FE

平成27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 27 年 4 月 19 日実施

基本情報技術者試験　午後　問題

ページ	問題番号	行	誤	正	訂正の内容
63	13	上から 8 行目	…，解答群の中から選べ。	…，解答群の中から選べ。ここで，関数“表引き”は，行の位置として 0 が指定された場合には，null を返すものとする。	下線部分を追加する。

〔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	インターネットを利用した受注管理システムのセキュリティ

●問 2～問 7（6 問中 4 問選択）

問題番号	出題分野	テーマ
問 2	ソフトウェア	言語処理系
問 3	データベース	自治会員の情報を管理する関係データベースの設計及び運用
問 4	ネットワーク	ホスト名の衝突
問 5	ソフトウェア設計	営業支援システム
問 6	プロジェクトマネジメント	プロジェクトにおけるコミュニケーションの計画
問 7	システム戦略	システム開発の投資評価

●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	クイックソートを応用した選択アルゴリズム

●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）	換字式暗号
問 10	ソフトウェア開発（COBOL）	従業員の勤務管理
問 11	ソフトウェア開発（Java）	セキュアプログラミング
問 12	ソフトウェア開発（アセンブラ）	階乗の計算
問 13	ソフトウェア開発（表計算）	学習進捗管理

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　インターネットを利用した受注管理システムのセキュリティに関する次の記述を読んで，設問1～4に答えよ。

製造業のK社では，インターネットを利用した受注管理システムを開発している。受注管理システムは，取引先も利用するので，セキュリティ上の欠陥があった場合，自社だけでなく取引先にも損害を与える可能性がある。そこで，K社は，セキュリティ診断サービスを行っているZ社に，受注管理システムの脆弱性診断を依頼した。

〔受注管理システム〕

受注管理システムのアプリケーション（以下，受注管理アプリケーションという）は，Webサーバ上で稼働する。受注や出荷などの情報は，データベース（以下，DBという）サーバ上で稼働する受注情報DBに格納され，受注管理アプリケーションから，参照，更新される。取引先PCにダウンロードできるファイルや，取引先PCからアップロードされたファイルは，Webサーバに接続されているディスクに格納される。受注管理システムの構成を図1に示す。

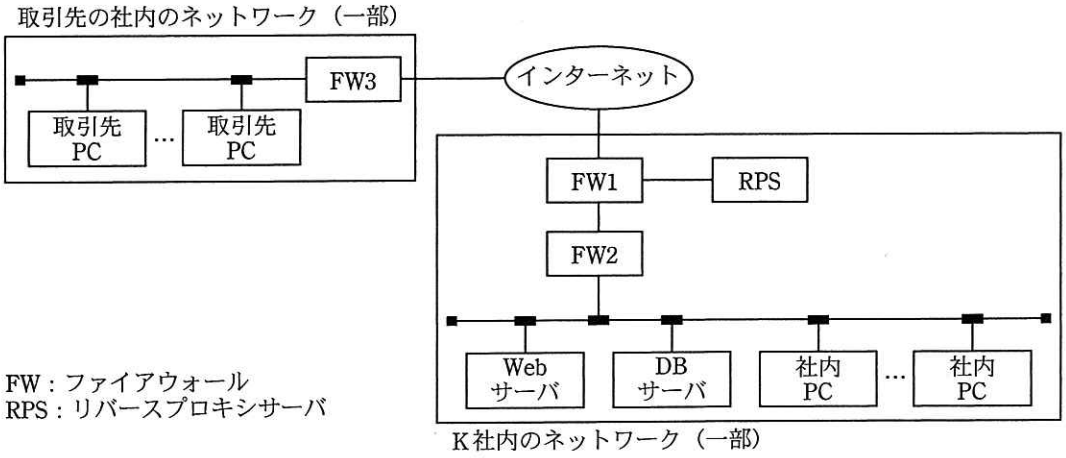


図1　受注管理システムの構成

RPS には，ディジタル証明書を設定しておく。受注管理システムを利用する取引先の担当者は，取引先 PC のブラウザから RPS を経由して受注管理アプリケーションにアクセスし，ログイン画面で利用者 ID とパスワードを入力してログインする。その際，取引先 PC のブラウザからの通信には，HTTP over SSL/TLS（以下，HTTPS という）を使用する。RPS ではディジタル証明書を使って，HTTPS から HTTP にプロトコルを変換する。

〔Z 社の脆弱性診断の結果〕

受注管理アプリケーションには，想定していない操作を DB サーバに実行させて，DB に不正アクセスするような［ a ］については，対策がされている。しかし，Z 社の脆弱性診断の結果，受注管理アプリケーションに対策が必要なセキュリティ上の脆弱性が複数指摘された。表 1 に Z 社からの指摘事項（抜粋）を示す。

表 1　Z 社からの指摘事項（抜粋）

指摘事項	原因
取引先の担当者が別の取引先の発注情報や出荷情報にアクセス可能である。	
取引先の担当者が Web サーバ上の任意のファイルをダウンロード可能である。	①受注管理アプリケーションでのファイルのダウンロード処理に問題がある。
攻撃者によって Web ページ内にスクリプトが埋め込まれてしまう［ b ］の脆弱性があるので，取引先の担当者が他の Web サイトに誘導されて，利用者 ID とパスワードを奪取される可能性がある。	
	②取引先の担当者がログイン時にパスワードを連続して間違えても利用者 ID がロックされない。

注記　網掛けの部分は表示していない。

K 社は，表 1 中の下線①及び②に対策を行った。さらに，Z 社からのその他の指摘事項にも対策を行って，K 社は，受注管理システムの運用を開始することにした。

設問1　図1中の通信経路を表2に示す1～5とした場合，取引先PCからWebサーバにアクセスするときに，HTTPSが通信に使われる通信経路だけを全て示す正しい答えを，解答群の中から選べ。

表2　各機器の間の通信経路

経路番号	通信経路
1	取引先PCとFW3との間
2	FW3とFW1との間
3	FW1とRPSとの間
4	FW1とFW2との間
5	FW2とWebサーバとの間

解答群

ア　1　　イ　1，2，3　　ウ　1，2，3，4

エ　1，2，3，4，5　　オ　2，3，4　　カ　2，3，4，5

キ　3，4

設問2　本文中の［　　］に入れる適切な答えを，解答群の中から選べ。

a，bに関する解答群

ア　DoS攻撃　　イ　SQLインジェクション

ウ　クロスサイトスクリプティング　　エ　辞書攻撃

オ　ディレクトリトラバーサル　　カ　トラッシング

キ　ブルートフォース攻撃　　ク　ポートスキャン

設問3　表1中の下線①の対策として適切な答えを，解答群から選べ。

解答群

ア　ダウンロードしたいファイルを絶対パスで指定させ，該当ファイルが存在する場合には，ダウンロードの処理を行う。

イ　ダウンロードしたいファイルを相対パスで指定させ，該当ファイルが存在する場合には，ダウンロードの処理を行う。

ウ　ダウンロードしたいファイルのファイル名だけを指定させ，取引先ごとに決められたフォルダ内に該当ファイルが存在する場合には，ダウンロードの処理を行う。

エ　取引先 PC のブラウザに，Web サーバ上の全てのフォルダ構成及びファイルを表示し，ダウンロードしたいファイルを指定させ，ダウンロードの処理を行う。

設問4　表 1 中の下線②の脆弱性から考えられるセキュリティ事故として適切な答えを，解答群の中から選べ。

解答群

ア　取引先の担当者が誕生日をパスワードにしていると，誕生日を知っている者がログインできてしまう。

イ　パスワードの候補を自動で次々と入力するプログラムを利用することで，ログインできてしまう。

ウ　パスワードを記載したメモを取引先の担当者が落とし，それを拾った者がログインできてしまう。

エ　ログイン操作を背後から盗み見て，パスワードを入手し，ログインできてしまう。

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　言語処理系に関する次の記述を読んで，設問1～3に答えよ。

言語処理系とは，プログラム言語（以下，言語という）の文法に従って記述された原始プログラムを読み取り，目的とする計算機で実行するための変換をするか，又は逐次解釈しながら実行するためのソフトウェアである。

設問1　言語処理系の種類には，インタプリタやコンパイラなどがある。インタプリタによる実行では，原始プログラムを入力して，逐次解釈しながら実行する（以下，インタプリタ方式という）。コンパイラによる実行では，原始プログラムを入力して，目的とする計算機で実行できるプログラム（以下，実行形式プログラムという）に変換し，その実行形式プログラムを実行する（以下，コンパイラ方式という）。インタプリタ方式とコンパイラ方式を比較したとき，コンパイラ方式の利点として最も適切な答えを，解答群の中から選べ。

解答群

ア　原始プログラムを対話的に確認しながら実行できる。

イ　原始プログラムを変更して，直ちに実行できる。

ウ　実行途中に異常が発生した場合，原始プログラムのどこでどのような原因によって異常が発生したのかを確認しやすい。

エ　目的とする計算機に対応して，実行時間を短縮するための最適化が図れる。

設問2　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。

原始プログラムを，ソフトウェアによって仮想的に構築した計算機（以下，仮想計算機という）で実行できるコード（以下，中間コードという）に変換し，仮想計算機上で中間コードをインタプリタ方式で実行する方法がある。言語 X で記述された原始プログラムを中間コードに変換し，仮想計算機 V のインタプリタで実行する例を，図 1 に示す。ここで，異なる OS やハードウェア上で動作する仮想計算機 V を用意することによって，［ a ］中間コードの形でプログラムを配布することが可能となる。

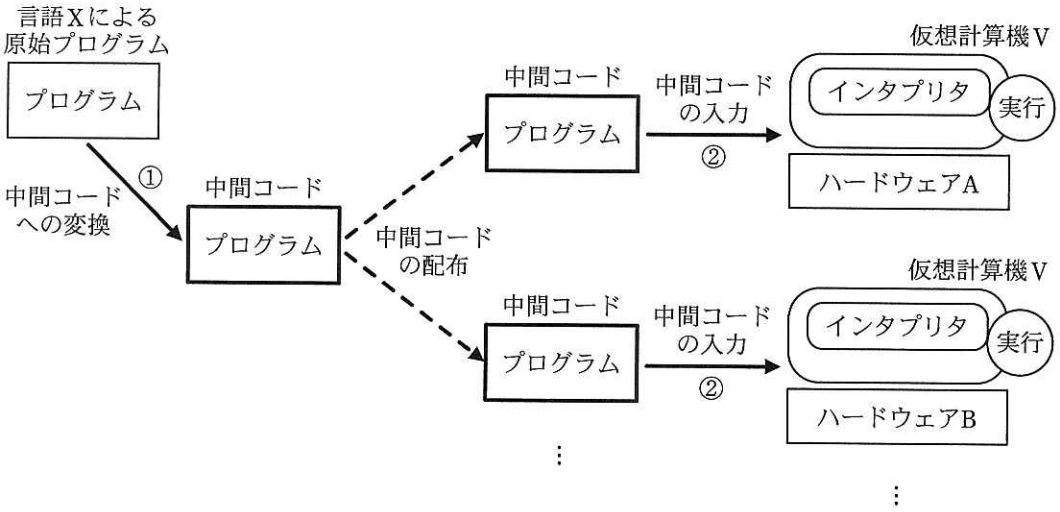


図 1　仮想計算機によるインタプリタ方式の実行例

〔図 1 の説明〕

(1)　言語 X で記述された原始プログラムを，仮想計算機 V で実行できる中間コードに変換する（図 1①）。

(2)　中間コードに変換されたプログラムを仮想計算機 V に入力し，V のインタプリタで逐次解釈しながら実行する（図 1②）。

aに関する解答群

ア　特定のOSやハードウェアに依存しない

イ　特定のOSやハードウェアに依存する

ウ　特定のハードウェアの性能を引き出す

エ　特定のハードウェアのメモリ使用量を低減する

設問3　次の記述中の［　　］に入れる正しい答えを，解答群の中から選べ。

仮想計算機 V におけるインタプリタ方式による実行時に，繰り返し実行される処理などの情報（以下，プロファイル情報という）を収集しておき，その内容を解析して，特定の処理の中間コードをプログラムの実行途中で実行形式プログラムに変換し，以後の実行に実行形式プログラムを利用する方法として，図2のような動的コンパイル方式がある。実行形式プログラムを実行途中から利用することによって，以後のプログラムの実行性能の向上が期待できるので，Java などの言語処理系で採用されている方式である。

