR-FLAG     PIC X(1) VALUE SPACE.
20      88 SCR-EOF   VALUE "E".
21   77 W-CNT        PIC 9(2).
22   77 TEST-DATE    PIC 9(8).
23   77 SCR-TOTAL    PIC 9(4).
24   77 HST-NUM      PIC 9(1).
```

```
25   PROCEDURE DIVISION.
26   MAIN-PROC.
27       OPEN INPUT SCR-FILE
28            I-O   MST-FILE.
29       READ SCR-FILE.
30       MOVE SCR-DATE TO TEST-DATE.
31       PERFORM MATCHING-PROC.
32       CLOSE MST-FILE SCR-FILE.
33       STOP RUN.
34   MATCHING-PROC.
35       PERFORM UNTIL SCR-EOF
36          READ SCR-FILE
37             AT END
38                SET SCR-EOF TO TRUE
39             NOT AT END
40                PERFORM [          a          ]
41                   READ MST-FILE END-READ
42                END-PERFORM
43                PERFORM UPDATE-PROC
44          END-READ
45       END-PERFORM.
46   UPDATE-PROC.
47       MOVE ZERO TO SCR-TOTAL HST-NUM.
48       PERFORM [          b          ]
49          MOVE MST-ELM(W-CNT) TO MST-ELM(W-CNT + 1)
50          IF [          c          ] THEN
51             ADD 1 TO HST-NUM
52             ADD MST-SCORE(W-CNT) TO SCR-TOTAL
53          END-IF
54       END-PERFORM.
55       MOVE TEST-DATE TO MST-DATE( [    d    ] ).
56       MOVE SCR-SCORE TO MST-SCORE( [    d    ] ).
57       COMPUTE MST-AVG = (SCR-TOTAL + SCR-SCORE) / (HST-NUM + 1).
58       REWRITE MST-REC.
```

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　TEST AFTER UNTIL SCR-ENUM = MST-ENUM

イ　TEST AFTER VARYING W-CNT FROM ZERO BY 1 UNTIL W-CNT > 5

ウ　TEST BEFORE UNTIL SCR-ENUM NOT = MST-ENUM

エ　TEST BEFORE VARYING W-CNT FROM 1 BY 1 UNTIL W-CNT > 4

オ　TEST BEFORE VARYING W-CNT FROM 4 BY -1 UNTIL W-CNT = ZERO

cに関する解答群

ア　MST-DATE(W-CNT) = ZERO

イ　MST-DATE(W-CNT) NOT = ZERO

ウ　MST-SCORE(W-CNT) = ZERO

エ　MST-SCORE(W-CNT) NOT = ZERO

dに関する解答群

ア　1　　イ　5　　ウ　HST-NUM　　エ　W-CNT

設問2　検定結果の反映時に，今回の検定における得点の上位10名の従業員番号と得点を表示するようプログラムを変更する。このとき，10人目の受検者と同得点の受検者が複数いた場合は，当該受検者を全て表示する。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	SD SRT-FILE. 01 SRT-REC. 　02 SRT-ENUM　　PIC 9(6). 　02 SRT-SCORE　　PIC 9(3).
行番号24と25の間に追加	77 DISP-FLAG　　PIC X(1) VALUE SPACE. 　88 DISP-END　VALUE "E". 77 PREV-SCORE　　PIC 9(3) VALUE ZERO.
行番号31を変更	SORT SRT-FILE DESCENDING KEY SRT-SCORE 　INPUT　PROCEDURE IS MATCHING-PROC 　OUTPUT PROCEDURE IS DISPLAY-PROC.
［　e　］に追加	MOVE SCR-REC TO SRT-REC RELEASE SRT-REC

行番号58の後ろに追加	DISPLAY-PROC.
	　PERFORM TEST BEFORE
	　　VARYING W-CNT FROM 1 BY 1 UNTIL DISP-END
	　RETURN SRT-FILE
	　　AT END
	　　SET DISP-END TO TRUE
	　　NOT AT END
	　　IF ［　　f　　］ THEN
	　　SET DISP-END TO TRUE
	　　ELSE
	　　DISPLAY SRT-ENUM ": " SRT-SCORE
	　　［　　g　　］
	　　END-IF
	　END-RETURN
	END-PERFORM.

eに関する解答群

ア　行番号34と35の間　　イ　行番号41と42の間

ウ　行番号43と44の間　　エ　行番号53と54の間

fに関する解答群

ア　W-CNT = 10 AND SRT-SCORE = PREV-SCORE

イ　W-CNT = 10 AND SRT-SCORE > PREV-SCORE

ウ　W-CNT > 10 AND SRT-SCORE < PREV-SCORE

エ　W-CNT > 10 AND SRT-SCORE = PREV-SCORE

gに関する解答群

ア　ADD 1 TO W-CNT

イ　ADD SRT-SCORE TO PREV-SCORE

ウ　MOVE PREV-SCORE TO SRT-SCORE

エ　MOVE SRT-SCORE TO PREV-SCORE

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〕

雑誌記事のオンライン購読サイトをモデルとしたプログラムである。記事には，無料記事と有料記事がある。サイトの利用者は，誰でも無料記事を閲覧できる。利用者は，名前を登録することで登録会員となり，無料記事に加え，有料記事を1日に1本だけ無料で閲覧できる。登録は無料である。登録会員は，購読料を支払うことで有料会員となり，全ての記事を無制限に閲覧できる。

このプログラムは，次のクラスから成る。

(1) クラスArticle：記事を表す。

各記事には，記事識別子（id）が割り当てられる。各記事は，見出し（headline），本文（body）及び無料記事を表すフラグ（free）から成る。記事識別子と記事は，1対1に対応し，同一記事に対してArticleのインスタンスは高々1個しか存在しない。

① 静的メソッドcreate：引数で与えられた記事識別子及び記事のデータからArticleのインスタンスを生成し，登録する。記事識別子に重複はないものとする。

② 静的メソッドgetArticle：引数で与えられた記事識別子から登録されている記事であるArticleのインスタンスを返す。記事識別子に誤りはないものとする。

③ 静的メソッドgetIds：登録されている全記事の記事識別子の集合を返す。

④ メソッドisFree：無料記事であればtrueを返す。そうでなければ，falseを返す。

⑤ その他，記事のデータを返すメソッドが用意されている。

(2) 抽象クラスUser：利用者を表す。

① コンストラクタ：引数で与えられた利用者名をもつインスタンスを生成する。

② メソッドgetName：利用者名を返す。

③ 抽象メソッドtestAndMark：引数で与えられた記事が閲覧可能かどうかを調べ，閲覧可能であればtrueを返す。そうでなければ，falseを返す。この

メソッドが true を返したとき，サイトは利用者に記事を表示するものとし，必要に応じて記事が閲覧済みであることを記録する。

(3) クラス Guest：未登録の利用者（ゲスト）を表す。

① コンストラクタ：利用者名を“ゲスト”とするインスタンスを生成する。

② メソッド testAndMark：引数で与えられた記事が無料記事であれば true を返す。そうでなければ，false を返す。

(4) クラス Member：登録会員を表す。

① コンストラクタ：引数で与えられた利用者名をもつインスタンスを生成する。

② メソッド testAndMark：引数で与えられた記事が無料記事であれば，true を返す。有料記事の場合は，今日最初の有料記事の閲覧のとき，又は今日閲覧済みの有料記事と同一であるときは，true を返す。それら以外のときは，false を返す。加えて，今日最初の有料記事の閲覧のときには，閲覧済みの記事として今日の日付とともに記録する。

③ メソッド today：今日の日付（ローカル時間）を 1970 年 1 月 1 日からの日数で返す。

(5) クラス PaidMember：有料会員を表す。

① コンストラクタ：引数で与えられた利用者名をもつインスタンスを生成する。

② メソッド testAndMark：全ての記事に対して true を返す。

(6) クラス SubscriptionSite：上記クラスのテスト用プログラムである。メソッド main を実行すると，図 1 の結果が得られる。

なお，Member 及び PaidMember のコンストラクタに与える利用者名の衝突は，ないものとする。

