●平成30年度秋期

午後問題 解答・解説

問1　情報セキュリティ事故と対策（情報セキュリティ）　(H30秋-FE 午後問1)

【解答】

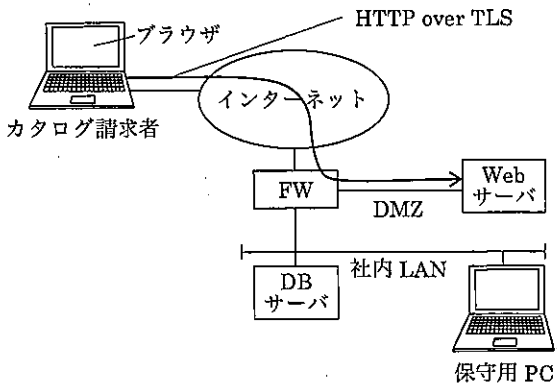
［設問1］　ウ
［設問2］　a－エ，b－ア，c－エ
［設問3］　イ

【解説】

公開サーバのWebアプリケーションソフト（本問ではWebアプリ）への攻撃，及びセキュリティ対策をテーマとした出題である。「SQLインジェクション」，「クロスサイトスクリプティング」，「WAF」などのセキュリティの専門的な言葉が数多く登場する。しかし，これらはいずれも基本情報技術者試験の出題範囲に含まれる用語である。なお，表1の「プレースホルダ」は，「?」などの特別な文字列を使ってSQL文を完成させるSQLインジェクション対策である。

この問題は，難易度は標準的と言えるが，たとえ難しい問題であっても，4択という利点を生かし，基礎知識を使って消去法で正解を導くテクニックも活用して，合格ラインを突破してほしい。

さて，設問の解説に入る前に，問題文から想定されるA社のシステム構成図を図Aに示す。構成図をイメージしながら設問を解くことは重要である。



図A　A社のシステム構成図

事させる契約で，受託者は作業の完成を約束するのでなく労働者の派遣を約束する。派遣労働者は派遣先の指揮命令に従って作業することになり，労働場所や就業時間その他の労働条件は必要に応じて派遣契約の内容に取り入れて両社で取り決める（労働者派遣法）。休暇取得ルールは雇用契約に伴う就業条件であり，請負契約なら請負作業の受託者側の責任で決めればよいので，雇用関係がない発注者側には発言権はない。請負契約を締結していながら労働者の休暇取得ルールを発注者側の指示に従って取り決めているとすると，労働者を雇用している受託者側で管理すべき労働条件を発注者側で指示しているので，請負契約といいながら実態は派遣契約の方法で運用していることになる。したがって，（ア）が正解である。このような運用は偽装派遣と呼ばれ，請負契約があったとしても法律上は労働者派遣とみなされる。

イ：請負業務の受託者が自己の雇用する労働者指導や評価を行うのは当然である。

ウ：請負業務の受託者が職場規律や職場秩序の保持に努めるのは当然である。

エ：請負業務では発注者側の要請は受託者側の責任者が窓口となって受け付けることになる。

［設問1］

下線①は，「インターネット経由でSQLインジェクション攻撃を行い」とある。この攻撃の説明として適切な答えを，解答群の中から選ぶ。

SQLインジェクション攻撃では，攻撃者がWebアプリに対して，データベース操作の命令文（SQL文）を混入（インジェクト）したデータを入力することで，データベースを不正に操作する。したがって，（ウ）が適切である。

ア：攻撃者が，DNSに登録されているドメインの情報を改ざんして攻撃者のWebサイトに誘導する攻撃は，DNSキャッシュポイズニングである。

イ：「インターネット経由でDBサーバに不正ログインする」とあるが，A社のDBサーバは社内LANに接続されており，インターネットから直接アクセスできない。誤りである。

エ：「攻撃者が，インターネット経由で送信されている情報を盗聴する」とある。SQLインジェクション攻撃とは無関係である。盗聴をするには専用の盗聴ツールや暗号を解除する仕組みが必要である。

午後解答

［設問2］

表1中の空欄に入れる対策を答える。

・空欄a：「SQLインジェクション攻撃からの防御」に関する対策を選ぶ。

SQLインジェクション攻撃は，データベース操作の命令文の組立てを，文字列連結を悪用して行われる。例えば，入力された「' OR 1=1」などの文字列をデータ操作の命令文の一部分にすることで，SQL文のWHERE句の結果を強制的に真にするといった手法である。そこで，文字列にエスケープ処理（文字列の変換）を施し，命令文ではなく単なる文字列にすることが有効な対策となる。したがって，（エ）が正解である。

ア：「Webサーバ内のファイル名を直接指定できないようにする」という対策は，ディレクトリトラバーサルへの対策である。

イ：SQLインジェクション攻撃は，Webサーバのメモリを直接操作されなくても発生し得る脅威である。一方で，Webサーバのメモリが直接操作されることによる攻撃としては，DoS攻撃や，root奪取が挙げられる。この対策はこうした攻撃に対して有効である。

ウ：SQLインジェクション攻撃は，Webページに不正な文字列を入力して攻撃が成立する。したがって，「Webページに出力する要素に対して，エスケープ処理を施す」ことは対策にならない。