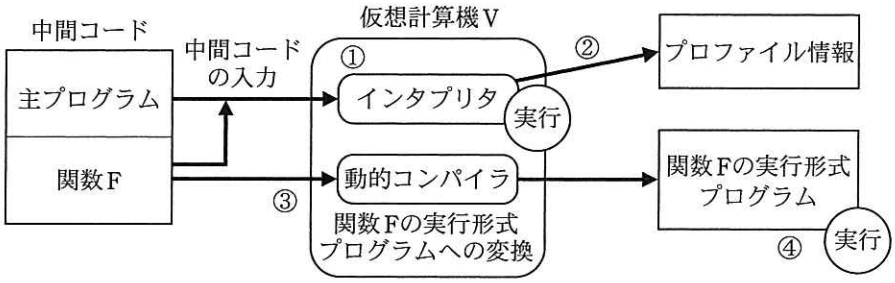


図2　動的コンパイル方式を適用した処理の流れ

〔図2の説明〕

(1)　主プログラムと主プログラムから呼び出される関数 F が中間コードに変換されたプログラムがある。

(2)　仮想計算機 V のインタプリタは，中間コードを逐次解釈しながら実行する（図2①）。このとき，関数 F が呼び出される回数をプロファイル情報として収集する（図2②）。

(3)　仮想計算機 V は，関数 F が呼び出される都度，プロファイル情報を解析し，関数 F を実行形式プログラムに変換するかどうかを判定する。

(4) (3)で変換すると判定した場合，仮想計算機 V は，動的コンパイラを起動して関数 F を実行形式プログラムに変換する（図2③）。

(5) (4)で関数 F を変換した後は，関数 F が呼び出されたときには実行形式プログラムが実行される（図2④）。

動的コンパイル方式を適用する場合には，実行時間に加えて動的コンパイラの起動時間やコンパイル時間を考慮する必要がある。図 2 に示す関数 F を，インタプリタ方式で実行した場合と，動的コンパイル方式で実行した場合の実行時間について考える。ここで，各方式の実行条件は，次の条件に示すとおりである。主プログラムから関数 F の呼出し回数が 400 回のとき，インタプリタ方式の場合の実行時間は［ b ］秒であり，動的コンパイル方式を適用した場合の実行時間は［ c ］秒である。

〔条件〕

・主プログラムの実行時間は考えない。
・プロファイル情報を収集する時間と，仮想計算機 V から実行形式プログラムを呼び出すのに必要な処理時間は考えない。
・関数 F の中間コードは 400 命令から成り，関数 F が 1 回呼び出されたときに実行する中間コードの命令数は，2,000 命令である。
・動的コンパイル方式を適用した場合，関数 F が 101 回目に呼び出されるときに動的コンパイラが起動され，関数 F の中間コードを実行形式プログラムに変換する。
・動的コンパイラの起動時間とコンパイル時間は，実行時間に含める。
・動的コンパイラの起動時間は 0.1 秒とし，コンパイル時間は，中間コード 1,000 命令当たりで 0.1 秒とする。
・インタプリタによる中間コード 1 命令の実行時間は 500 ナノ秒とし，中間コード 1 命令に対応する実行形式プログラムの実行時間は 10 ナノ秒とする。

b，c に関する解答群

ア 0.106	イ 0.146	ウ 0.206
エ 0.246	オ 0.4	カ 0.406

問3　自治会員の情報を管理する関係データベースの設計及び運用に関する次の記述を読んで，設問1～4に答えよ。

X地区の自治会では，世帯数の増加と，個人情報管理の厳格化を背景に，手書きの帳票で管理していた自治会員の情報を電子化することにした。この自治会には，236世帯，667人が登録されていて，各世帯は1～8班のいずれかに所属している。

従来は図1に示すとおり，世帯ごとに，世帯主氏名，住所，電話番号，登録日，所属する班，同居者氏名，続柄，性別，生年月日などの情報を管理していた。

世帯主	氏名	住所		電話番号	登録日	班
	情報太郎	桜ヶ丘３０４－８		999-999-9999	2005年4月7日	2
世帯情報	氏名	続柄	性別	生年月日	備考	
	情報太郎	本人	男	1969年1月23日		
	情報花子	配偶者	女	1972年8月24日		
	情報一郎	子	男	2004年6月13日		
	情報二郎	子	男	2007年1月　5日		

図1　世帯情報を管理していた帳票と記入例

図1の帳票を基に，図2に示す表構成をもつ関係データベースを作成した。下線付きの項目は，主キーを表す。

世帯表

世帯番号	世帯主番号	住所	電話番号	登録日	班
0181	0412	桜ヶ丘３０４－８	999-999-9999	20050407	02
⋮	⋮	⋮	⋮	⋮	⋮

会員表

会員番号	世帯番号	氏名	続柄	性別	生年月日	備考
0412	0181	情報太郎	01	01	19690123	
0413	0181	情報花子	02	02	19720824	
⋮	⋮	⋮	⋮	⋮	⋮	⋮

続柄表

続柄	名称
01	本人
02	配偶者
⋮	⋮

性別表

性別	名称
01	男性
02	女性

図２　表構成とデータ格納例

〔表の説明〕

(1)　世帯表の世帯番号には，各世帯に一意に割り当てた番号が格納されている。

(2)　世帯表の世帯主番号には，世帯主の会員番号が格納されている。

(3)　会員表の会員番号には，各会員に一意に割り当てた番号が格納されている。

(4)　会員表の世帯番号には，会員が属する世帯の世帯番号が格納されている。

設問１　図 2 の表構成を完成するまでの設計について，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。

図 1 の帳票は非正規形なので，まず，［ a ］世帯情報を取り出して会員表を作成した。次に，［ b ］続柄表と性別表を作成した。続柄表と性別表のレコード件数は少ないが，例えば，帳票の記載で“男”と“男性”のように同じ意味を表すデータに対する定義の一意性を保証できる効果がある。

解答群

ア　障害回復のために　　イ　第1正規化に基づいて　ウ　第2正規化に基づいて

エ　第3正規化に基づいて　オ　排他制御のために

設問2　地区の福祉委員会から，1940 年よりも前に生まれた会員が含まれる世帯の世帯番号について，情報提供を求められた。該当する世帯番号を抽出する正しいSQL 文を解答群の中から選べ。

なお，同じ世帯番号は一つだけ抽出する。

解答群

ア
```
SELECT DISTINCT 世帯番号 FROM 会員表 WHERE 会員表.生年月日 >= 19400101
```

イ
```
SELECT 世帯番号 FROM 会員表 WHERE 会員表.生年月日 >= 19391231
```

ウ
```
SELECT 世帯番号 FROM 会員表 WHERE 会員表.生年月日 < 19400101
                                  GROUP BY 世帯番号
```

エ
```
SELECT 世帯番号 FROM 会員表 WHERE 会員表.生年月日 <= 19391231
                                  GROUP BY 世帯番号，会員番号
```

設問3　班ごとの会員数に偏りがあるとの意見が挙がったので，班の再編を検討することになった。現在の状況を確認するために，班ごとの世帯数と会員数を集計する。次の SQL 文の［　　］に入れる正しい答えを，解答群の中から選べ。

```
SELECT 世帯表.班，［　　c　　］
   FROM 世帯表，会員表
   WHERE 世帯表.世帯番号 = 会員表.世帯番号
   GROUP BY 世帯表.班
```

解答群

ア　COUNT(*)，COUNT(会員表.会員番号)

イ　COUNT(*)，MAX(会員表.会員番号)

ウ　COUNT(DISTINCT 世帯表.世帯番号)，COUNT(*)

エ　COUNT(世帯表.世帯番号)，MAX(会員表.会員番号)

設問4　地区の子供会役員から，子供会に所属する子供の情報を照会できるようにしてほしいとの要望が挙がったので，図3に示すビューを作成することにした。子供会には，生年月日が 20030402 ～ 20090401 の会員が所属する。次の SQL 文の［　　］に入れる正しい答えを，解答群の中から選べ。

子供会表

会員番号	世帯番号	電話番号	氏名	生年月日
0414	0181	999-999-9999	情報一郎	20040613
0415	0181	999-999-9999	情報二郎	20070105
⋮	⋮	⋮	⋮	⋮

図3　子供会表（ビュー）の構成とデータの表示例

```
CREATE VIEW 子供会表 AS
   SELECT 会員表.会員番号, 会員表.世帯番号,
          世帯表.電話番号, 会員表.氏名, 会員表.生年月日
     FROM 世帯表, 会員表
     WHERE [        d        ]
```

解答群

ア　会員表.会員番号 = ANY(SELECT 会員表.会員番号 FROM 会員表
　　WHERE 会員表.生年月日 IN(20030402, 20090401))

イ　会員表.会員番号 = 世帯表.世帯主番号 AND
　　会員表.生年月日 BETWEEN 20030402 AND 20090401

ウ　会員表.世帯番号 = ANY(SELECT 会員表.世帯番号 FROM 会員表
　　WHERE 会員表.生年月日 IN(20030402, 20090401))

エ　会員表.世帯番号 = 世帯表.世帯番号 AND
　　会員表.生年月日 BETWEEN 20030402 AND 20090401

問4　ホスト名の衝突に関する次の記述を読んで，設問1，2に答えよ。

DNS（Domain Name System）は，ルートサーバを頂点とする多数のDNSサーバから成る階層的な分散型データベースシステムであり，ホスト名とIPアドレスの変換に使用される。

A社は，一般に通用している自社ドメイン“example.co.jp”の他に，正式なTLD（最上位のドメイン）として運用されていない“corp”を，自社のネットワークだけで通用する独自のTLDとして使っている。自社ネットワークに設置しているDNSサーバで，これら二つのドメインのホスト名とIPアドレスの対応を管理している。

A社のDNSサーバで管理しているホスト名を，表1に示す。

表1　A社のDNSサーバで管理しているホスト名

mail.example.co.jp
www.example.co.jp
www.bunkyo.example.co.jp
www.minato.example.co.jp
mail.corp
www.corp

ホスト名に対応するIPアドレスを知りたいアプリケーションは，DNSリゾルバ（各端末で動作し，DNSサーバにIPアドレスの問合せを行うプログラム）に，ホスト名に対応するIPアドレスを問い合わせる。問合せを受けたDNSリゾルバは，A社のDNSサーバに問い合わせる。

問合せを受けたDNSサーバは，問い合わせられたホスト名が自分で管理しているホスト名であれば対応するIPアドレスを返す。そうでなければ外部のDNSサーバに問い合わせ，その結果として得られたIPアドレス，又は見つからなかったことを示すエラーのいずれかを返す（図1）。

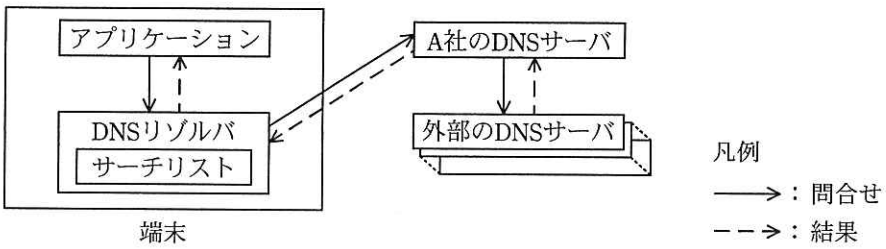


図1　IPアドレスの問合せ

A社では，ドメイン名を補完できるように，各端末のDNSリゾルバの設定で，A社のドメイン“example.co.jp”をサーチリストに登録することを指示している。このため，DNSリゾルバに問合せを行うアプリケーションは，ドメイン名“example.co.jp”を省略しても，ホスト名に対応するIPアドレスを得ることができる。