```
ゲスト: 記事「PC入門」PC初心者…
ゲスト: 記事「スマホ特集」<閲覧不可>
ゲスト: 記事「アプリガイド」使えるアプリ…
登録会員A: 記事「PC入門」PC初心者…
登録会員A: 記事「スマホ特集」最新のスマホ…
登録会員A: 記事「アプリガイド」使えるアプリ…
有料会員B: 記事「PC入門」PC初心者…
有料会員B: 記事「スマホ特集」最新のスマホ…
有料会員B: 記事「アプリガイド」使えるアプリ…
```

図 1　メソッド main の実行結果

〔プログラム1〕

```
import java.util.Map;
import java.util.Set;
import java.util.TreeMap;

class Article {
   private static final Map<String, Article> ARTICLES =
         new TreeMap<String, Article>();
   private final String id, headline, body;
   private final boolean free;

   private Article(String id, String headline,
                   String body, boolean free) {
      this.id = id;
      this.headline = headline;
      this.body = body;
      this.free = free;
   }

   static Article create(String id, String headline,
                         String body, boolean free) {
      Article article = new Article(id, headline, body, free);
      ARTICLES.put(id, article);
      return article;
   }

   static Article getArticle(String id) {
      return ARTICLES.get(id);
   }

   static Set<String> getIds() {
      return ARTICLES.keySet();
   }

   String getId() { return id; }
   String getHeadline() { return headline; }
   String getBody() { return body; }
   boolean isFree() { return free; }
}
```

〔プログラム2〕

```
abstract class User {
   private final String name;

   User(String name) {
      this.name = name;
   }

   String getName() {
      return name;
```

```
    }

    abstract boolean testAndMark(Article article);
}
```

〔プログラム3〕

```
class Guest extends User {
    Guest() {
        super("ゲスト");
    }

    boolean testAndMark(Article article) {
        return [ a ];
    }
}
```

〔プログラム4〕

```
import java.util.TimeZone;

class Member extends User {
    private static final long MILLIS_PER_DAY = 24 * 60 * 60 * 1000L;
    private Article browsedArticle;            // 閲覧済みの有料記事
    private long browseDate = Long.MIN_VALUE;  // 記事を閲覧した日付