・空欄b：「情報流出リスクの低減」に関する対策を選ぶ。

〔情報セキュリティ事故を踏まえたシステム面での対策〕で焦点になっているのは，カタログ請求者の情報流出であることを踏まえて考える必要がある。「カタログ請求者の情報の適切な保管期間を定め」，「保管期間を過ぎた時点で

データベースから消去」すれば，情報を保持する期間が短くなり情報流出のリスクが低減される。したがって，（ア）が適切である。

イ：「カタログ請求者の情報を，カタログ送付後に直ちに，データベースから消去」してしまうと，〔カタログ請求者の情報の登録〕の要件が満たされない。具体的には，カタログ請求者が別のカタログを請求したいときなどに，登録した電子メールアドレスとパスワードによるログインができない。

ウ：「電子メールにディジタル署名を付ける」ことは，データの改ざんの検知などには有効であるが，データを暗号化するものではないため，情報流出リスクの低減にはつながらない。

エ：「情報を定期的にバックアップする」ことは，データの破壊などによる対策にはなるが，情報流出リスクの低減にはつながらない。

・空欄c：「情報流出の原因と流出した情報の範囲の特定」に関する対策を選ぶ。

「データベースへのアクセスログを取得」すれば，どこからのアクセスなのか，いつ流出したか，どの範囲のデータにアクセスしたのかなどが分かる。その結果，情報流出の原因と流出した情報の範囲の特定につながる。したがって，（エ）が適切である。

ア：保守用PCを必要なときだけ起動しても，情報流出の原因特定などにはつながらない。

イ：インストールするミドルウェアを必要最低限にすれば，流出するリスクが低減するかもしれない。しかし，情報流出の原因特定などにはつながらない。

ウ：ハードディスクのデフラグメンテーションによって，処理のパフォーマンス向上は期待できる。しかし，情報流出の原因特定などにはつながらない。

［設問3］

クロスサイトスクリプティングなどの攻撃の対策として適切な答えを選ぶ。

Webアプリでは，その入力データに不正な通信が含まれる場合に，通信を遮断したり，出力内容をマスキングしたりすることで，情報流出を防ぐという対策が有効であるが，これを装置（あるいはソフトウェア）として行うのがWAF（Web Application Firewall）である。したがって，（イ）が適切である。

ア：「DBサーバを，Webサーバと同じく，DMZに設置する」と，攻撃者がDBサーバに直接アクセスできるようになる。その結果，攻撃されるリスクが増えてしまう。

ウ：「Webサーバを増設して冗長化」すれば，可用性の向上にはつながるが，情報流出を狙った攻撃の防御策にはならない。

エ：保守用PCのログインパスワードを推測が難しい複雑なものにすれば，保守用PCへの不正ログインへの有効な対策となり得る。しかし，Webサーバへ直接攻撃を仕掛けるクロスサイトスクリプティングの対策にはならない。

問2　プロセスのスケジューリング（ソフトウェア）　(H30 秋・FE 午後問 2)

【解答】

［設問 1］　オ

［設問 2］　a－イ，b－イ，c－キ

【解説】

プロセス（又はタスク）のスケジューリングに関する問題である。スケジューリングに関する知識問題は午前試験での出題頻度が高い。ラウンドロビン方式や状態遷移図は基本情報技術者試験でも必須の内容なので，理解しておく必要がある。平成 26 年秋に同類の問題が出題されているので参考にするとよい。上位の応用情報技術者試験では定番の内容であり，今後も出題される可能性が高い内容である。

本問題ではラウンドロビン方式と優先度順方式を取り上げている。内容は示されているので，その手順に則ってタイムチャートなどを作成し，解決していけばよい。

設問 2 の優先度順方式では，プロセスの優先度が変化し，どのプロセスに CPU の実行権が割り当てられるかを正確に把握する必要がある。全てのプロセスが終了（消滅）するまで吟味すると時間を要するし，ミスも発生しやすいが，求められている内容はプロセス A の処理 1 が終了するまでの時刻となっているので，正確な追跡ができれば解答できる。一つ一つ丁寧に確認，検証していくことが求められる。

［設問 1］

図 3 のプロセスの状態遷移について図 A で確認しておく。

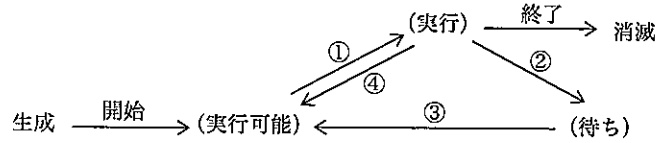


図 A　プロセスの状態遷移

生成されたプロセスは，生成と同時に実行可能状態になり，開始の矢印に沿って遷移する。実行可能状態とは，いつでも CPU の実行権の割当てが可能な状態である。

次に，実行可能状態にある幾つかのプロセスの中から一つを取り出し，CPU の実行権を割り当て，①に沿って実行状態に遷移する。このとき，複数ある実行可能状態のプロセスのどれに CPU の実行権を割り当てるかは，スケジューリング方式によって決まる。