例えば，ホスト名“www.bunkyo.example.co.jp”のIPアドレスを知りたいときは，ホスト名“www.bunkyo”で問い合わせれば，ホスト名“www.bunkyo.example.co.jp”のIPアドレスが，次のようにして得られる（図2）。ここで，“bunkyo”というTLDは存在しないものとする。

① アプリケーションが，DNSリゾルバにホスト名“www.bunkyo”のIPアドレスを問い合わせる。

② DNSリゾルバは，A社のDNSサーバにホスト名“www.bunkyo”のIPアドレスを問い合わせる。

③ ホスト名“www.bunkyo”に対応するIPアドレスが見つからないので，A社のDNSサーバは，エラーをDNSリゾルバに返す。

④ エラーを受け取ったDNSリゾルバは，ホスト名“www.bunkyo”に，サーチリストに登録されたドメイン“example.co.jp”を連結したホスト名“www.bunkyo.example.co.jp”のIPアドレスをA社のDNSサーバに問い合わせる。

⑤ A社のDNSサーバは，“www.bunkyo.example.co.jp”のIPアドレスをDNSリゾルバに返す。

⑥ DNSリゾルバは，A社のDNSサーバから返された“www.bunkyo.example.co.jp”のIPアドレスをアプリケーションに返す。

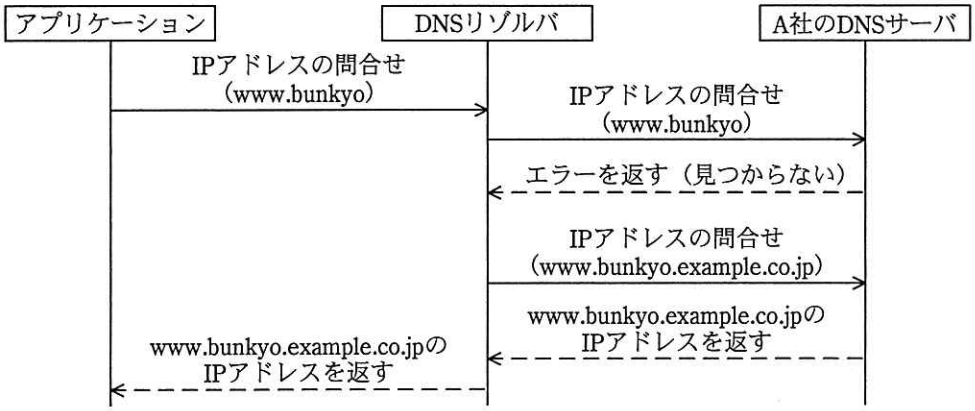


図2　サーチリストを利用したIPアドレスの問合せ

近年，多数の正式なTLDが追加されていて，これまでになかった新しいTLDが運用されることによって，名前が衝突するリスクが高まっている。

例えば，新しい正式なTLDとして，“bunkyo”が追加され，インターネット上でホスト名“www.bunkyo”のWebサーバの運用が開始されたとき，A社内の端末のDNSリゾルバに，ホスト名“www.bunkyo”を問い合わせると，［　a　］が返される。

また，新しく正式なTLDとして，“corp”が追加され，インターネット上でホスト名“www.corp”のWebサーバの運用が開始されたとき，A社内の端末のDNSリゾルバを使っても［　b　］のIPアドレスを得ることができない。

名前の衝突が起こることによって，本来アクセスしたいサーバにアクセスできないだけではなく，［　c　］ことによる情報漏えいなど，セキュリティ上のリスクが発生する。

設問1　本文中の［　　］に入れる適切な答えを，解答群の中から選べ。

aに関する解答群

ア　A社内のホスト“www.bunkyo.example.co.jp”のIPアドレス

イ　A社のDNSサーバのIPアドレス

ウ　インターネット上のWebサーバ“www.bunkyo”のIPアドレス

エ　見つからなかったことを示すエラー

bに関する解答群

ア　“www.corp”で始まるTLDが“corp”以外のホスト

イ　A社内のホスト“www.corp”

ウ　A社内のホスト“www.corp”，及びインターネット上のWebサーバ“www.corp”

エ　インターネット上のWebサーバ“www.corp”

cに関する解答群

ア　ウイルスが混入した電子メールを受信する

イ　外部サーバに意図せずアクセスする

ウ　組織内の人間が機密情報を意図的に流出させる

エ　ファイアウォールで守られたネットワークへの侵入を許す

設問2　新たなTLDが追加されることによって生じる，名前が衝突するリスクを低減させる対策として**適切でない答え**を，解答群の中から選べ。

解答群

ア　各組織独自のTLDの利用を停止する。

イ　各組織は，自組織のDNSサーバと外部のDNSサーバとの通信を遮断する。

ウ　各端末でサーチリストの利用をやめる。

問5　営業支援システムに関する次の記述を読んで，設問1，2に答えよ。

生命保険会社の E 社は，保険の新規契約の手続業務，既契約の保全手続業務（顧客の住所の変更，及び契約の解約）をサポートする営業社員のためのシステム（以下，営業支援システムという）を開発することになった。

営業支援システムは，新規契約の申込受付，既契約の保全手続受付，及び帳票出力の機能から成る。営業支援システムの各機能の要件を次に，E-R 図を図1に，機能階層図を図2に示す。

〔新規契約の申込受付機能の要件〕

顧客が新規に生命保険契約を締結するときは，生命保険契約の申込書に所定の内容を記入する。この申込書に記入された内容から，当該契約を担当する E 社の営業社員が，必要な情報を営業支援システムに登録する。

(1)　契約情報登録

契約情報を登録する。登録時に，契約ごとに一意な契約番号を付与する。

(2)　契約担当情報登録

当該契約の契約担当情報を登録する。契約は E 社に在籍する営業社員が担当し，主担当と副担当の2人になることもある。

(3)　顧客情報登録・更新

営業支援システムは，顧客1人の情報は1レコードで管理する。しかし，複数の契約と関係する顧客については，1 人の顧客を複数の異なる顧客情報で管理してしまう場合がある。その例を，表1に示す。

表1　1人の顧客を複数の異なる顧客情報で管理してしまう例

契約	顧客氏名	生年月日	性別	住所
2010年に締結した契約	情報　太朗	1975/11/23	男	XX県YY市ZZ区2-28-8
2014年に締結した契約	情報　太朗	1975/11/23	男	AA県BB市CC区4-1

注記　2013年に転居をしたあと，住所変更の手続を忘れたまま（下線部），転居後の住所で新たな契約を締結したので，複数の異なる顧客情報で管理されている。

営業支援システムでは，顧客情報を一元管理するために，新規契約の申込受付時，顧客の住所の変更受付時に，“顧客氏名”，“生年月日”，“性別”，“住所”を比較項

目（以下，名寄せ項目という）として，全ての名寄せ項目が同一である顧客情報が既に登録されているかどうかを調べることにしている。

新規契約の申込受付時の要件は次のとおりである。

① 当該契約の契約者，被保険者及び受取人の3名を顧客として，顧客情報を登録する。既に全ての名寄せ項目が同一である顧客情報が登録されている人物については，新たに登録しない。

② 顧客情報の登録時には，顧客ごとに一意な顧客IDを付与する。

③ 顧客情報の一つに，その顧客が契約者として関わる契約の件数を管理する保有契約の件数がある。契約者の顧客情報の登録時には1件と登録し，更新時には1をプラスする。

④ 契約者，被保険者及び受取人の3名の顧客情報ごとに，契約情報との関係を示す，関係顧客情報を登録する。

〔既契約の保全手続受付機能の要件〕

顧客から，既契約に対する保全手続の依頼を受け付けたとき，当該契約を担当するE社の営業社員が，保全手続に必要な情報を営業支援システムに登録する。

(1) 保全手続情報登録

保全手続に必要な情報の登録時に，保全手続ごとに一意な保全受付番号を付与する。保全手続情報は，契約単位に管理する。また，一つの契約に対する保全手続は，1日1回まで受け付ける。

(2) 契約情報更新

契約の解約を受け付けたときは，契約情報の一つである契約状況を“解約”に更新する。

(3) 顧客情報更新

① 登録された保全手続情報から，顧客情報を更新する。

② 顧客の住所の変更を受け付けたときは，変更後の顧客情報の全ての名寄せ項目と一致する顧客情報が既に登録されているかどうかを調べる。見つかった場合は，見つかった顧客情報を削除する。さらに，削除した顧客情報に関係する関係顧客情報を全て削除し，変更後の顧客情報によって新たに関係顧客情報を登録する。また，削除・登録した関係顧客情報に，契約者としての関係が含まれる場合は，