    Member(String name) {
        [ b ];
    }

    boolean testAndMark(Article article) {
        if (article.isFree()) {
            return true;
        }
        long today = today();
        if (browseDate [ c ] today) {
            return browsedArticle == article;
        }
        // 閲覧済みの有料記事と日付を記録
        browsedArticle = article;
        browseDate = today;
        return true;
    }

    private long today() {
        long time = System.currentTimeMillis(); // 現時刻（協定世界時）
        TimeZone tz = TimeZone.getDefault();
        time += tz.getOffset(time);             // ローカル時間に変換
        return time / MILLIS_PER_DAY;
    }
}
```

〔プログラム5〕

```
class PaidMember extends Member {
   PaidMember(String name) {
      [ b ];
   }

   boolean testAndMark(Article article) {
      return [ d ];
   }
}
```

〔プログラム6〕

```
public class SubscriptionSite {
   public static void main(String[] args) {
      User[] readers = {
         new Guest(),
         new Member("登録会員A"),
         new PaidMember("有料会員B")
      };
      Article.create("0001", "PC入門", "PC初心者…", true);
      Article.create("0002", "スマホ特集", "最新のスマホ…", false);
      Article.create("0003", "アプリガイド", "使えるアプリ…", true);
      for ( [ e ] ) {
         for ( [ f ] ) {
            Article article = Article.getArticle(id);
            String body;
            if (reader.testAndMark(article)) {
               body = article.getBody();
            } else {
               body = "<閲覧不可>";
            }
            System.out.printf("%s: 記事「%s」%s%n",
               reader.getName(), article.getHeadline(), body);
         }
      }
   }
}
```

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，dに関する解答群

ア　!article.isFree()　　イ　article.isFree()

ウ　false　　エ　this.article != article

オ　this.article == article　　カ　this.article == null

キ　true

bに関する解答群

ア　new User(name)　　イ　super()　　ウ　super(name)

エ　super.name = name　　オ　this()　　カ　this(name)

キ　this.name = name　　ク　User(name)

cに関する解答群

ア　!=　　イ　<　　ウ　==　　エ　>　　オ　instanceof

e，fに関する解答群

ア　int i = 0; i < Article.getIds().size(); i++

イ　int i = 0; i < readers.length; i++

ウ　Member reader : readers

エ　PaidMember reader : readers

オ　String id : Article.articles

カ　String id : Article.getIds()

キ　User reader : readers

設問2　プログラム 6 で記事識別子 "0003" の記事の 無料記事を表すフラグの値を false にしてプログラムを実行したとき，"＜閲覧不可＞" は何回出力されるか，正しい答えを，解答群の中から選べ。ここで，プログラム中の［ a ］～［ f ］には正しい答えが入っているものとし，プログラムは日をまたいで実行しないものとする。

解答群

ア　0回	イ　1回	ウ　2回
エ　3回	オ　4回	カ　5回

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

〔プログラム1の説明〕

文字列中の単語を切り出して，単語管理テーブルを作成する副プログラム GETTKN である。

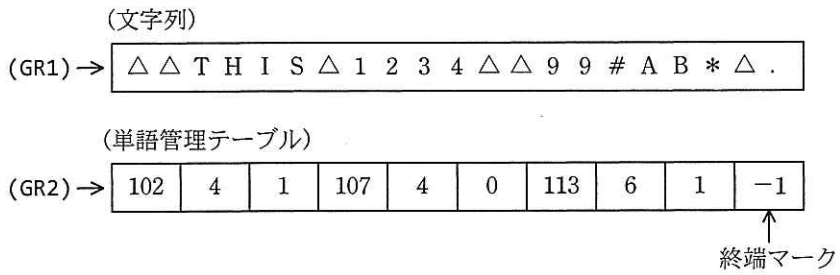
(1)　文字列は英字，数字，空白文字の 0 文字以上の並びで，最後にピリオドが置かれる。単語は，1 文字以上の空白文字又はピリオドで区切られた英数字の並びである。英字には，アルファベットの他に，ピリオド以外の記号を含むものとする。文字列表記中の“△”は空白文字を示す。

(2)　単語管理テーブルには，文字列中に現れる単語ごとに，3 語から成る要素を作成し，単語の先頭アドレス，単語の長さ，単語の種別の順に格納する。種別は，単語が数字だけから成るときは 0，英字を含むときは 1 とする。単語の切出しが終了したとき，単語管理テーブルの終端を示すマークとして −1 を格納する。

(3)　主プログラムは，文字列の先頭アドレスを GR1 に，単語管理テーブルの先頭アドレスを GR2 に設定して，GETTKN を呼ぶ。

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

副プログラム GETTKN の実行例を，図1に示す。



注記　文字列の先頭アドレスは 100 番地とし，数字は 10 進数表記とする。

図1　副プログラム GETTKN の実行例

〔プログラム1〕