実行状態のプロセスに入出力が発生した場合は②に沿って待ち状態に遷移する。入出力が終了すると，③に沿って再び実行可能状態になる。

ここで，④の遷移は実行状態のプロセスを強制的に実行可能状態に戻す操作であ

午後解答

り，これをプリエンプション（横取り）という。実行状態のプロセスは，タイムクウォンタムの経過によってプリエンプションされる。プリエンプションはラウンドロビン方式だけではなく，優先度順方式ではより高い優先度のプロセスが実行可能状態になったときに発生する。

実行が終了したプロセスは，終了の矢印に沿って消滅する。

プロセスXの処理時間及び待ち時間は表1に示されているが，タイムクウォンタムが20ミリ秒なので，処理1，処理2の処理時間を分割（実行を中断）し，図Aの①～④の遷移を当てはめてみると，表Aのようになる。

表A　プロセスXの分割と遷移

プロセス名 分割 遷移	処理時間及び待ち時間							
	処理1		入出力 待ち1	処理2			入出力 待ち2	処理3
X	30		30	50			30	10
分割	20	10		20	20	10		
遷移	① ④	①	② ③	① ④	① ④	①	② ③	①

タイムクウォンタムが20ミリ秒なので，処理1は30＝20＋10に分割される。処理2は50＝20＋20＋10に分割される。処理時間が終了するまでは，①と④及び①の遷移となる。また，入出力待ち1及び2では，②，③の遷移となる。処理3はタイムクウォンタム未満の処理時間なので，①から消滅へと遷移することになる。

このため，①は6回，②は2回，③は2回，④は3回なので，正解は（オ）である。

解き方としては，表Aのような表を作るほかに，タイムチャートを作成し，処理1から時間の経過に沿って，①～④を書き込んで解いてもよいだろう。選択肢の内容から，ポイントは①と④の回数なので，特に注意する。

［設問2］

優先度順方式に関する内容である。優先度を正確に把握しながら解く必要があり，設問数が少ないので，ミスをすると設問2で得点できないことになる。

・空欄a：プロセスXの処理をタイムクウォンタムで分割し，設問文の(1)～(3)に沿って優先度の推移を確認していくと表Bのようになる。

(1) プロセスが生成された場合，優先度3を与える。

(2) タイムクウォンタムが経過して，実行を中断した場合，優先度が1のときは優先度1，優先度が2～5のときは，1段階下げた優先度を与える。

(3) 入出力が完了した場合，優先度5を与える。

表B　プロセスXの優先度の推移

プロセス名 分割 優先度の推移	処理時間及び待ち時間							
	処理1		入出力待ち1	処理2			入出力待ち2	処理3
X	30		30	50			30	10
分割	20	10		20	20	10		10
優先度の推移	3	2		5	4	3		5

したがって，優先度の推移は，「3→2→5→4→3→5」の（イ）である。

・空欄b，c：プロセスがA～Cの三つあり，複雑な内容になるが，求められているのは最初の流れだけなので，タイムチャートを作成しながら正確にプロセスの動きを把握する。

プロセスAの処理1の開始から終了する時間までを追跡する。プロセスA～Cの処理1に着目すると，タイムクウォンタムが20ミリ秒であるため，プロセスA，Bの分割時間は20，10となる。このことを踏まえて，生成時刻に沿って吟味すると，図Bのようになる。

(1) プロセスAが生成されると実行可能状態になり，優先度3のキューに登録されると同時にCPUの実行権が与えられ実行される。

(2) プロセスAが実行されてから10ミリ秒後にプロセスBが生成され，実行可能状態になり，優先度3のキューに登録されるが，CPUは同じ優先度のプロセスAが実行中のため実行可能状態となる。

(3) プロセスAの処理1（20ミリ秒）が終了すると，優先度は3から2になり，キューに登録される。一方，実行可能状態であるプロセスBの優先度は3で高いため，プロセスBにCPUの実行権が与えられ実行される。

したがって，空欄bの解答は「20」ミリ秒後の（イ）である。

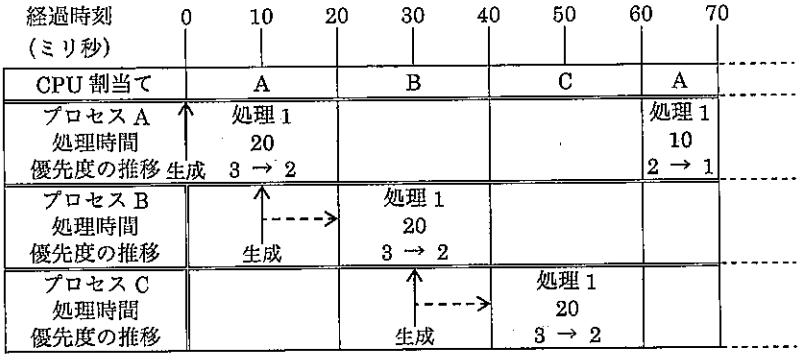
(4) プロセスAが生成されてから30ミリ秒後にプロセスCが生成され，優先度3のキューに登録されるが，同じ優先度のプロセスBが実行中のため実行

可能状態になる。

(5) プロセスBの処理1（20ミリ秒）が終了すると，優先度は3から2になる。この時点で，優先度2のキューには，プロセスA，Bの順に登録される。一方，待ち状態であるプロセスCの優先度は3で高いため，プロセスCにCPUの実行権が与えられ実行される。