当該顧客情報の保有契約の件数を更新する。

③　契約の解約を受け付けたときは，契約者の顧客情報の保有契約の件数を更新（1をマイナス）する。また，関係顧客情報は変更しない。

〔帳票出力機能の要件〕

契約者に送付するための保全手続確認書を出力する。ここで，新規契約に関する帳票出力については省略する。

(1)　帳票印刷

登録されている保全手続情報から，保全手続確認書を作成して印刷する。印刷時に，帳票ごとに一意な帳票出力番号を付与する。

(2)　保全手続情報更新

帳票出力が完了した保全手続情報の工程ステータスを帳票出力済みに更新する。

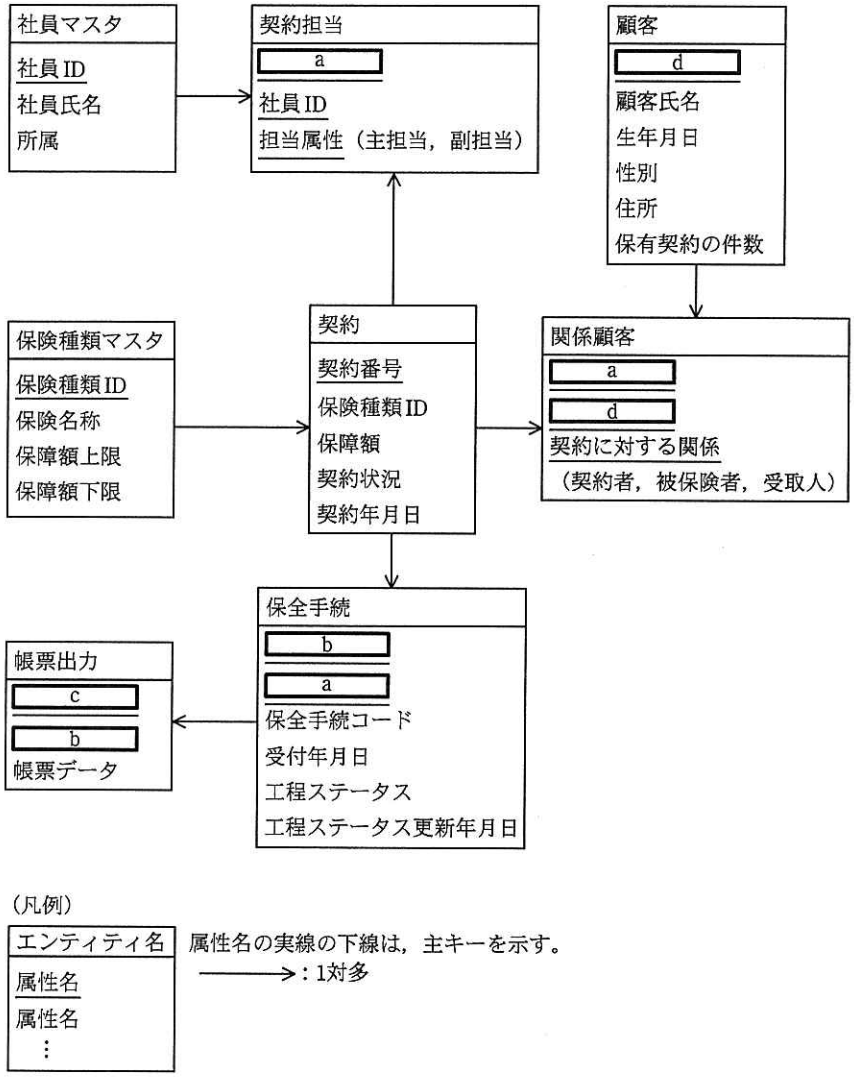


図１　営業支援システムの E-R 図

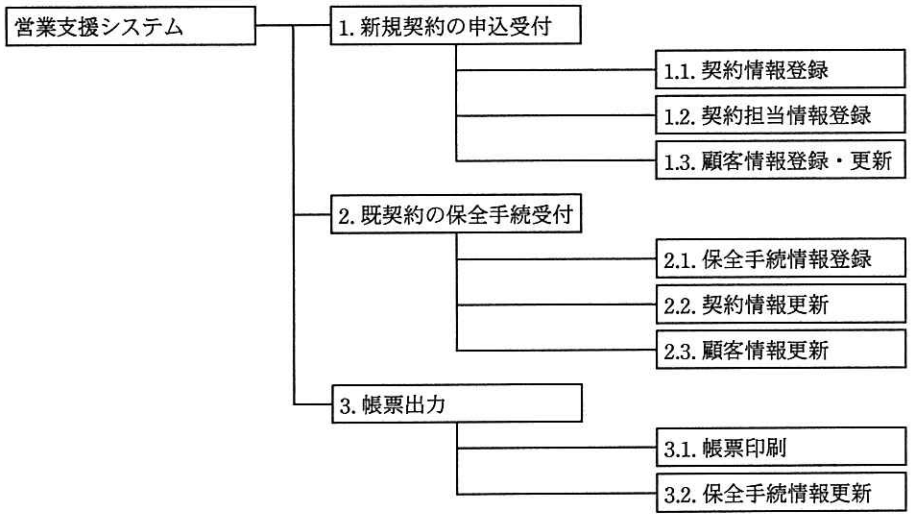


図2　営業支援システムの機能階層図

設問1　図1中の［　　］に入れる適切な属性名を，解答群の中から選べ。

a～dに関する解答群

ア　関係顧客番号　　イ　契約担当番号　　ウ　契約番号　　エ　顧客ID

オ　社員ID　　カ　帳票出力番号　　キ　保険種類ID　　ク　保全受付番号

設問2　営業支援システムの E-R 図のエンティティと機能の関係を表 2 に示す。表 2 中の［　　］に入れる正しい答えを，解答群の中から選べ。

表2　営業支援システムの E-R 図のエンティティと機能の関係

	契約情報登録	契約担当情報登録	顧客情報登録・更新	保全手続情報登録	契約情報更新	顧客情報更新	帳票印刷	保全手続情報更新
社員マスタ	—	R	—	—	—	—	R	—
契約担当	—	C	—	—	—	—	R	—
契約	C	R	R	R	R, U	—	R	—
顧客	—	—	e	—	—	R, U, D	R	—
関係顧客	—	—	C	—	—	f	R	—
保全手続	—	—	—	C, R	R	R	R	U
帳票出力	—	—	—	—	—	—	C	—
保険種類マスタ	R	—	—	—	—	—	R	—

注記　C は登録（Create），R は参照（Read），U は参照して更新（Update），D は削除（Delete）を示す。複数の操作が該当する欄には，コンマ（,）で区切って記入している。ただし，機能を実行する際にエンティティに対して何も操作しない場合は，—（ハイフン）を記入している。

e，f に関する解答群

ア　C　　イ　R　　ウ　U　　エ　D

オ　C, D　　カ　C, R, U　　キ　C, R, D　　ク　C, U, D

ケ　—

問6　プロジェクトにおけるコミュニケーションの計画に関する次の記述を読んで，設問1，2に答えよ。

B社で立ち上げた社内システムの刷新プロジェクトでは，関係者間における情報の配布を考慮したコミュニケーションの計画を立案することにした。関係者とは，B社の情報システム部及びベンダL社である。このプロジェクトは，作業を効率よく進めることを目的に，プロジェクトマネージャ（以下，PMという）を頂点とするプロジェクト体制を構築することにした。また，計画段階における認識のずれを防止するために，社内の各業務の主管部門の要求事項を，システム化対象の機能要件として，プロジェクト計画書に盛り込むことにした。プロジェクトの体制を，図1に示す。

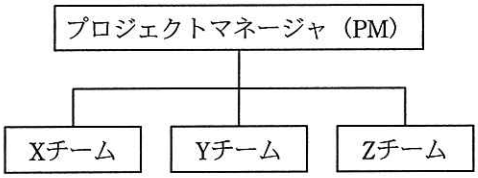


図1　社内システムの刷新プロジェクトの体制

〔各チームの構成〕

(1) Xチームは，社内システムの処理Xを担当する情報システム部員で構成される。

(2) Yチームは，社内システムの処理Yを担当する情報システム部員で構成される。

(3) Zチームは，社内システムの処理Zを担当するベンダL社の社員で構成される。

(4) 各チームのメンバは，チームリーダ（以下，TLという）と担当者から構成される。各チームのメンバの人数を，表1に示す。

表1　各チームのメンバの人数

チーム名	Xチーム	Yチーム	Zチーム
TL（人）	1	1	1
担当者（人）	2	3	4

E
後

〔コミュニケーションの計画の説明〕

プロジェクト計画書に記載されているコミュニケーションの計画の一部を，図2に示す。

2. 計画

2.1 プロジェクト管理文書

項番	文書名	管理者	配布先	配布方法
1	プロジェクト計画書	PM	全チームの TL	全体進捗会議において
2	プロジェクト実績報告書	PM	全チームの TL	全体進捗会議において
3	プロジェクト会議議事録	PM	全チームのメンバ	電子メール
4	設計書	PM	各チームのメンバ	ファイル閲覧
5	設計検討会議事録	各チームの TL	各チームの担当者	電子メール
6	仕様問合せ票	各チームの TL	各チームの担当者	電子メール

注記 プロジェクト管理文書の登録及び配布，並びに配布した履歴の保管は，管理者が行う。

2.2 プロジェクト会議

項番	会議名	開催主幹	開催頻度	参加者
1	全体進捗会議	PM	月2回	PM，全チームの TL
2	チーム進捗会議	PM	チームごとに月4回	PM，各チームのメンバ

2.3 仕様関連会議

項番	会議名	開催主幹	開催頻度	参加者
1	設計検討会	各チームの TL	随時	各チームのメンバ
2	設計書レビュー	各チームの TL	随時	PM，各チームのメンバ

図2 コミュニケーションの計画の一部

設問1 定期的に開催されるプロジェクト会議のプロジェクト会議議事録（以下，議事録という）を配布する際のルールについて，PM が行った検討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。

〔PMが行った検討作業〕

コミュニケーションの計画に定められた方法に従って，情報をより正確に配布できるルールの検討に当たり，コミュニケーションを取る関係者間の関係の数を明らかにした。関係の数は，ある作業において，コミュニケーションを取る二者間の関係を1と数えて，作業の関係者間の全ての関係を数えたものである。関係の数の例を，図3に示す。

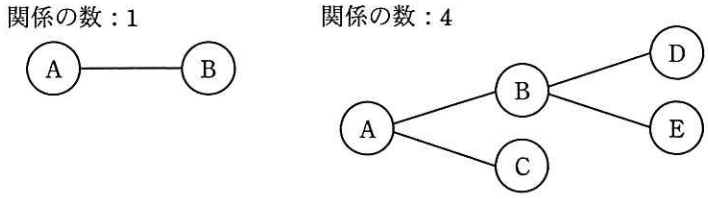


図3　関係の数の例

(1)　議事録は，［　a　］が配布する。議事録の配布の頻度は，定期的に開催されるプロジェクト会議の開催頻度から月［　b　］回となる。議事録の配布については，各チームの TL を議事録の配布の窓口とし，TL から担当者に配布する方法（以下，方法①という）と，［　a　］から全チームのメンバに直接配布する方法（以下，方法②という）を検討した。方法①の場合，議事録を配布するための関係の数は，［　a　］と各チームの TL の間の数と，各チームの TL と担当者の間の数との合計なので，［　c　］となる。方法②と比較して，関係の数は［　d　］が，各チーム内で TL が窓口として担当者を束ねることになるので，コミュニケーションを密に取ることができる。そこで，方法①を選択することにした。

(2)　方法①で，議事録の内容が配布先に確かに伝わったことを確認し，その報告の履歴を残すために，配布先の全ての TL に，［　e　］よう指示することにした。

aに関する解答群

ア　PM　　イ　Xチームの TL　　ウ　Yチームの TL

エ　Zチームの TL

bに関する解答群

ア 2　　イ 6　　ウ 12　　エ 14

cに関する解答群

ア 3　　イ 6　　ウ 12　　エ 24

dに関する解答群

ア 多い　　イ 少ない　　ウ 変わらない

eに関する解答群

ア TLが担当者全員から内容を確認した旨の電子メールを受け，TLからPMに口頭で報告する

イ TLが担当者全員から内容を確認した旨の電子メールを受け，TLからPMに電子メールで報告する

ウ メンバが，内容を確認した旨を直接PMに電子メールで報告する

設問2 設計工程の状況に基づく，グループウェア（以下，GWという）の導入についての記述中の［　　］に入れる適切な答えを，解答群の中から選べ。

〔設計工程の状況とGWの導入について〕

設計工程の進行に伴い，各チームの進捗に遅延が発生し始めた。原因は，各チームの担当者が，出席を要請されている設計検討会に十分に参加できていないことと，プロジェクト関係者間の情報流通が悪いことであった。

PMは，メンバのスケジュールを確保した上で今後の設計検討会を実施するとともに，プロジェクト関係者間の情報流通を改善することに着手した。情報流通として，プロジェクト計画書の配布先が原因となり，システム化対象の機能要件を十分に確認しないまま設計したことと，機能要件の変更がメンバに伝わっていない場合があることに問題があると分析している。解決に向け，GWの導入を踏まえた検討を行った。

プロジェクト計画書については，現状では配布先が限定されているので，GW

導入後の配布先を［ f ］とした。プロジェクト管理文書の配布に関するGWの設定ルールの検討結果を表2に整理した。

表2　GWの設定ルールの検討結果

項番	文書名	PM	Xチーム		Yチーム		Zチーム	
			TL	担当者	TL	担当者	TL	担当者
1	プロジェクト計画書							
2	プロジェクト実績報告書	●	○	×	○	×	○	×
3	プロジェクト会議議事録	g						
4	設計書							
5	設計検討会議事録		●	○	●	○	●	○
6	仕様問合せ票		●	○	●	○	●	○

注記　網掛けの部分は表示していない。
表中で使用している記号の意味は，次のとおりである。
●：プロジェクト管理文書の登録及び参照を許可
○：プロジェクト管理文書の参照だけ許可
×：プロジェクト管理文書の参照不可

fに関する解答群

ア　全チームの担当者　　イ　全チームのメンバ

gに関する解答群

	PM	Xチーム		Yチーム		Zチーム	
		TL	担当者	TL	担当者	TL	担当者
ア	○	○	○	○	○	○	○
イ	○	●	○	●	○	●	○
ウ	●	○	○	○	○	○	○
エ	●	●	●	●	●	●	●

問7　システム開発の投資評価に関する次の記述を読んで，設問1，2に答えよ。

T社では，営業支援システムの開発プロジェクトを検討している。開発方式が異なるA案，B案の二つの案があり，それぞれの投資評価を行うことにした。二つの案の投資対効果の概要を表1に，キャッシュフローの年度別の推移を表2に示す。

現時点を平成27年度とし，営業支援システムは平成28年度から運用され，5年間にわたって効果が得られる。また，投資額及び効果額の発生は年度末とする。

表1　二つの案の投資対効果の概要

単位　万円

		A案		B案	
投資額	初期開発費	（平成27年度に発生）	15,000	（平成27年度に発生）	10,000
	追加開発費		0	（平成30年度に発生）	4,000
	年間運用費	（平成28～32年度）	800	（平成28～30年度） （平成31，32年度）	600 800
効果額	各年度での効果額	（平成28～32年度）	5,000	（平成28～30年度） （平成31，32年度）	3,700 5,000