```
GETTKN  START
        RPUSH
        LD      GR3,=-1         ; 単語種別の初期化
        LAD     GR1,-1,GR1
LP      LAD     GR1,1,GR1
        LD      GR4,0,GR1       ; 1文字を取り出す
        CPL     GR4,='.'
        JZE     FIN
        CPL     GR4,=' '
        JNZ     ALNUM
        CALL    SETTKN
        [      a      ]
ALNUM   LD      GR3,GR3         ; 単語の処理中？
        JPL     LP              ; 処理中の単語が英字を含む場合はLPへ
        JZE     ACHK            ; 処理中の単語が数字だけから成る場合はACHKへ
        LD      GR3,=0          ; 次の単語の処理開始
                                ; 単語種別を“数字だけから成る”に設定
        LD      GR6,GR1         ; 先頭アドレスを退避
ACHK    CPL     GR4,='9'
        JPL     NEXT
        CPL     GR4,='0'
        JMI     NEXT
        JUMP    LP              ; 取り出した文字が数字の場合はLPへ
NEXT    LD      GR3,=1          ; 単語種別を“英字を含む”に設定
        JUMP    LP
FIN     CALL    SETTKN
        LD      GR5,=-1
        ST      GR5,0,GR2       ; 終端マークを格納
        RPOP
        RET
SETTKN  LD      GR3,GR3
        JMI     FIN2            ; 単語処理中でなければ何もしない
        ST      GR6,0,GR2       ; 単語の先頭アドレスを格納
        LD      GR5,GR1
        [      b      ]
        ST      GR5,1,GR2       ; 単語の長さを格納
        ST      GR3,2,GR2       ; 単語の種別を格納
        LD      GR3,=-1         ; 単語種別を初期化（処理中状態を解除）
        LAD     GR2,3,GR2
FIN2    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	JUMP	ACHK	イ	JUMP	LP	ウ	LAD	GR1,-1,GR1
エ	LAD	GR1,1,GR1	オ	LD	GR4,0,GR1	カ	LD	GR4,1,GR1

bに関する解答群

ア	ADDL	GR5,GR6	イ	LAD	GR3,1,GR3	ウ	SLL	GR3,1
エ	SLL	GR5,1	オ	SRL	GR5,1	カ	SUBL	GR5,GR6

設問2　次の文字列が与えられ，プログラム1のラベル NEXT が付いた命令を2度目に実行した直後に GR4 に設定されている文字として，正しい答えを，解答群の中から選べ。

(GR1)→［△ 1 A 2 B △ C 3 .］

解答群

ア	1	イ	2	ウ	3
エ	A	オ	B	カ	C

設問3　GETTKN を使用して，指示に従い2数の和，差，積のいずれかを求める副プログラム CALC 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)　主プログラムは，文字列の先頭アドレスを GR1 に設定して，CALC を呼ぶ。CALC は演算結果を GR0 に設定して呼出し元に戻る。文字列の形式を図2に示す。

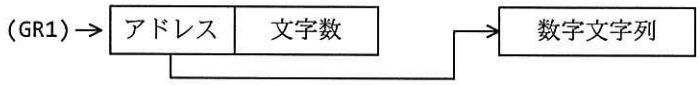
(GR1)→［数字文字列1△演算子△数字文字列2△ .］

図2　文字列の形式

(2) 数字文字列は 0 ～ 65535 の整数（符号なしの数字文字列）で 1 文字以上の文字列とし，演算子は加算（+），減算（−），乗算（*）を表す記号 1 文字とする。演算はそれぞれ論理加算，論理減算，整数の乗算として実行し，桁あふれは発生しないものとする。

(3) CALC は，GETTKN で切り出した数字だけから成る単語を 2 進数に変換するために，別に用意された副プログラム DTOB を呼ぶ。

(4) DTOB は，数字文字列格納領域の先頭アドレスと文字列の長さが順に格納された 2 語から成る領域の先頭アドレスが GR1 に設定されて，呼び出される。DTOB は数字文字列を 2 進数に変換し，GR0 に設定して呼出し元に戻る。



(5) 副プログラム CALC，DTOB 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〕

```
CALC    START
        RPUSH
        LAD     GR2,CTBL
        CALL    GETTKN
        [   c   ]
        CALL    DTOB            ; 数字文字列 1 を 2 進数に変換して
        LD      GR4,GR0         ; GR4 に設定
        LD      GR1,3,GR2
        LD      GR3,0,GR1       ; GR3 ← 演算子の文字コード
        SUBL    GR3,='*'
        LAD     GR1,6,GR2
        CALL    DTOB            ; 数字文字列 2 を 2 進数に変換して
        LD      GR5,GR0         ; GR5 に設定
        [   d   ]
        JUMP    0,GR3           ; 演算子で指定された処理にジャンプ
MULT    LD      GR0,=0          ; 乗算
        LD      GR5,GR5
LP      JZE     FIN
        LD      GR3,GR5
        AND     GR3,=#0001      ; 乗数の最下位ビットのチェック
        JZE     NEXT
```

```
        ADDL   GR0,GR4
NEXT    SLL    GR4,1          ; 被乗数を１ビット左論理シフト
               [ e ]
        JUMP   LP
PLUS    ADDL   GR4,GR5        ; 加算
        LD     GR0,GR4
        JUMP   FIN
MINUS   SUBL   GR4,GR5        ; 減算
        LD     GR0,GR4
FIN     RPOP
        RET
CTBL    DS     10             ; GETTKN 用単語管理テーブル
LTBL    DC     MULT           ; 演算の分岐先アドレステーブル
        DC     PLUS
        DS     1              ; ダミー
        DC     MINUS
        END
```

cに関する解答群

ア	LAD	GR1,3,GR2	イ	LD	GR1,0,GR2	ウ	LD	GR1,CTBL
エ	LD	GR1,GR2	オ	LD	GR2,CTBL	カ	LD	GR2,GR1

dに関する解答群

ア	LAD	GR3,LTBL	イ	LAD	GR3,LTBL,GR3	ウ	LD	GR3,LTBL
エ	LD	GR3,LTBL,GR3	オ	SLL	GR3,1			

eに関する解答群

ア	ADDL	GR0,GR5	イ	ADDL	GR5,GR4	ウ	LD	GR5,GR4
エ	SLL	GR5,1	オ	SRL	GR5,1			

問13　次の表計算の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〕

F社では，所有する顧客情報のデータを分析して営業活動に役立てている。このたび，情報漏えいなどの事故に備えるために，必要に応じて匿名性の高いデータに変換してから，分析を担当する部署に渡すことになった。そこで，顧客データに対して匿名化処理を行うプログラムを表計算ソフトで作成した。