(6) プロセスCの処理1（20ミリ秒）が終了したとき，優先度は3から2になるが，優先度2のキューにはプロセスA，Bが登録されており，最後尾に登録される。先頭はプロセスA（先入れ先出し）なので，処理1の残りの10ミリ秒がCPUで実行され，プロセスAの処理1の30ミリ秒全てが終了することになる。

したがって，空欄cの正解は，「70」ミリ秒後の（キ）である。



注記1：↑は各プロセスの生成時刻を示す。

注記2：処理時間はタイムクウォンタムで分割している。

注記3：-->はプロセスが生成されてから処理が開始されるまで実行可能状態で待つ。

注記4：優先度の推移は「処理の開始時→処理の終了時」を示す。

図B　プロセスAの処理1の開始から終了まで

【参考】

この問題では，コンピュータのCPUは一つとするとあるが，現在ではスマートフォンなどでも8個程度のCPU（マルチコア）が一般的になっている。この場合，プロセスは同時に最大8個が実行状態となると考えれば同じようなスケジューリング方式を適用することができる。また，設問2の優先度順方式は，現実の多くのOSで採用されているスケジューリング方式のひな型であり，動的にCPUを多く使うプロセスの優先度を低くし，入出力が多いプロセスの優先度を高くするとによって，全体のスループットを高めることができる。

問3 コンサートチケット販売サイトの関係データベースの設計及び運用（データベース） (H30秋-FE 午後問3)

【解答】

［設問1］ ア
［設問2］ a－エ
［設問3］ b－ア
［設問4］ ウ

【解説】

コンサートチケット販売サイトのチケット販売を管理する関係データベースに関する問題である。基本情報技術者試験のデータベース問題では，SELECT文を主体としたSQL文の基本的な問題が多く出題されている。本問題でも設問2，3がSELECT文に関する問題となっており，基礎知識があれば得点できる問題である。また，設問4ではCASE式に関する問題が出題されている。CASE式は，標準SQLのSQL-92から提供された機能であり，IF文がないSQLでは便利な機能であるが，SQLの基本的な範囲を若干超えているかもしれない。しかし，書式を覚えていれば解答できる問題であるため，少し幅広く関数やキーワードの書式を覚えておくとよいだろう。

［設問1］

データベースのデータの整合性を保つために，CREATE TABLE文等のDDL（Data Definition Language）文を使用する際に制約を付けることができる。

本問題では，問題文で示されている条件を基に，表名.列名と制約の正しい組合せを選ぶ。制約には次のものがあるので，選択肢を確認しながら考えていく。

・一意性制約：データの重複を排除する。
・非NULL制約：データのNULL保存を排除する。
・検査制約：データの値の範囲等をチェックする。
・参照制約：テーブル間の参照整合性を保証する。

ア：〔コンサートの席の説明〕(2)に「各席種の価格（常に有料）」とあり，決済額は必ず0円より多くなる（設問4でポイント制度が導入されると決済額＝0円の場合もあるが，ここでは考えないものとする）。また，〔販売サブシステムの説明〕(4)に「決済期限日を過ぎた販売表中のレコードと販売IDが同じレコードが決済表にない場合，その購入申込みは取り消されたものとして，バッチ処理によって決済表に当該販売IDを主キーとするレコードを追加する。このレコードの決済日はNULLで，決済額は－1とする」と記述されているので，決済額は－1になることがある。これらのことから，決済額＞0円や決済額＝－1円となり，0円や－1円より小さい決済額の登録を禁止するためには，決済額に検査制約を設定する必要がある。したがって，「決済表.決済額」と「検査制約」の組合せは正しい。

イ：（ア）と同様に〔販売サブシステムの説明〕(4)の記述を確認する。「このレコード

の決済日は NULL で，決済額は－1 とする」と記述されているので，決済表.決済日は NULL になることがある。したがって，「決済表.決済日」と「非 NULL 制約」の組合せは誤りである。

ウ：図 1「販売サブシステムで利用しているデータベースの表構成とデータの格納例」には席種を管理する表はなく，商品詳細表.席種は他のテーブルのデータを参照することはない。したがって，「商品詳細表.席種」と「参照制約」の組合せは誤りである。

エ：〔販売サブシステムの説明〕(2)に「会員が購入申込みを行うと，販売サブシステムは一意な販売 ID を生成して販売表にレコードを追加する」とあり，販売表のレコードは会員がチケットを購入するたびに作成されることが分かる。会員は，複数のコンサートチケットを購入することができるため，販売表の中では複数レコードで同じ会員 ID が保存される。したがって，「販売表.会員 ID」と「一意性制約」の組合せは誤りである。

正解は（ア）である。

［設問２］

“販売終了”の表示判定を行うために，販売できない席数を求める SQL 文の空欄に入れる適切な答えを選ぶ。対象はコンサート ID が C00001，席種が S である。

購入申込み時点での販売席数は，販売表の席数を集計すれば求められる。しかし，〔販売サブシステムの説明〕(4)に「決済期限日を過ぎた販売表中のレコードと販売 ID が同じレコードが決済表にない場合，その購入申込みは取り消され」とあり，実際の販売席数を出力するには販売表の席数の集計から，取り消された席数を差し引く必要がある。そのため，設問にある SQL 文では販売表と決済表を結合して席数を集計している。