表2　キャッシュフローの年度別の推移

単位　万円

		平成27年度（現在）	平成28年度（1年目）	平成29年度（2年目）	平成30年度（3年目）	平成31年度（4年目）	平成32年度（5年目）
A案	キャッシュアウト	15,000	800	800	800	800	800
	キャッシュイン		5,000	5,000	5,000	5,000	5,000
	キャッシュフロー	−15,000	4,200	4,200	4,200	4,200	4,200
B案	キャッシュアウト	10,000	600	600	4,600	800	800
	キャッシュイン		3,700	3,700	3,700	5,000	5,000
	キャッシュフロー	−10,000	3,100	3,100	−900	4,200	4,200

設問1　システム開発の投資評価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。

従来採用している投資評価方法は，キャッシュフローの累計額がプラスになるまでの期間（以下，投資回収期間という）を求め，その期間の長さによって投資の判断を行う方法である。

この方法では，A案の投資回収期間は4年，B案の投資回収期間は［　a　］年であり，［　b　］ことになる。

aに関する解答群

ア　3　　イ　4　　ウ　5　　エ　6

bに関する解答群

ア　A案よりもB案の方が回収期間は長く，A案を採用する

イ　A案よりもB案の方が回収期間は長く，B案を採用する

ウ　A案よりもB案の方が回収期間は短く，A案を採用する

エ　A案よりもB案の方が回収期間は短く，B案を採用する

オ　投資回収期間はA案もB案も同じなので，この方法だけでは判断できない

設問2　システム開発の投資評価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。

T社の企画課は，システム投資のリスクを低減するために，いきなり開発の投資判断を行うのではなく，平成27年度に，初期開発費とは別に100万円を投資してシステムの要求分析を行って投資評価の精度を高めることにした。

また，投資評価方法に正味現在価値（以下，NPVという）を用いることにし，NPVが大きい案を採用することにした。ここで，

NPV＝キャッシュフローの現在価値の合計値

とし，千円の位を四捨五入して，万円単位まで求めることにする。

評価に当たり，現在価値の評価期間は平成 32 年度までの 5 年間とし，割引率は 10%とする。

要求分析の結果を反映させる前のシステム投資の評価は，表 3 に示すキャッシュフローの現在価値から，A 案の NPV が 818 万円，B 案の NPV が［ c ］万円であり，A 案を採用することになる。

キャッシュフローの現在価値は，n 年後のキャッシュフローが C，割引率が r のとき，$C \times \{1/(1+r)^n\}$ である。

計算には，次の値を用いる。

$1/(1+0.1)=0.91$　　$1/(1+0.1)^2=0.83$

$1/(1+0.1)^3=0.75$　　$1/(1+0.1)^4=0.68$

$1/(1+0.1)^5=0.62$

表 3　キャッシュフローの現在価値

単位　万円

		平成 27 年度（現在）	平成 28 年度（1 年目）	平成 29 年度（2 年目）	平成 30 年度（3 年目）	平成 31 年度（4 年目）	平成 32 年度（5 年目）
A 案	キャッシュアウト	15,100	800	800	800	800	800
	キャッシュイン		5,000	5,000	5,000	5,000	5,000
	キャッシュフロー	−15,100	4,200	4,200	4,200	4,200	4,200
	キャッシュフローの現在価値	−15,100	3,822			2,856	2,604
B 案	キャッシュアウト	10,100	600	600	4,600	800	800
	キャッシュイン		3,700	3,700	3,700	5,000	5,000
	キャッシュフロー	−10,100	3,100	3,100	−900	4,200	4,200
	キャッシュフローの現在価値	−10,100	2,821			2,856	2,604

注記　網掛けは表示していない。

要求分析の結果，A 案の初期開発費は 14,500 万円，B 案の追加開発費は 3,100 万円となることが分かった。

要求分析の後，この結果を用いてシステム投資の評価をやり直した結果，A 案の NPV は［ d ］万円，B 案の NPV は 754 万円となり，［ e ］ことになる。

c，d に関する解答群

ア 79　　イ 754　　ウ 1,318　　エ 1,418

e に関する解答群

ア A 案を採用する

イ A 案と B 案のどちらを採用しても同じ

ウ B 案を採用する

E 後

次の**問8**は必須問題です。必ず解答してください。

問8　次のプログラムの説明及びプログラムを読んで，設問1～3に答えよ。

与えられたn個のデータの中からk番目に小さい値を選択する方法として，クイックソートを応用したアルゴリズムを考える。クイックソートとは，n個のデータをある基準値以下の値のグループと基準値以上の値のグループに分割し（基準値はどちらのグループに入れても構わない），更にそれぞれのグループで基準値を選んで二つのグループに分割するという処理を繰り返してデータを整列するアルゴリズムである。クイックソートを応用してk番目に小さい値を選択するアルゴリズムでは，データを二つのグループに分割した時点で，求める値はどちらのグループに含まれるかが確定するので，そのグループだけに，更に分割する処理を繰り返し適用する。グループの分割ができなくなった時点で，k番目に小さい値が選択されている。

〔プログラムの説明〕

n個の数値が格納されている配列xと値kを与えて，k番目に小さい値を返す関数Selectである。ここで，配列xの要素番号は1から始まる。また，配列xの大きさは，配列に格納される数値の個数分だけ確保されているものとする。Selectの処理の流れを次に示す。

(1)　行番号3～4

k番目に小さい値を選択するために走査する範囲（以下，走査範囲という）の左端をTop，右端をLastとし，まず配列全体を走査範囲とする。

(2)　行番号5～32

①　走査範囲に含まれる要素の数が1以下になるまで，②，③を繰り返す。

②　基準値Pivotを選び，走査範囲内の値で基準値以下のものを左に，基準値以上のものを右に集める（行番号6～24）。

③　走査範囲が基準値以下の値から成るグループと基準値以上の値から成るグループに分割されるので，k番目に小さい値が含まれるグループを新たな走査範囲とする（行番号25～30）。

④ 繰返しが終了したときに，要素 x[k] の値が k 番目に小さい値として，選択される。

Select の引数と返却値の仕様は次のとおりである。

〔関数 Select の引数／返却値の仕様〕

引数名／返却値	データ型	入力／出力	意味
x[]	整数型	入力	数値が格納されている一次元配列
n	整数型	入力	数値の個数
k	整数型	入力	選択する数値の小ささの順位を示す値
返却値	整数型	出力	選択された数値

〔プログラム〕

（行番号）

```
 1 ○整数型: Select(整数型: x[], 整数型: n, 整数型: k)
 2 ○整数型: Top, Last, Pivot, i, j, work

 3 • Top ← 1                          /* 走査範囲の左端の初期値を設定 */
 4 • Last ← n                         /* 走査範囲の右端の初期値を設定 */
 5 ■ Top ＜ Last
 6   • Pivot ← x[k]
 7   • i ← Top                                                    ← α
 8   • j ← Last
 9   ■ true                           /* ループ */
10     ■ x[i] ＜ Pivot                                            ← β
11       • i ← i + 1
12     ■
13     ■ Pivot ＜ x[j]
14       • j ← j − 1
15     ■
16     ▲ i ≧ j
17     ▼ • break                      /* ループから抜ける */
18
19     • work ← x[i]
20     • x[i] ← x[j]
21     • x[j] ← work                                              ← γ
22     • i ← i + 1
23     • j ← j − 1
24   ■
25   ▲ i ≦ k
26   ▼ • Top ← j + 1
27
28   ▲ k ≦ j
29   ▼ • Last ← i − 1
30
31 ■
32 • return x[k]
```

設問1　関数 Select の追跡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。

関数 Select の引数で与えられた配列 x の要素番号 1 ～ 7 の内容が 3，5，6，4，7，2，1 であり，n が 7，k が 3 のとき，配列 x の走査範囲の左端 Top と右端 Last の値は次のとおりに変化する。

・Top と Last の初期値は，それぞれ 1 と 7 である。

・Top ＜ Last が成り立つ間，次に示す (1) 選択処理 1 回目の ① ～ ③，(2) 選択処理 2 回目の ① ～ ③，… と実行する。

(1)　選択処理 1 回目

①　配列 x の走査範囲を二つの部分に分ける基準値 Pivot に配列 x の 3 番目の要素 x[3] の値 6 を設定する。次に，i に Top の値 1，j に Lastの値 7 を設定する。

②　配列 x の Top から Last までの走査範囲内にある数値を，6 以下の数値のグループと 6 以上の数値のグループの二つに分ける処理を行う。その結果，配列 x の内容は次のとおりになる。

3，5，1，4，2，7，6

③　［　a　］を設定して選択処理の 2 回目に進む。

(2)　選択処理 2 回目

①　基準値 Pivot に x[3] の値 1 を設定する。

②　配列 x の Top から Last までの走査範囲内にある数値を，1 以下の数値のグループと 1 以上の数値のグループの二つに分ける処理を行う。その結果，配列 x の内容は次のとおりになる。

1，5，3，4，2，7，6

③　［　b　］を設定して選択処理の 3 回目に進む。

(3)　選択処理 3 回目

⋮

この選択処理を繰り返して，Top ＜ Last でなくなったときに処理を終了する。このとき，関数の返却値 x[k] には与えられた数値の中から k 番目に小さい値が選択されている。

a，b に関する解答群

ア　Top に値 1，Last に値 5　　イ　Top に値 1，Last に値 6

ウ　Top に値 2，Last に値 5　　エ　Top に値 2，Last に値 6

オ　Top に値 3，Last に値 5　　カ　Top に値 3，Last に値 6

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。

引数で与えられた配列 x の要素番号 1 ～ 7 の内容が 1，3，2，4，2，2，2 であり，n が 7，k が 3 のとき，選択処理が終了するまでにプログラム中の α の部分は［ c ］回実行され，γ の部分は［ d ］回実行される。

c，d に関する解答群

ア　1　　イ　2　　ウ　3　　エ　4　　オ　5　　カ　6

設問 3　次の記述中の［　　］に入れる正しい答えを，解答群の中から選べ。

プログラム中の β の行 x[i] ＜ Pivot を誤って x[i] ≦ Pivot とした。この場合，引数で与えられた配列 x の要素番号 1 ～ 6 の内容が 1，1，1，1，1，1 であり，n が 6，k が 3 のとき，［ e ］。また，引数で与えられた配列 x の要素番号 1 ～ 6 の内容が 1，3，2，4，2，2 であり，n が 6，k が 3 のとき，［ f ］。

e，f に関する解答群

ア　Last に値 0 が設定される　　イ　Pivot に値 0 が設定される

ウ　Top に値 0 が設定される　　エ　処理が終了しない

オ　配列の範囲を越えて参照する

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，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。

〔プログラムの説明〕

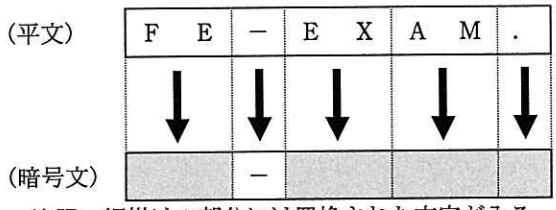
平文(ひらぶん)の文字列を暗号化する手法として，平文の文字を換字(かえじ)表に従って別の文字に置き換える方法がある。関数 enc_str は，与えられた平文を，換字表を用いて暗号文に変換するプログラムである。

(1)　平文は，JIS X 0201（7ビット及び8ビットの情報交換用符号化文字集合）の文字で構成されている。

(2)　置換の対象となる文字（以下，置換対象文字という）は，英字（A～Z，a～z），数字（0～9），空白文字，“.”及び“,”の65種類の文字であり，換字表に格納されている。

(3)　換字表は5行13列の2次元文字型配列であり，全ての要素に異なる文字が格納されている。

(4)　換字表による置換の方法について，図1に示す平文と暗号文の対応の例を用いて説明する。平文の文字列の先頭から置換対象文字を2文字ずつ選び出して行い，置換されたそれぞれの文字を暗号文の文字として，暗号文中で平文と同じ位置に入れる。ここで，置換対象文字の組合せによって，置換後の文字の組合せが決まる。置換対象文字数が奇数の場合，最後の置換対象文字については1文字で置換を行う。置換対象文字以外の文字については，平文中の文字をそのまま暗号文中で平文と同じ位置に入れる。