〔ワークシート：顧客リスト〕

分析対象の顧客は5,000人である。各顧客に対して固有のID，氏名，郵便番号，住所，電話番号，年齢，職業コードの情報を所有している。顧客情報を格納したワークシート“顧客リスト”の例を図1に示す。ここで，郵便番号は7桁の数値として取り扱う。

	A	B	C	D	E	F	G
1	ID	氏名	郵便番号	住所	電話番号	年齢	職業コード
2	0001	滝本卓也	1070062	東京都港区南青山…	03xxxxxxxx	66	111
3	0002	中野浩二	1690051	東京都新宿区西早稲田…	03xxxxxxxx	59	132
4	0003	下山健	1110042	東京都台東区寿…	03xxxxxxxx	37	214
5	0004	飯山由美	1830003	東京都府中市朝日町…	042xxxxxxx	19	312
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
5000	4999	山本礼子	5300042	大阪府大阪市北区天満橋…	06xxxxxxxx	29	112
5001	5000	田中陽子	2400012	神奈川県横浜市保土ヶ谷…	045xxxxxxx	36	321

図1　ワークシート“顧客リスト”の例

〔ワークシート：匿名化顧客リスト〕

ワークシート“顧客リスト”を基に，匿名化処理を施したワークシート“匿名化顧客リスト”を作成した。ワークシート“匿名化顧客リスト”の例を図2に示す。

なお，列H，列Iは，後述のマクロGenerateListで使用する。

	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1	ID	氏名	郵便番号	住所	電話番号	年齢	職業コード	マーキング	評価値		表示年齢	職業コード変換		多重度
2	0001	*	10	*	*	60	100				0	100	*	3
3	0002	*	16	*	*	50	100				20	110		
4	0003	*	11	*	*	35	210				25	120	*	
5	0004	*	18	*	*	0	312				30	200		
6	0005	*	27	*	*	50	100				35	210	*	
7	0006	*	20	*	*	25	312				40	220		
8	0007	*	20	*	*	30	230				50	230	*	
9	0008	*	25	*	*	60	100				60	240		
10	0009	*	26	*	*	40	400				1000	300		
11	0010	*	22	*	*	30	222					310		
12	0011	*	22	*	*	35	225					320	*	
5000	4999	*	53	*	*	25	100							
5001	5000	*	24	*	*	35	320							

図2　ワークシート“匿名化顧客リスト”の例

設問1　ワークシート“匿名化顧客リスト”に関する次の記述中の［　　］に入れる正しい答えを，解答群の中から選べ。ここで，c1 と c2 に入れる答えは，c 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。

(1)　氏名，住所，電話番号の情報は完全に秘匿するために，“*”にする。そこでセル B2，D2，E2 に“*”を入力し，それぞれセル B3 ～ B5001，D3 ～ D5001，E3 ～ E5001 に複写する。

(2)　郵便番号は，7 桁の郵便番号のうち上位 2 桁だけの値に置き換えることで匿名化する。そこで，次の式をセル C2 に入力し，セル C3 ～ C5001 に複写する。

［　a　］

(3)　セル F2 ～ F5001 には，図 3 に示す各年齢区分の最小値である表示年齢を表示することで匿名化する。各年齢区分の最小値である表示年齢を，セル K2 ～ K9 に昇順に入力する。また，セル K10 には，終端を示す 1000 を入力する。

表計算

年齢区分	表示年齢
0～19歳	0
20～24歳	20
25～29歳	25
30～34歳	30
35～39歳	35
40～49歳	40
50～59歳	50
60歳以上	60

図3　年齢区分

例えば，顧客の年齢が23歳の場合，“20”と表示する。そこで，次の式をセルF2に入力し，セルF3～F5001に複写する。

```
b
```

(4)　職業コードは図4に示す項目から成る3桁の数値である。各桁は，1～9の整数値である。

大分類 1桁	中分類 1桁	小分類 1桁

図4　職業コードの様式

セルG2～G5001には，必要に応じて大分類化するか中分類化することで匿名化した職業コードを表示する。大分類化とは，中分類，小分類の値をいずれも0に置き換えた数値に変換することであり，中分類化とは，小分類の値を0に置き換えた数値に変換することである。大分類化，中分類化された3桁の数値をそれぞれ大分類コード，中分類コードと呼ぶ。

セルL2～M41には，職業コードの大分類化，中分類化に関する情報が入力されている。存在し得る大分類コード，中分類コードは合わせて40種類であり，セルL2～L41に昇順で格納されている。次の規則によって，職業コードを変換する。