・空欄 a1：まず，空欄 a1 について考える。SQL 文の FROM 句内の空欄であり，解答群から結合演算に関するキーワードが入ることが分かる。

〔販売サブシステムの説明〕(3)に「会員が支払手続を行うと，決済処理として販売サブシステムは販売 ID を主キーとするレコードを決済表に追加する」とあり，支払手続が行われるまで決済表にはレコードが追加されないため，販売表のレコードには決済表と結合できないレコードがある。しかし，まだ支払手続が行われていない販売も販売済の席数として集計する必要があるため，決済表と結合できない販売表のレコードも抽出しなければならない。そのようなレコードを抽出するためには，外（そと）（外部）結合を使用する。この SQL 文では空欄の左側に販売表，右側に決済表があり，販売表（左側）のレコードを全て抽出したいため，左外（ひだりそと）結合（LEFT OUTER JOIN）を使用する。したがって，空欄 a1 には「LEFT OUTER JOIN」が入る。

なお，INNER JOIN，LEFT OUTER JOIN，RIGHT OUTER JOIN の違いは次の

とおりである（OUTER は省略可能）。

＜JOIN 句＞
INNER JOIN ：結合条件と一致するレコードだけを抽出
LEFT OUTER JOIN ：左側の表のレコードを全て抽出
RIGHT OUTER JOIN ：右側の表のレコードを全て抽出

・空欄 a2：次に，空欄 a2 について考える。WHERE 句内の抽出条件であり，解答群から決済表の決済額に関する条件であることが分かる。
この SQL 文は，既に販売済の席数を集計する。販売済と扱うのは，販売済かつ支払手続が行われた（決済表の決済額が 0 円以上）レコードと，販売済かつ販売期限前でまだ支払手続が行われていない（決済表の決済額が NULL）レコードである。したがって，空欄 a2 には「決済表.決済額 IS NULL OR 決済表.決済額 >= 0」が入る。

正しい組合せは（エ）である。

［設問3］

決済期限日まで残り 3 日となっても支払手続が行われていない会員の氏名，電子メールアドレス及び販売 ID を出力する SQL 文の空欄に入れる適切な答えを選ぶ。

会員の氏名，電子メールアドレスは会員表で，販売 ID，決済期限日は販売表でそれぞれ管理されているため，これらの表を使用する必要がある。また，支払手続が行われているかどうかは決済表で管理されているため，会員表と販売表に加えて，決済表も使用する必要がある。

解答群を見ると，選択肢の違いは FROM 句，WHERE 句内の販売表.販売 ID に関する条件，決済表などに関する条件である。そこで，選択肢の違いから考えていく。

ア：WHERE 句内の副問合せで決済表にない販売 ID の販売を抽出している。支払手続が行われていない販売は，決済表にレコードのない販売であるため，この条件は電子メールの送信条件に適合している。正しい。

イ：WHERE 句内の副問合せで決済表から決済額が 0 円以上のレコードの販売 ID を抽出し，販売表と結合している。しかし，この条件で抽出されるのは支払手続が行われた販売であり，電子メールの送信条件とは異なる。誤りである。

ウ：WHERE 句で会員表，販売表，決済表を結合しているが，この条件で抽出されるのは決済済み，又は取消し済みの販売であり，電子メールの送信条件とは異なる。誤りである。

エ：WHERE 句で会員表，販売表，決済表を結合し，決済表.決済額が−1 でないレコードを抽出している。しかし，この条件で抽出されるのは支払手続が行われた販売であり，電子メールの送信条件とは異なる。誤りである。

したがって，正解は（ア）である。

［設問４］

ポイント制度の導入に当たり，決済表の付与ポイントを更新する正しいSQL文を選ぶ。

設問文では，ポイントを更新するSQL文の条件として「前日に決済処理された販売IDごとに，その決済額が20,000円以上，10,000円以上20,000円未満，10,000円未満の場合に，それぞれ3%，2%，1%のポイントを付与する」とあり，決済額によって付与するポイントが異なることが分かる。

SQL文において，条件によってアクションを変更したい場合は，CASE式を使用する。

＜書式＞

（単純CASE式）

```
CASE　項目名
  WHEN　値１　THEN　項目名 = 値１の場合のアクション
  WHEN　値２　THEN　項目名 = 値２の場合のアクション
  ⋮
  WHEN　値X　THEN　項目名 = 値Xの場合のアクション
  ELSE　値1～Xに該当しない場合のアクション
END
```

（検索CASE式）

```
CASE
  WHEN　条件１　THEN　条件１に該当する場合のアクション
  WHEN　条件２　THEN　条件２に該当する場合のアクション
  ⋮
  WHEN　条件X　THEN　条件Xに該当する場合のアクション
  ELSE　条件1～Xに該当しない場合のアクション
END
```

単純CASE式は，評価対象の項目と値が同じかどうかを評価し，アクションを選択する。一方，検索CASE式は，条件を自由に設定できるため，単純CASE式に比べて条件分岐の応用の幅が広い。