注記　網掛けの部分には置換された文字が入る。

図1　平文と暗号文の対応の例

(5) 関数 enc_str の仕様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。

機能： 文字列 str を，換字表 xchg_t を用いて平文から暗号文に変換する。

引数： str　　文字列

xchg_t　換字表

C

〔プログラム〕

```
#define RSIZ  5  /* 換字表の行数 */
#define CSIZ 13  /* 換字表の列数 */

void enc_str(char[], char[RSIZ][CSIZ]);

void enc_str(char str[], char xchg_t[RSIZ][CSIZ]) {
    int cp[2],  /* 置換対象文字の換字表中の列位置 */
        rp[2],  /* 置換対象文字の換字表中の行位置 */
        pos[2], /* 置換対象文字の文字列中の位置 */
        flg, i = 0, col, row, p = 0;

    while (str[p] != '\0') {
        flg = 0;
        for (row = 0; row < RSIZ; row++) {
            for (col = 0; col < CSIZ; col++) {
                if (str[p] == xchg_t[row][col]) {
                    flg = 1;
                    break;
                }
            }
            if (flg != 0) break;
        }
        if (flg != 0) {                                    α
            cp[i]    = col;
            rp[i]    = row;
            pos[i++] = p;
            if (i == 2) {
                if (str[pos[0]] == str[pos[1]]) {
                    str[pos[0]] = str[pos[1]] =
                                    xchg_t[(rp[0] + 1) % RSIZ]
                                          [(cp[0] + 1) % CSIZ];
                } else {
                    if (rp[0] == rp[1]) {
                        str[pos[0]] = xchg_t[rp[1]][cp[1]];
                        str[pos[1]] = xchg_t[rp[0]][cp[0]];
                    } else if (cp[0] == cp[1]) {
                        str[pos[0]] = xchg_t[rp[0]]
                                            [(cp[0] + 1) % CSIZ];
                        str[pos[1]] = xchg_t[rp[1]]
                                            [(cp[1] + 1) % CSIZ];
                    } else {
                        str[pos[0]] = xchg_t[rp[1]][cp[0]];
                        str[pos[1]] = xchg_t[rp[0]][cp[1]];
                    }
                }
                i = 0;                                     β
            }
        }
        p++;
    }
    if (i != 0) {
        str[pos[0]] = xchg_t[RSIZ - 1 - rp[0]][CSIZ - 1 - cp[0]];
    }
}
```

設問1　図2の平文と換字表を引数として関数 enc_str を実行したとき，プログラムの動作に関する次の記述中の［　　］に入れる正しい答えを，解答群の中から選べ。

文字列［位置］

```
          0         10        20   25
          +---------+---------+----+
(平文)    FunctionΔΔf(x,y)ΔisΔgiven.
```

換字表[行位置][列位置]

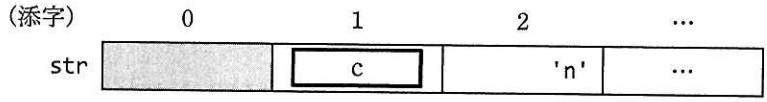
	0	1	2	3	4	5	6	7	8	9	10	11	12
0	a	b	c	d	e	f	g	h	i	j	k	l	m
1	n	o	p	q	r	s	t	u	v	w	x	y	z
2	0	1	2	3	4	5	6	7	8	9	,	.	Δ
3	Z	Y	X	W	V	U	T	S	R	Q	P	O	N
4	M	L	K	J	I	H	G	F	E	D	C	B	A

注記　“Δ”は空白文字を表す。

図2　平文と換字表の例

(1)　プログラムの**α**の行が最初に実行されるとき，変数 flg の値は 1，i の値は 0，col の値は［　a　］，row の値は［　b　］になっている。

(2)　プログラムの**β**の行が最初に実行されるとき，引数 str，配列 cp，rp，pos に格納されている値は，図3に示すとおりである。



(添字)

	0	1
cp	a	7
rp	b	1
pos	0	1

注記　網掛けの部分は表示していない。

図3　引数 str，配列 cp，rp，pos に格納されている値

(3) プログラムのβの行が 5 回目に実行されるとき，pos[1]の値は 9，引数 str[9]の値は［ d ］になっている。

(4) プログラムのβの行が 6 回目に実行されるとき，pos[1]の値は［ e ］になっている。

a，bに関する解答群

ア 1　　イ 2　　ウ 3　　エ 4　　オ 5

カ 6　　キ 7　　ク 8　　ケ 9　　コ 10

cに関する解答群

ア 'F'　　イ 'S'　　ウ 't'

エ 'v'　　オ '7'

dに関する解答群

ア 'n'　　イ '0'　　ウ 'y'

エ 'Z'　　オ 空白文字

eに関する解答群

ア 11　　イ 12　　ウ 13　　エ 14　　オ 15

設問 2　図 2 に示す換字表を使って平文“IPA”を暗号文に変換した結果として正しい答えを，解答群の中から選べ。

解答群

ア CVa　　イ iAP　　ウ iCN

エ iNC　　オ PIa　　カ VCa

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〕

C社では，全従業員の出社時刻と退社時刻を入口のゲートで自動的に記録し，出退勤記録ファイルに格納している。このプログラムは，1か月分の出退勤記録ファイルを読み込み，出社した従業員の勤務管理ファイルに書き込むものである。

(1) 出退勤記録ファイルは，図1に示すレコード様式の順ファイルで，ある月の1か月分の出退勤記録が格納されている。

日付 8桁	時刻 4桁	出退勤フラグ 1桁	従業員番号 6桁

図1 出退勤記録ファイルのレコード様式

① レコードは日付，時刻，出退勤フラグの昇順に格納されている。

② 日付には，年，月，日が，それぞれ4桁，2桁，2桁の西暦で格納されている。

③ 時刻には，出社又は退社の時刻の時，分が，それぞれ2桁の24時間表記で格納されている。

④ 出退勤フラグには，出社時は0が，退社時は1が格納されている。

⑤ 従業員番号には，従業員ごとに一意な6桁の番号が格納されている。

⑥ 同一従業員の出社時の記録と退社時の記録は，必ず対になっている。同じ日付のレコードは1組だけである。

⑦ 出社時刻は6時から22時までである。退社時刻は同じ日付の24時までである。

(2) 勤務管理ファイルは，図2に示すレコード様式の従業員番号をキーとする索引ファイルである。

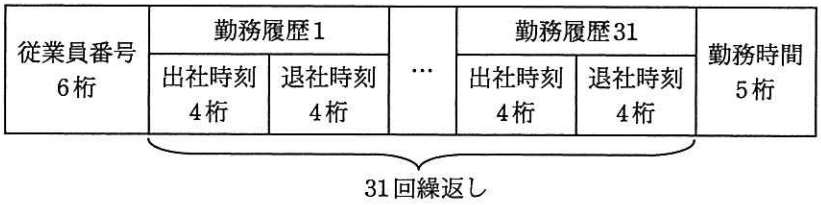


図2　勤務管理ファイルのレコード様式

① 勤務履歴1～31には，当該月の1～31日の出社時刻及び退社時刻が格納される。出社しなかった日と暦にない日の出社時刻及び退社時刻には，9999が格納される。

② 出社時刻及び退社時刻には，時，分が，それぞれ2桁の24時間表記で格納される。

③ 勤務時間には，1か月の勤務時間の合計が，3桁の時間と2桁の分で格納される。毎日の出社時刻から退社時刻までの時間が1日の勤務時間である。途中の休み時間も勤務時間に含める。

〔プログラム〕

```
(行番号)
 1  DATA DIVISION.
 2  FILE SECTION.
 3  FD LOG-FILE.
 4  01 LOG-REC.
 5     02 LOG-DATE    PIC 9(8).
 6     02 LOG-TIME    PIC 9(4).
 7     02 LOG-FLAG    PIC 9(1).
 8        88 LOG-IN   VALUE 0.
 9        88 LOG-OUT  VALUE 1.
10     02 LOG-ENUM    PIC X(6).
11  FD ATD-FILE.
12  01 ATD-REC.
13     02 ATD-ENUM       PIC X(6).
14     02 ATD-HST.
15        03 ATD-ELM     OCCURS 31.
16           04 ATD-IN   PIC 9(4).
17           04 ATD-OUT  PIC 9(4).
18     02 ATD-SUM        PIC 9(5).
```

```
19  WORKING-STORAGE SECTION.
20  77 KEY-FLAG        PIC 9(1).
21     88 ATD-VALID    VALUE 0.
22     88 ATD-INVALID  VALUE 1.
23  77 EOF-FLAG    PIC 9(1) VALUE 0.
24     88 LOG-EOF  VALUE 1.
25  01 W-DATE.
26     02 FILLER  [    a    ].
27     02 IDX     PIC 9(2).
28  01 W-TIME-FROM.
29     02 HF  PIC 9(2).
30     02 MF  PIC 9(2).
31  01 W-TIME-TO.
32     02 HT  PIC 9(2).
33     02 MT  PIC 9(2).
34  01 W-TIME-ACCUM.
35     02 HA  [    b    ].
36     02 MA  PIC 9(2).
37  77 W-MINUTE  PIC 9(5).
38  PROCEDURE DIVISION.
39  MAIN-PROC.
40      OPEN INPUT LOG-FILE
41           I-O   ATD-FILE.
42      PERFORM READ-PROC.
43      CLOSE ATD-FILE LOG-FILE.
44      STOP RUN.
45  READ-PROC.
46      PERFORM UNTIL LOG-EOF
47         READ LOG-FILE
48            AT END
49               SET LOG-EOF TO TRUE
50            NOT AT END
51               PERFORM UPDATE-PROC
52         END-READ
53      END-PERFORM.
54  UPDATE-PROC.
55      MOVE LOG-ENUM TO ATD-ENUM.
56      SET ATD-VALID TO TRUE.
57      READ ATD-FILE
58         INVALID KEY
59            SET ATD-INVALID TO TRUE
60            MOVE LOG-ENUM TO ATD-ENUM
61            [              c              ]
62            MOVE ZERO TO ATD-SUM
63      END-READ.
```

COBOL

```
64      MOVE LOG-DATE TO W-DATE.
65      EVALUATE TRUE
66         WHEN LOG-IN
67            MOVE LOG-TIME    TO ATD-IN(IDX)
68         WHEN LOG-OUT
69            MOVE LOG-TIME    TO ATD-OUT(IDX) W-TIME-TO
70            MOVE ATD-IN(IDX) TO              W-TIME-FROM
71            MOVE ATD-SUM     TO              W-TIME-ACCUM
72            PERFORM CALC-PROC
73            MOVE [    d    ] TO ATD-SUM
74      END-EVALUATE.
75      IF ATD-VALID THEN
76         REWRITE ATD-REC
77      ELSE
78         WRITE ATD-REC
79      END-IF.
80  CALC-PROC.
81      COMPUTE W-MINUTE = (HT - HF + HA) * 60 + MT - MF + MA.
82      DIVIDE W-MINUTE BY 60 GIVING HA REMAINDER MA.
```

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　PIC 9(2)　　イ　PIC 9(3)　　ウ　PIC 9(4)

エ　PIC 9(5)　　オ　PIC 9(6)

cに関する解答群

ア　INITIALIZE ATD-HST

イ　INITIALIZE ATD-HST REPLACING NUMERIC BY 9999

ウ　MOVE 9999 TO ATD-HST

エ　MOVE ALL ZERO TO ATD-HST

dに関する解答群

ア　ATD-OUT(IDX)　イ　HA　　ウ　W-MINUTE　　エ　W-TIME-ACCUM

設問2　C社では22時以降の勤務を深夜手当の対象としている。勤務管理ファイルのレコードに深夜勤務時間の領域を追加し，1か月の深夜勤務時間の合計も集計するようにプログラムを変更する。表1中の［　　］に入れる正しい答えを，解答群の中から選べ。ここで，表1中の［　d　］には設問1の正しい答えが入っているものとする。