① 大分類コードが格納されているセルの右側の対応するセルに“＊”が格納されている場合，その大分類に属する職業コードは全て大分類コードに変換する。例えば，セル L2 には大分類コード 100 が格納されていて，その右隣のセル M2 は“＊”なので，100 番台の全ての職業コードは 100 に変換する。

② ①に該当しない場合であって，中分類コードが格納されているセルの右側の対応するセルに“＊”が格納されている場合，その中分類に属する職業コードは全て中分類コードに変換する。例えば，セル L6 には中分類コード 210 が格納されていて，右隣のセル M6 は“＊”なので，210 番台の全ての職業コードは 210 に変換する。

③ ①，②のいずれも該当しない場合，職業コードは変換しない。

そこで，次の式をセル G2 に入力し，セル G3～G5001 に複写する。

IF(垂直照合(c1 , L$2:M$41, 2, 0)＝'＊', c1 ,
IF(垂直照合(c2 , L$2:M$41, 2, 0)＝'＊', c2 ,
顧客リスト!G2))

a に関する解答群

ア 剰余(10 ^ 5, 顧客リスト!C2)

イ 剰余(顧客リスト!C2, 10 ^ 5)

ウ 整数部((顧客リスト!C2／10) ^ 5)

エ 整数部(顧客リスト!C2／10 ^ 5)

オ 整数部(顧客リスト!C2／10) ^ 5

b に関する解答群

ア 垂直照合(顧客リスト!F2, K$2:K$10, 1, 0)

イ 垂直照合(顧客リスト!F2, K$2:K$10, 1, 1)

ウ 垂直照合(顧客リスト!F2＋1, K$2:K$10, 1, 0)

エ 垂直照合(顧客リスト!F2＋1, K$2:K$10, 1, 1)

オ 表引き(K$2:K$10, 照合一致(顧客リスト!F2, K$2:K$10, －1), 1)

カ 表引き(K$2:K$10, 照合一致(顧客リスト!F2 ＋ 1, K$2:K$10, 1), 1)

cに関する解答群

	c1	c2
ア	切捨て(顧客リスト!G2, −1)	顧客リスト!G2
イ	切捨て(顧客リスト!G2, −2)	切捨て(顧客リスト!G2, −1)
ウ	切捨て(顧客リスト!G2, −2)	顧客リスト!G2
エ	顧客リスト!G2	切捨て(顧客リスト!G2, −1)
オ	顧客リスト!G2	切捨て(顧客リスト!G2, −2)

〔ワークシート：提供リスト〕

匿名化された後の郵便番号，年齢，職業コードが全て同一となる顧客の数（多重度と呼ぶ）のリストをデータ分析の担当部署に渡すために，図 5 に示すワークシート“提供リスト”を作成する。

	A	B	C	D
1	郵便番号	年齢	職業コード	多重度
2	00	20	230	4
3	00	60	500	4
4	01	20	241	5
⋮	⋮	⋮	⋮	⋮
206	98	60	500	4
207	99	60	500	3

注記　郵便番号は 2 桁で表示している。

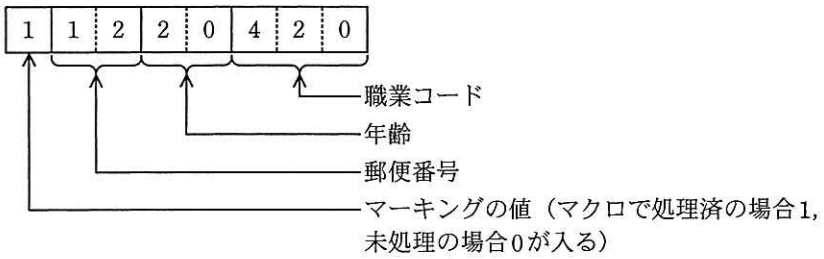
図 5　ワークシート“提供リスト”

(1) ワークシート“匿名化顧客リスト”にマクロ GenerateList を実装する。

(2) マクロ GenerateList は，ワークシート“提供リスト”の列 A ～ D に，ワークシート“匿名化顧客リスト”を郵便番号，年齢，職業コードの優先順で昇順に整列し，多重度とともに表示する。

(3) ワークシート“匿名化顧客リスト”のセル H2 ～ H5001 は，マクロ GenerateList の実行中に，どの顧客の情報が処理済みであるかを示すのに用いる。処理済みの顧客の場合は 1 を，そうでない場合は 0 を表示する。

(4) (2)の処理を行う際，次に処理する顧客を決定するために，ワークシート“匿名化顧客リスト”中の郵便番号，年齢，職業コードから成る次のような 8 桁の数値（以下，評価値という）を定義する。



この評価値が最小の顧客を処理対象とし，多重度を順次求めていく。

そこで，次の式をワークシート“匿名化顧客リスト”のセル I2 に入力し，セル I3～I5001 に複写する。

H2 * 10000000 + C2 * 100000 + F2 * 1000 + G2

(5) ワークシート“匿名化顧客リスト”のセル N2 には，表示する多重度の下限値が入力されている。ここで，表示する多重度の下限値は 2 以上の整数とする。多重度がこの値以上の郵便番号，年齢，職業コードの組を，多重度とともにワークシート“提供リスト”に表示する。

設問 2 マクロ GenerateList をワークシート“匿名化顧客リスト”に格納した。マクロ GenerateList 中の［　　］に入れる正しい答えを，解答群の中から選べ。