解答群から書式にあったものを選べばよく，（ウ）が検索CASE式である。

```
UPDATE 決済表 SET 付与ポイント = (
  CASE WHEN 決済額 >= 20000 THEN FLOOR (決済額 * 0.03)
  WHEN 決済額 >= 10000 THEN FLOOR(決済額 * 0.02)
  ELSE FLOOR(決済額 * 0.01) END )
       WHERE DATEDIFF(NOW(), 決済日) = 1
```

なお，このUPDATE文で付与ポイントを更新すると，取り消された申込み（決済額＝－1円）のレコードも更新され，付与ポイントに－1が登録される。しかし，ポイント残高に加える処理の中で配慮すればよいため，ここでは問題ないと考えられる。

ア，エ：SQLにIF文はなく，誤りである。

イ：検索CASE式の書式であるが，一つ目の条件の前にWHEN句がないため，誤りである。

問4　ネットワークの障害分析と対策（ネットワーク）　(H30 秋・FE 午後問 4)

【解答】

［設問］　a－イ，b－オ（順不同）
　　c－ウ
　　d－ア，e－カ（順不同）

【解説】

本問は，最初にネットワーク構成図とネットワーク構成の問題文があり，〔障害の発生〕，〔セグメントの追加〕，〔障害発生の予防〕と続き，空欄が設けられている。

〔障害の発生〕では，障害発生のシチュエーションが示され，障害箇所を特定するために試行した結果から原因を特定する。〔セグメントの追加〕では，セグメントの追加の内容とそれに伴って設定したファイアウォールについて，その設定誤りから発生する事象を解答する。〔障害発生の予防〕では，ルータの増設の二つの構成案から構成案1を採用した理由を解答する。

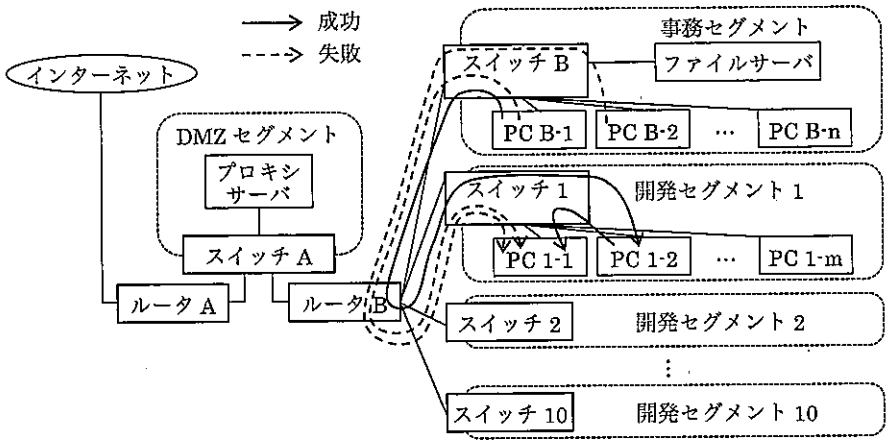
［設問］

最初の〔障害の発生〕に，「ある日，“事務セグメント内の PC B-1 から，リモートデスクトップ機能を用いて，開発セグメント 1 内の PC 1-1 を遠隔操作しようとしたが，接続できなかった”と報告があった」という記述があり，この障害の報告から二つの試行をしている。

① 事務セグメント内の PC B-2 から，PC 1-1 にリモートデスクトップ機能を用いて接続を試みたが，失敗した。
② 事務セグメント内の PC B-1 から SSH を用いてログインした開発セグメント 1 内の PC 1-2 で ping コマンドを実行し，PC 1-1 から応答が返ってくることを確認した。

障害内容と①と②の結果をネットワーク構成図上に表現すると図 A になる。

②の成功から，PC B-1 から PC 1-2 までの経路上の機器と PC 1-2 から PC 1-1 までの経路上の機器は OSI 参照モデルのネットワーク層レベルの問題がないことが分かる。障害の発生内容と①の失敗から，リモートデスクトップ接続で利用しているプロトコルが，経路上もしくは PC 1-1 で受け付けられなくなっていると推測できる。具体的には，ネットワーク構成の(6)に「ルータ B は，内蔵のパケットフィルタ型のファイアウォール機能によってセグメント間の通信の可否を制御しており」とあるように，ファイアウォール設定ができるルータ B，もしくは PC 1-1 上で動作しているリモートデスクトップ接続サービスのいずれかとなる。空欄 a，b の解答群では（イ）の「PC 1-1 のソフトウェア」，（オ）の「設定を含むルータ B のソフトウェア」が該当するので，（イ）と（オ）が障害の原因と考えられる不具合であり，解答となる。



図A　G社のネットワーク構成と障害

次の〔セグメントの追加〕では，開発セグメント 11 を追加し，ルータ B に対してファイアウォールの設定を追加したが誤りがあり，それに関することを空欄 c で解答する。このファイアウォール設定に必要な事項については，ネットワーク構成の(3)，(4)に記述されている。