表1　プログラムの変更内容

処置	変更内容
行番号18と19の間に追加	02 ATD-NIGHTSUM PIC 9(5).
行番号19と20の間に追加	77 NIGHT-TIME PIC 9(4) VALUE 2200.
行番号62を変更	MOVE ZERO TO ATD-SUM ATD-NIGHTSUM
行番号73と74の間に追加	IF ATD-OUT(IDX) > NIGHT-TIME THEN 　MOVE［　e　］TO W-TIME-FROM 　MOVE［　f　］TO W-TIME-ACCUM 　PERFORM CALC-PROC 　MOVE［　d　］TO ATD-NIGHTSUM END-IF

e，fに関する解答群

ア　ATD-IN(IDX)　　イ　ATD-NIGHTSUM　　ウ　ATD-OUT(IDX)

エ　ATD-SUM　　オ　NIGHT-TIME

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〕

D社では，利用者が入力した文字列をHTMLに埋め込むことによって，動的にWebページを生成するアプリケーションを開発している。

このようなアプリケーションでは，利用者が入力した文字列をHTMLに埋め込む際の処理が不適切な場合，情報漏えいの発生などシステムの安全な運用を脅かすおそれがある。

利用者が入力した文字列をHTMLに埋め込む前には，サニタイズすることが必要である。サニタイズとは，システムの安全な運用を脅かす文字列を，無害な文字列に変換することをいう。文字列の変換方法は，文字列を埋め込む場所によって異なる。

当該アプリケーションでは，HTMLのタグの間（ただし，“<script>”と“</script>”との間は除く）に文字列を埋め込むケースと，JavaScriptの文字列として埋め込むケースの二つのケースでサニタイズする必要がある。サニタイズは，変換前の文字列中の各文字を，それぞれ次のように変換することによって行うことにした。

(1)　HTMLのタグの間に文字列を埋め込むケース

①　英数字と次に示す文字は，そのまま出力する。

```
!#$%()*+,-.:;=?@[\]^_`{|}~
```

②　表1に示す文字は，実体参照に変換する。

表1　文字と実体参照の対応

文字	実体参照
<	<
>	>
&	&
"	"

③　その他の文字は，“&#ddd;”に変換する。ここで，dddは変換対象文字の文字コードを表す，最大で5桁の10進数である。また，変換前の文字列に含まれる各文字は，2バイトで表すことができるものとする。

(2) JavaScript の文字列として埋め込むケース

① 英数字は，そのまま出力する。

② 文字コードが 256 未満の英数字以外の文字は，“\xXX”に変換する。ここで，XX は変換対象の文字の文字コードを表す，2 桁の 16 進数である。

③ その他の文字は，“\uXXXX”に変換する。ここで，XXXX は変換対象の文字の文字コードを表す，4 桁の 16 進数である。また，変換前の文字列に含まれる各文字は，2 バイトで表すことができるものとする。

クラス Encoder は，サニタイズを行うクラスが継承する抽象クラスである。パブリックメソッド encode は，引数で与えられた文字列をサニタイズし，結果を返す。クラス HtmlEncoder は，HTML のタグに埋め込む文字列をサニタイズするクラスである。クラス JavaScriptEncoder は，JavaScript の文字列として埋め込む文字列をサニタイズするクラスである。

〔プログラム 1〕

```
import java.util.HashMap;
import java.util.Map;

abstract public class Encoder {
   private Map<Character, String> conversionTable =
      new HashMap<Character, String>();

   protected void addConversion(char c, String s) {
      conversionTable.put(c, s);
   }

   protected void addNoConversion(char c) {
      conversionTable.[ a ];
   }

   protected void addNoConversion(char[] collection) {
      for (char c : collection) {
         addNoConversion(c);
      }
   }

   abstract protected String encode(char c);

   public String encode(String s) {
      if (s == null) {
         return null;
```

```
        }

        String result = "";
        for (char c [ b ]) {
            String t = [ c ];
            if (t == null) {
                t = encode(c);
            }
            result += t;
        }
        return result;
    }
}
```

〔プログラム2〕

```
public class HtmlEncoder extends Encoder {
    private static String ALPHAS = "ABCDEFGHIJKLMNOPQRSTUVWXYZ";
    private static String NUMERICS = "0123456789";
    private static String PUNCTUATIONS =
        "!#$%()*+,-.:;=?@[\\]^_`{|}~";

    public HtmlEncoder() {
        addNoConversion(ALPHAS.toCharArray());
        addNoConversion(ALPHAS.toLowerCase().toCharArray());
        addNoConversion(NUMERICS.toCharArray());
        addNoConversion(PUNCTUATIONS.toCharArray());
        addConversion('<', "&lt;");
        addConversion('>', "&gt;");
        addConversion('&', "&amp;");
        addConversion('"', "&quot;");
    }

    protected String encode(char c) {
        return [ d ];
    }
}
```

〔プログラム3〕

```
public class JavaScriptEncoder extends Encoder {
    private static String ALPHAS = "ABCDEFGHIJKLMNOPQRSTUVWXYZ";
    private static String NUMERICS = "0123456789";

    public JavaScriptEncoder() {
        addNoConversion(ALPHAS.toCharArray());
        addNoConversion(ALPHAS.toLowerCase().toCharArray());
        addNoConversion(NUMERICS.toCharArray());
    }

    protected String encode(char c) {
```

```
        if (c [ e ] 256) {
            return String.format("\\x%02X", (int) c);
        }
        return String.format("\\u%04X", (int) c);
    }
}
```

設問1　プログラム1～3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　clear()　　イ　get(c)

ウ　put(c, String.valueOf(c))　　エ　remove(c)

bに関する解答群

ア　: conversionTable.keySet()

イ　: s.toCharArray()

ウ　= 0; c < s.length(); c++

エ　= 0; s.indexOf(c) > 0; c++

cに関する解答群

ア　addConversion(c, t)　　イ　addNoConversion(c)

ウ　conversionTable.get(c)　　エ　conversionTable.remove(c)

dに関する解答群

ア　"&#" + (int) c + ";"　　イ　"&#" + c + ";"

ウ　"\\u" + (int) c　　エ　"\\u" + c

eに関する解答群

ア　!=　　イ　<　　ウ　==　　エ　>

設問2　プログラム4を実行したときに，プログラム2のメソッド encode が呼び出される回数として正しい答えを，解答群の中から選べ。

ここで，プログラム1～3中の［　　］には正しい答えが入っているものとする。

〔プログラム4〕

```
public class HtmlEncoderTest {
    static public void main(String[] args) {
        new HtmlEncoder().encode("<script>alert('注意!');</script>");
    }
}
```

解答群

ア　0　　イ　4　　ウ　5　　エ　13　　オ　30

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1，2の説明〕

非負整数nについて，階乗の値F(n)を求める副プログラムFACTである。

$F(n)=1 \quad (n=0,\ 1)$

$F(n)=n\times(n-1)\times\cdots\times 1 \quad (n\geqq 2)$

例えば，$F(3)=3\times2\times1=6$である。

(1) FACTは，nがGR1に設定されて呼ばれ，計算結果をGR0に設定して呼出し元に戻る。nは，F(n)の値が16ビット符号なし2進整数の範囲に収まるように与えられる。FACTは副プログラムMULTを利用する。

(2) MULTは，16ビット符号なし2進整数同士の乗算を行う副プログラムであり，被乗数と乗数がそれぞれGR0，GR2に設定されて呼ばれ，乗算結果をGR0に設定して呼出し元に戻る。乗算結果の桁あふれは発生しないものとする。

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

〔プログラム1〕

(行番号)

```
 1  FACT    START
 2          RPUSH
 3          LD      GR0,=1
 4          SUBL    GR1,=1
 5          [   a   ]
 6          JUMP    FIN
 7  NEXT    LD      GR2,=1
 8  LP      LAD     GR2,1,GR2
 9          CALL    MULT
10          [   b   ]
11          JNZ     LP
12  FIN     RPOP
13          RET
14          END
```

〔プログラム 2〕

```
MULT    START               ; シフトによる乗算
        RPUSH
        LD      GR1,GR0     ; 被乗数を GR1 に保持
        LD      GR0,=0      ; 乗算結果の初期化
        LD      GR3,=15
        LD      GR2,GR2
LP      JZE     FIN
        JPL     CONT
        LD      GR4,GR1
        [   c   ]
        ADDL    GR0,GR4
CONT    LAD     GR3,-1,GR3
        SLL     GR2,1
        JUMP    LP
FIN     RPOP
        RET
        END
```

設問 1　プログラム 1, 2 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　NEXT　　イ　JNZ　NEXT　　ウ　JOV　NEXT

エ　JPL　NEXT　　オ　JZE　NEXT

b に関する解答群

ア　ADDL　GR1,=1　　イ　LAD　GR1,1,GR1　　ウ　LD　GR1,GR1

エ　SRL　GR1,1　　オ　SUBL　GR1,=1

c に関する解答群

ア　SLL　GR0,0,GR3　　イ　SLL　GR1,0,GR3　　ウ　SLL　GR4,0,GR3

エ　SRL　GR0,0,GR3　　オ　SRL　GR1,0,GR3　　カ　SRL　GR4,0,GR3

設問2　プログラム1のFACTを使用してF(4)を求めるとき，行番号12のRPOP命令実行直前のGR2に設定されている値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　　オ 5

設問3　F(n)は次のように再帰的に表現することができる。

$F(n)=1 \quad (n=0,\ 1)$

$F(n)=n\times F(n-1) \quad (n\geqq 2)$

上記に基づいて，プログラム3のFACT2を作成した。ここで，FACT2はFACTと同様に呼ばれる。また，FACT2に制御が移り，行番号2のRPUSH命令を実行した直後のSPの値をBとする。

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。

FACT2を使用してF(3)を求めるとき，行番号11の命令を初めて実行した直後のSPの値は［　d　］である。また，当該SPで示される番地に格納されている値は［　e　］である。

〔プログラム3〕

(行番号)

```
 1  FACT2   START
 2          RPUSH
 3          CALL    RSUB            ; 再帰処理の本体を呼ぶ
 4          RPOP
 5          RET                     ; 主プログラムへ戻る
 6  RSUB    LD      GR4,GR1
 7          SUBL    GR4,=1
 8          JPL     NEXT            ; n≧2 の場合 NEXT へ
 9          LD      GR0,=1          ; GR0 ← F(0), F(1)の値
10          RET
11  NEXT    PUSH    0,GR1
12          LAD     GR1,-1,GR1
13          CALL    RSUB            ; F(n−1)を計算
```

```
14          POP     GR2
15          CALL    MULT
16          RET
17          END
```

dに関する解答群

ア　B + 1　　イ　B + 2　　ウ　B

エ　B - 1　　オ　B - 2

eに関する解答群

ア　1

イ　2

ウ　3

エ　行番号4の命令が置かれたアドレス

オ　行番号14の命令が置かれたアドレス

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1～3に答えよ。

〔表計算の説明〕

ある会社では，社員ごとの社内向け学習教材の学習進捗状況を表計算ソフトで管理している。

〔ワークシート：学習進捗管理（単元記述部分）〕

(1)　社員ごとにワークシート“学習進捗管理”が用意され，学習進捗状況が記載される。ワークシート“学習進捗管理”の単元記述部分である列A～Gの例を，図1に示す。

	A	B	C	D	E	F	G
1	ID	単元名	項目学習順序	標準日数	学習開始日	学習完了日	判定
2	100	1. 基礎理論	任意	10	2014-05-20	2014-05-29	○
3	200	2. アルゴリズム	順次	15	2014-05-30	2014-06-20	△
4	300	3. コンピュータ構成要素	順次	15	2014-06-22		
5	400	4. システム構成要素	任意	25			
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
13	1200	12. 企業と法務	任意	10			

注記　列Cは設問2で使用する。

図1　ワークシート“学習進捗管理”（単元記述部分）の例

(2)　学習教材は12の単元から構成されており，単元にはそれぞれ固有のIDが割り振られている。

(3)　セルA2～A13には，単元のIDが昇順に格納されている。単元のIDは100から昇順に100刻みで付与されている。

(4)　セルB2～B13には，単元名が格納されている。

(5)　各単元は複数の項目から構成されている。セルC2～C13には，単元に含まれる項目の学習順序（以下，項目学習順序という）が格納されている。セルの値が“任意”の場合はどの項目から学習してもよいことを示し，セルの値が“順次”の場合は各項目を項目のIDが小さいものから順番に学習しなければならないことを示し