表計算

〔マクロ：GenerateList〕

○マクロ：GenerateList
○数値型：numCustomer, minID, previousValue, minMultiplicity, I, J, K

- numCustomer ← 5000
- previousValue ← 0
- minMultiplicity ← N2
- 相対(提供リスト!A1, 0, 0) ← '郵便番号'
- 相対(提供リスト!A1, 0, 1) ← '年齢'
- 相対(提供リスト!A1, 0, 2) ← '職業コード'
- 相対(提供リスト!A1, 0, 3) ← '多重度'

■ I: 1, I ≦ numCustomer, 1
- 相対(H1, I, 0) ← 0

■

- J ← 1
- K ← 0

■ I: 1, I ≦ numCustomer, 1
- minID ← [d]
- 相対(H1, minID, 0) ← 1

▲ 相対(I1, minID, 0) ≠ previousValue

▲ K ≧ minMultiplicity
- [e]

- 相対(提供リスト!A1, J, 0) ← 相対(A1, minID, 2)
- 相対(提供リスト!A1, J, 1) ← 相対(A1, minID, 5)
- 相対(提供リスト!A1, J, 2) ← 相対(A1, minID, 6)
- K ← 1
- 相対(提供リスト!A1, J, 3) ← K
- previousValue ← 相対(I1, minID, 0)

—

- [f]
- 相対(提供リスト!A1, J, 3) ← K

■

▲ 相対(提供リスト!A1, J, 3) ＜ minMultiplicity
- 相対(提供リスト!A1, J, 0) ← null
- 相対(提供リスト!A1, J, 1) ← null
- 相対(提供リスト!A1, J, 2) ← null
- 相対(提供リスト!A1, J, 3) ← null

dに関する解答群

ア　条件付個数(I2～I5001, ＞相対(I1, I, 0))

イ　条件付個数(I2～I5001, ＜相対(I1, I, 0))

ウ　照合一致(最小(I2～I5001), I2～I5001, 0)

エ　照合一致(最小(I2～I5001), I2～I5001, 1)

オ　照合一致(最大(I2～I5001), I2～I5001, 0)

カ　照合一致(最大(I2～I5001), I2～I5001, 1)

キ　相対(I1, I, 0)

ク　相対(I1, J, 0)

e, fに関する解答群

ア　I ← I + 1　　イ　I ← I + J　　ウ　I ← I + K

エ　J ← I　　オ　J ← J + 1　　カ　J ← J + K

キ　J ← K + 1　　ク　K ← J　　ケ　K ← K + 1

コ　K ← K + J

■ Java プログラムで使用する API の説明

java.util

public interface Map<K, V>

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

メソッド

`public V get(Object key)`

指定されたキーに対応付けられた値を返す。

引数： key — キー

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

`public Set<K> keySet()`

登録されているキーの集合を返す。

戻り値：登録されているキーの集合

`public V put(K key, V value)`

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

引数： key — キー
value — 値

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

---

java.util

public class TreeMap<K, V>

インタフェース Map の実装である。マップはキーの昇順に整列される。

コンストラクタ

`public TreeMap()`

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

---

java.util

public interface Set<E>

型 E の要素を集合として管理するインタフェースを提供する。

java.util

public class TimeZone

クラス TimeZone は，タイムゾーンを表す。

メソッド

---

public static TimeZone getDefault()

デフォルトの TimeZone のインスタンスを返す。通常は JVM を実行している OS のタイムゾーン設定と同じである。

戻り値：デフォルトのTimeZoneのインスタンス

---

public int getOffset(long date)

引数 date で指定された時刻における協定世界時（UTC）からの時差（ミリ秒）を返す。例えば，日本標準時を表す TimeZone のインスタンスに対して getOffset(0L)を呼び出すと，32400000（9 時間）を返す。

引数： date — 時刻（1970 年 1 月 1 日午前 0 時（協定世界時）からのミリ秒単位の相対時間）

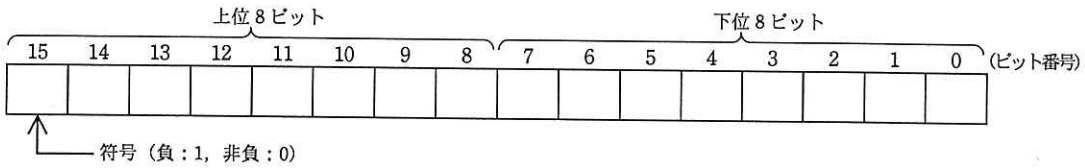
戻り値：指定された時刻における協定世界時（UTC）からの時差

■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) (r) ＞ (実効アドレス)	0	0
(r1) ＝ (r2) (r) ＝ (実効アドレス)	0	1
(r1) ＜ (r2) (r) ＜ (実効アドレス)	1	0

(4) シフト演算命令

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

(5) 分岐命令

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

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

(6) スタック操作命令

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

(7) コール，リターン命令

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

(8) その他

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

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

1.3 文字の符号表

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

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

2. アセンブラ言語 CASLⅡの仕様

2.1 言語の仕様

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

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

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

2.2 命令の種類

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

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

2.3 アセンブラ命令

アセンブラ命令は，アセンブラの制御などを行う。

(1)

START	[実行開始番地]