(3) 事務セグメント内の PC は，リモートデスクトップのクライアント機能又は SSH を用いて，開発セグメント内の PC を遠隔操作できる。

(4) 開発セグメント内の PC は，事務セグメント内のファイルサーバにアクセスできる。

(3)，(4)からルータ B に設定すべきファイアウォールの設定は表 A のようになる。網掛け部分は，開発セグメント 11 内の PC から事務セグメント内のファイルサーバにアクセスできるようにする，送信元ネットワークと宛先ネットワークの設定である。この部分が表 1 では逆に設定されているため，アクセスできない。したがって，(ウ)の「当該 PC から事務セグメント内のファイルサーバにアクセスできない」が解答になる。

表A　ルータ B に追加すべきファイアウォールの設定

送信元ネットワーク	宛先ネットワーク	ポート番号（サービス）	可否
10.1.11.0/24	10.0.0.0/16	445（ファイル）	可
10.0.0.0/16	10.1.11.0/24	22（SSH）	可
10.0.0.0/16	10.1.11.0/24	3389（リモートデスクトップ）	可

最後の〔障害発生の予防〕では，ルータCを増設することでルータBの負荷を分散させることを目的に構成案1と構成案2が検討され，最終的に構成案1を採用することになった。ここで構成案1を採用するときに重視した点を，空欄d，eとして解答する。（ア）～（カ）について，構成案1と構成案2を比較してみる。

ア：「可用性を高められる」……構成案1ではルータB，ルータCのどちらかが故障しても，もう一方が動作していれば通信は継続できるので可用性を高められる。構成案2ではルータBが故障した場合は事務セグメントとインターネット間の通信ができなくなり，ルータCが故障した場合は事務セグメントと開発セグメント間の通信ができなくなる。このため，構成案1の方が可用性を高められる。

イ：「機密性を高められる」……どちらの構成案でも機密性を高めることはできない。

ウ：「障害発生時に原因を特定しやすい」……構成案2では何らかの障害が発生した場合にルータB，ルータCのどちらの経路で障害が発生したのか，原因を特定しやすいが，構成案1では構成案2に比べてルータB，ルータCのどちらの経路で障害が発生したのか，原因を特定しにくい。

エ：「セグメント間で通信する際に経由する機器が少なくなる」……セグメント間の通信で許可されているものは，事務セグメントとDMZセグメント，事務セグメントと開発セグメント1～11であり，どちらの構成案でも経由する機器の数は同じである。

オ：「ルータB，Cとスイッチ間をつなぐLANケーブルの本数が少なくて済む」……図2から，スイッチ1～11の接続について考えみる。構成案1ではスイッチ1～11がルータBにもルータCにも接続されている。構成案2ではルータCだけに接続されている。このことから考えても，LANケーブルの本数が少なくて済むのは構成案2である。

カ：「ルータBとルータCの負荷に大きな差が生じないように調整できる」……構成案1はルータBとルータCの役割が同じであるため，負荷の調整はできる。構成案2はインターネットとの通信と開発セグメントとの通信で負荷の差があれば，そのままルータBとルータCの負荷の差が生じる。

これらの比較結果から，構成案1を採用するときに重視したのは，（ア）と（カ）の二つである。

問5　購買管理システムで行う処理（ソフトウェア設計）　(H30秋-FE 午後問5)

【解答】

[設問1]　a－ア
[設問2]　b－ア，c－オ，
[設問3]　d－イ，e－ア

【解説】

購買部門が部品を発注する際に利用する購買管理システムで行う処理に関する問題である。部品の購入依頼情報を格納したファイル（以下，依頼ファイルという）中の各レコードについて，購買ファイルを更新できるかどうかをチェックする購買ファイル更新可否チェック処理の設計及びテスト設計を行う。

購買ファイル更新可否チェック処理によって，更新することができるレコードは更新用依頼ファイル，更新することができないレコードは更新対象外依頼ファイルに振り分けられ，更新用依頼ファイルは，購買ファイル更新処理に引き渡される。これについて，問題文には「一連の処理として実行する購買ファイル更新処理に引き渡す」と記述されている。また，「依頼ファイルは，1日に1回，バッチ処理時間帯に製造部門から受け取る」という記述もあり，図1「購買ファイル更新可否チェック処理の位置付け」が示されている。つまり，購買ファイル更新可否チェック処理は単体で実行するプログラムではなく，購買ファイルを更新する機能の一部である。具体的には，バッチ処理で依頼ファイルを受け取った後に実行され，チェック結果の出力ファイルを購買ファイル更新処理に引き渡す処理である。購買ファイル更新可否チェック処理の出力ファイルは，購買ファイル更新処理の入力ファイルとして使用される。そのため，ソフトウェア設計には，出力ファイルとなる更新用依頼ファイルと，更新対象外依頼ファイルを振り分ける条件を適切にプログラムするためのアルゴリズム（流れ図の作成）が求められる（設問1）。

また，テスト設計には，そのアルゴリズムが適切であることを検証するためのテストケースの作成が求められる（設問2）。そして，テスト結果が妥当であると判定するには，そのテストケースを実行するためのテスト用レコードを事前に作成し，テスト結果を予測しておく必要がある。テスト結果と予測が一致したから妥当なのであり，テスト結果だけでは妥当とはいえない（設問3）。