表計算

ている。

(6) セル D2 ～ D13 には，その単元の学習を開始してから完了するまでの学習に要する標準的な日数（以下，標準日数という）が格納されている。

(7) セル E2 ～ E13 とセル F2 ～ F13 には，それぞれ対応する単元の学習を開始した日（以下，学習開始日という），学習を完了した日（以下，学習完了日という）がそれぞれマクロ StartLearning と FinishLearning を用いて設定される。日付は yyyy-mm-dd の形式で表示されるが，表計算ソフトの内部では 1970 年 1 月 1 日からの経過日数を整数値で保持している。計算にはこの内部の整数値を利用する。

(8) セル G2 ～ G13 には，それぞれの単元の学習に関する判定結果が表示される。学習完了日が設定されたとき，学習に要した日数が標準日数以下の場合には“○”が表示され，そうでなければ“△”が表示される。学習開始日も学習完了日も学習に要した日数に含まれる。ここで，休日は考慮せず，休日も 1 日と数える。例えば，学習開始日が 2014 年 5 月 10 日で学習完了日が 2014 年 5 月 12 日の場合，学習に要した日数は 3 である。また，学習完了日が設定されていない場合には空値が表示される。

設問 1　ワークシート“学習進捗管理”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。

それぞれの単元の学習が標準日数内に完了したかどうかを表示するために，次の式をセル G2 に入力し，セル G3 ～ G13 に複写する。

IF(F2 ＝ null, ［ a ］)

a に関する解答群

ア　IF(F2 － E2 ＋ 1 ≦ D2, '○', '△'), null

イ　IF(F2 － E2 ≦ D2, '○', '△'), null

ウ　IF(F2 － E2 － 1 ≦ D2, '○', '△'), null

エ　null, IF(F2 － E2 ＋ 1 ≦ D2, '○', '△')

オ　null, IF(F2 － E2 ≦ D2, '○', '△')

カ　null, IF(F2 － E2 － 1 ≦ D2, '○', '△')

表計算

〔ワークシート：学習進捗管理（項目記述部分）〕

(1) この学習教材の項目の総数は 52 である。ワークシート“学習進捗管理”の項目記述部分である列 I～M の例を，図 2 に示す。

	I	J	K	L	M	N	O
1	ID	項目名	学習開始可能	学習開始日	学習完了日		登録
2	101	1.1 離散数学	可	2014-05-20	2014-05-23		301
3	102	1.2 応用数学	可	2014-05-23	2014-05-23		
4	103	1.3 情報通信	可	2014-05-22	2014-05-23		
5	104	1.4 計測制御	可	2014-05-25	2014-05-29		
6	201	2.1 データ構造	可	2014-05-30	2014-06-02		
7	202	2.2 アルゴリズム	可	2014-06-03	2014-06-08		
8	203	2.3 プログラミング	可	2014-06-12	2014-06-20		
9	301	3.1 プロセッサ	可	2014-06-22			
10	302	3.2 メモリ	不可				
11	303	3.3 入出力装置	不可				
12	401	4.1 システムの構成	不可				
⋮	⋮	⋮	⋮	⋮	⋮		⋮
53	1203	12.3 法務	不可				

注記 列 O は設問 3 で使用する。

図 2 ワークシート“学習進捗管理”（項目記述部分）の例

(2) セル I2～I53 には，項目の ID が昇順に格納されている。項目の ID は，その項目が属する単元の ID の下 2 桁を 01 から始まる連番にしたものである。

(3) セル J2～J53 には，項目名が格納されている。

(4) セル K2～K53 には，それぞれの項目が学習開始可能であれば“可”が表示され，そうでなければ“不可”が表示される。学習開始可能であるとは，次の条件①，②をともに満たしていることをいう。

① 項目が属する単元が学習開始可能である。単元が学習開始可能であるとは，単元の ID が 100 であるか，又は，より小さい ID の単元の学習が全て完了していることをいう。

② 次の条件(a)又は(b)のどちらかを満たしている。

(a) 項目が属する単元の項目学習順序が“任意”である。

表計算

(b) 項目が属する単元の項目学習順序が“順次”であり，その項目の ID の下 2 桁が 01 であるか，又は，その単元の中でより小さい ID の項目の学習が全て完了している。

(5) セル L2 ～ L53 とセル M2 ～ M53 には，それぞれ対応する項目の学習開始日，学習完了日がそれぞれマクロ StartLearning とマクロ FinishLearning を用いて設定される。

設問 2 ワークシート“学習進捗管理”に関する次の記述中の［　　］に入れる正しい答えを，解答群の中から選べ。

(1) それぞれの項目が学習開始可能か否かを表示するために，セル K2 に“可”を入力する。

(2) 次の式をセル K3 に入力し，セル K4 ～ K53 に複写する。

```
IF(論理積(
    論理和(
        切捨て(I3, −2)＝100,
        表引き($F$2～$F$13, [ b ], 1)≠null),
    論理和(
        垂直照合(切捨て(I3, −2), $A$2～$C$13, 3, 0)＝[ c ],
        剰余(I3, 100)＝1,
        [ d ])),
  '可', '不可')
```

b に関する解答群

ア　照合一致(切捨て(I2, −2), A2～A13, 0)

イ　照合一致(切捨て(I2, −2), A2～A13, 0)−1

ウ　照合一致(切捨て(I2, −2), A2～A13, 0)＋1

エ　照合一致(切捨て(I3, −2), A2～A13, 0)

オ　照合一致(切捨て(I3, −2), A2～A13, 0)−1

カ　照合一致(切捨て(I3, −2), A2～A13, 0)＋1

表計算

c に関する解答群

ア　'順次'　　イ　'任意'

d に関する解答群

ア　L3＝null　　イ　L3≠null　　ウ　M2＝null

エ　M2≠null　　オ　M3＝null　　カ　M3≠null

設問 3　学習開始日を設定するマクロ StartLearning と学習完了日を設定するマクロ FinishLearning をワークシート“学習進捗管理”に格納した。マクロ StartLearning 中の［　　］に入れる正しい答えを，解答群の中から選べ。

(1)　セル O2 に学習を開始する項目の ID を入力し，マクロ StartLearning を実行すると，次の①，②をともに満たす場合，対応する項目の学習開始日にマクロが実行された日付を設定する。

①　その項目が学習開始可能である。

②　その項目の学習開始日が空値である。

また，その項目が属する単元の学習開始日が空値ならば，単元の学習開始日にもマクロが実行された日付を設定する。

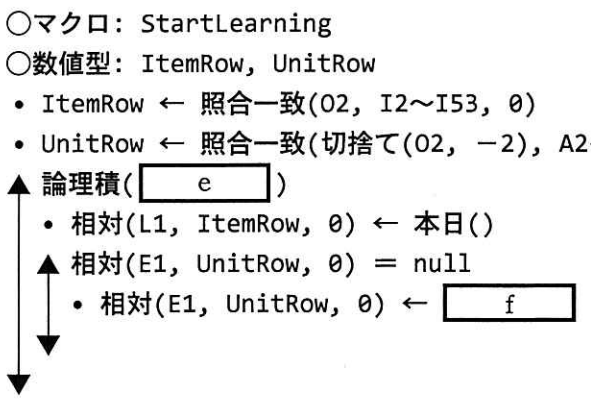
(2)　マクロ StartLearning の実行時のセル O2 には，セル I2 ～ I53 にあるいずれかの項目の ID が正しく入力されているものとする。

(3)　マクロ StartLearning では，マクロの実行日を取得するために表 1 に示す関数を用いる。

表 1　マクロ StartLearning で用いる関数

書式	説明
本日()	1970 年 1 月 1 日からこの関数を実行した日までの経過日数を整数値で返す。

〔マクロ: StartLearning〕



eに関する解答群

ア　表引き(K2～K53, ItemRow, 1) = '可', 表引き(E2～E13, UnitRow, 1) = null

イ　表引き(K2～K53, ItemRow, 1) = '可', 表引き(E2～E13, UnitRow, 1) ≠ null

ウ　表引き(K2～K53, ItemRow, 1) = '可', 表引き(L2～L53, ItemRow, 1) = null

エ　表引き(K2～K53, ItemRow, 1) = '可', 表引き(L2～L53, ItemRow, 1) ≠ null

オ　表引き(K2～K53, ItemRow, 1) = '不可', 表引き(E2～E13, UnitRow, 1) = null

カ　表引き(K2～K53, ItemRow, 1) = '不可', 表引き(E2～E13, UnitRow, 1) ≠ null

キ　表引き(K2～K53, ItemRow, 1) = '不可', 表引き(L2～L53, ItemRow, 1) = null

ク　表引き(K2～K53, ItemRow, 1) = '不可', 表引き(L2～L53, ItemRow, 1) ≠ null

fに関する解答群

ア　表引き(E2～E13, ItemRow, 1)

イ　表引き(F2～F13, ItemRow, 1)

ウ　表引き(F2～F13, UnitRow, 1)

エ　表引き(L2～L53, ItemRow, 1)

オ　表引き(L2～L53, UnitRow, 1)

カ　表引き(M2～M53, ItemRow, 1)

キ　表引き(M2～M53, UnitRow, 1)

■ Javaプログラムで使用するAPIの説明

java.lang
public final class String
クラス String は文字列を表す。

メソッド

static public String format(String format, Object... args)
指定された書式文字列と引数を使って，フォーマットされた文字列を返す。
引数： format — 書式文字列
args — 書式文字列の書式指示子により参照される引数
戻り値：フォーマットされた文字列

public int indexOf(int ch)
指定された文字が，この文字列中で最初に出現する位置のインデックスを返す。
引数： ch — 文字
戻り値：指定された文字が最初に出現する位置のインデックス
この文字列中に指定された文字がなければ −1

public int length()
この文字列の長さを返す。
戻り値：この文字列の長さ

public char[] toCharArray()
この文字列を文字配列に変換する。
戻り値：この文字列の文字の並びが格納されている文字配列

public String toLowerCase()
この文字列内の全ての大文字を，小文字に置き換えた文字列を返す。
戻り値：この文字列内の全ての大文字を小文字に置き換えた文字列

static public String valueOf(char ch)
指定された文字の文字列表現を返す。
引数： ch — 文字
戻り値：文字 ch の文字列表現

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 void clear()

保持しているキーと値の対応付けを，全て削除する。

---

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。このキーが既にほかの値と対応付けられていれば，その値は指定された値に置き換えられる。

引数： key — キー
value — 値

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

---

public V remove(Object key)

指定されたキーの対応付けが登録されていれば，削除する。

引数： key — キー

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 HashMap<K, V>

インタフェース Map のハッシュを用いた実装である。

メソッドの説明は，インタフェース Map の項を参照

コンストラクタ

---

public hashMap()

空の HashMap を作る。

java.util

public interface Set<E>

型 E の要素を集合（セット）として管理するインタフェースを提供する。

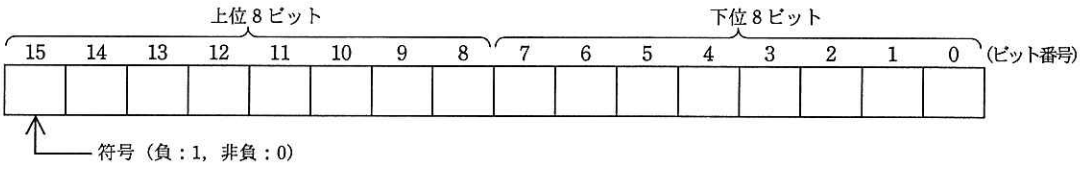
Java.lang.Iterableを継承する。

■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。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 〕

〔 メ モ 用 紙 〕

〔 メ モ 用 紙 〕

〔 メ モ 用 紙 〕

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

退室可能時間	13:40 ～ 15:20

7. **問題に関する質問にはお答えできません。**文意どおり解釈してください。
8. 問題冊子の余白などは，適宜利用して構いません。
9. Java プログラムで使用する API の説明，アセンブラ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。