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

(2) | END | |

END 命令は，プログラムの終わりを定義する。

(3) | DS | 語数 |

DS 命令は，指定した語数の領域を確保する。

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

(4) | DC | 定数［,定数］… |

DC 命令は，定数で指定したデータを（連続する）語に格納する。

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

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

2.4 マクロ命令

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

(1) | IN | 入力領域,入力文字長領域 |

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

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

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

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

(2) | OUT | 出力領域,出力文字長領域 |

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

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

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

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

(3) RPUSH

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

(4) RPOP

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

2.5 機械語命令

機械語命令のオペランドは，次の形式で記述する。

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

2.6 その他

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

(2) 生成された命令語，領域は，主記憶上で連続した領域を占める。

3. プログラム実行の手引

3.1 OS

プログラムの実行に関して，次の取決めがある。

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

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

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

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

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

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

3.2 未定義事項

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

表計算ソフトの機能・用語（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，読出し，印刷，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”のセル範囲 B5 ～ G10 を，別のワークシートから指定する場合には，シート1!B5～G10 と表す。

3. 値と式

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

(2) 文字列は一重引用符“ ’ ”で囲って表す。

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

(3) 論理値の真を true，偽を false と表す。

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

(5) セルには，式を入力することができる。セルは，式を評価した結果の値をもつ。

(6) 式は，定数，セル番地，演算子，括弧及び関数から構成される。定数は，数値，文字列，論理値又は空値を表す表記とする。式中のセル番地は，その番地のセルの値を参照する。

(7) 式には，算術式，文字式及び論理式がある。評価の結果が数値となる式を算術式，文字列となる式を文字式，論理値となる式を論理式という。

(8) セルに式を入力すると，式は直ちに評価される。式が参照するセルの値が変化したときには，直ちに，適切に再評価される。

4. 演算子

(1) 単項演算子は，正符号“＋”及び負符号 “－”とする。

(2) 算術演算子は，加算“＋”，減算“－”，乗算“＊”，除算“／”及びべき乗“＾”とする。

(3) 比較演算子は，より大きい“＞”，より小さい“＜”，以上“≧”，以下“≦”，等しい“＝”及び等しくない“≠”とする。

(4) 括弧は丸括弧“(”及び“)”を使う。

(5) 式中に複数の演算及び括弧があるときの計算の順序は，次表の優先順位に従う。

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

5. セルの複写

(1) セルの値又は式を，他のセルに複写することができる。

(2) セルを複写する場合で，複写元のセル中にセル番地を含む式が入力されているとき，複写元と複写先のセル番地の差を維持するように，式中のセル番地を変化させるセルの参照方法を相対参照という。この場合，複写先のセルとの列番号の差及び行番号の差を，複写元のセルに入力された式中の各セル番地に加算した式が，複写先のセルに入る。

[例] セル A6 に式 A1 ＋ 5 が入力されているとき，このセルをセル B8 に複写すると，セル B8 には式 B3 ＋ 5 が入る。

(3) セルを複写する場合で，複写元のセル中にセル番地を含む式が入力されているとき，そのセル番地の列番号と行番号の両方又は片方を変化させないセルの参照方法を絶対参照という。絶対参照を適用する列番号と行番号の両方又は片方の直前には“$”を付ける。

[例] セル B1 に式 A1 ＋ $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 A1 + $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る。

書式	解　　説
合計 (セル範囲[1])	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[1])	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[1])	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[1])	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[1])	セル範囲に含まれる数値の最大値を返す。
最小 (セル範囲[1])	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, …)[2]	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, …)[2]	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） 四捨五入（算術式，桁位置） 切捨て(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。
結合(式1,式2,…)[2]	式1，式2，… のそれぞれの値を文字列として扱い，それらを引数の順につないでできる一つの文字列を返す。 ［例］ 結合('北海道','九州',123,456) は，文字列“北海道九州123456”を返す。
順位（算術式，セル範囲[1]，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)	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。 ・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，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。 ・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。 ［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。
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[1)])	行数及び列数が共に同じ検索のセル範囲と合計のセル範囲に対して，検索と合計を行う。検索のセル範囲に含まれるセルのうち，検索条件の記述で指定される条件を満たすセルを全て探す。検索条件の記述を満たした各セルについての左上端からの位置と，合計のセル範囲中で同じ位置にある各セルの値を合計して返す。 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。 ［例1］条件付合計(A1 ～ B8,＞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。 ［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。

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

[2)] 引数として渡すことができる式の個数は，1 以上である。

7. マクロ

(1) ワークシートとマクロ

ワークシートには複数のマクロを格納することができる。

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

［例］○**マクロ**：Pro

例は，マクロ Pro の宣言である。

(2) 変数とセル変数

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

［例］○**数値型**：row，col

例は，数値型の変数 row，col の宣言である。

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

セル変数の表現方法には，絶対表現と相対表現とがある。

セル変数の絶対表現は，セル番地で表す。

セル変数の相対表現は，次の書式で表す。

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

［例1］**相対(B5, 2, 3)** は，セル E7 を表す変数である。

［例2］**相対(B5, −2, −1)** は，セル A3 を表す変数である。

(3)　配列

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

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

［例］○**文字列型**：table[100, 200]

例は，100 × 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 〕

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

退室可能時間	13:40 ～ 15:20

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