ちなみに，本問のように設計時にテスト設計も行う開発モデルを，W字モデルという。設計→製造→テスト（テスト設計を含む）の順に作業を行うV字モデルに対し，「W」という文字で，設計とテスト設計を並行して行うことを示している。W字モデルでは，設計→テスト設計と続けて作業することで，テストの観点から設計漏れや設計不備を検出し，プログラム作成前に設計フィードバックを行う。設計が完了すればテスト設計を行うことは可能であり，本問のように，複数の条件を組み合わせて処理実行を判断するような仕様では有効な手法である。ただし，仕様変更が発生すると，設計だけでなく，テスト設計もやり直さなければならない。

［設問1］

図2「購買ファイル更新可否チェック処理の流れ図」（以下，流れ図という）の空欄の穴埋めである。依頼ファイルから読み込んだレコードを，対象外レコードとして出力するか，対象レコードとして出力するか，振り分ける条件を考える。

流れ図の基となる仕様は〔購買ファイル更新可否チェック処理の概要〕(2)の記述である。「依頼IDの昇順に整列された依頼ファイルのレコードを先頭から順に読み込んで，全てのレコードについて次の処理を行う」とあり，依頼ファイルを対象に繰返し処理を行っている。そして，①，②から，振り分ける条件は，購買ファイルに同じ依頼IDをもつレコードの有無と依頼種別，購買ステータスの組合せである。また，表2「依頼ファイル及び購買ファイルのレコード項目の説明」から，依頼IDは，購入依頼情報を識別するために「製造部門で採番して設定する一意な番号」，依頼種別は，「“登録”，“変更”，“削除”」の三つ，購買ステータスは，「“購買受付”，“見積り中”，“発注済”，“納品済”」の四つである。これを踏まえ，振り分ける条件の組合せを確認する。

まず，購買ファイルに同じ依頼IDをもつレコードの有無は，購買ファイルを検索しないと分からない。そのため，繰返し処理の中で，最初に，依頼ファイルから読み込んだレコードの依頼IDをキーにして購買ファイルを検索している。依頼IDをキーにするのは，表1「依頼ファイル及び購買ファイルのレコードの項目」を参照すると分かるが，購買ファイルも依頼IDを主キーとしてもっているためである。空欄a1の上にある条件分岐「レコードを読み込んだ」は，「購買ファイルからレコードを読み込んだかどうか」を意味している。

・空欄a1：条件分岐「レコードを読み込んだ」からNoに分岐した後，つまり，空欄a1は，購買ファイルからレコードを読み込めなかったときに行う条件分岐である。そして，この条件に該当したとき（Yes）は，対象レコード出力処理を実行している。購買ファイルにレコードがなく，かつ，対象として処理する必要があるのは，“新規”に発生した依頼である。したがって，空欄a1の条件式は，「依頼種別＝“登録”」である。

・空欄a2：購買ファイルからレコードを読み込んだとき，つまり購買ファイルに同じ依頼IDをもつレコードが存在するときの処理である。この場合，対象となる依頼種別は，“変更”と“削除”だが，②には「購買ステータスが“購買受付”又は“見積り中”」とある。これは，発注が完了した部品（“発注済”）や納品が完了した部品（“納品済”）の個数は変更したり削除したりできないので，それを除外することを意味している。そのため，分岐後すぐに購買ステータスの判定をしている。そして，その条件判定結果がYes，つまり，購買ステータスが“購買受付”又は“見積り中”のレコードに対して，依頼種別が“変更”であるかどうかを判定し，Noのときに空欄a2に分岐している。したがって，空欄a2の条件式は，残る依頼種別の判定であり，「依頼種別＝“削除”」となる。

空欄a1，空欄a2から，空欄aは（ア）が正解である。

［設問2］

表3「出力するファイルの決定表」を作成する。

決定表（デシジョンテーブル）とは，条件に応じて行う処理や動作を記述した表である。表を四つの部分に分け，上部の左に条件，右に条件の指定を“Y”又は“N”で示し，二重線で区切られた下部の左に動作，右に実行する動作を“X”で指定する。複数の条件が関連する処理を定義したり，条件同士の関連や発生し得る条件の組合せを整理したりするために作成する。また，条件部の条件の組合せと，それに対応する動作部の組みを規則といい，そのままテストケースとして活用することもできる。

条件部	条件1	Y	Y	Y	N
	条件2	Y	Y	N	−
	条件3	Y	N	−	−
動作部	動作1	−	−	−	X
	動作2	−	X	−	−
	動作3	−	−	X	−
	動作4	X	−	X	−

この規則では，
条件の組合せが
条件1が“Y”
かつ
条件2が“Y”
かつ
条件3が“N”
の場合，動作2を実行する。

図A　決定表

本問では，条件部には大きく分けて，条件 1)，購買ステータス，依頼種別の三つの条件がある。

条件 1)は，注 1)に「依頼ファイルから読み込んだレコードと依頼IDが同じ購買ファイルのレコードがある」と記述されていることから，流れ図の最初の分岐（空欄 a1 の上）を指している。つまり，「Y」は「レコードを読み込んだ，Yes」，「N」は「レコードを読み込んだ，No」を意味する。

購買ステータスは，更に四つの条件に分かれており，それぞれの条件に，「該当する」を意味する「Y」が三つある。三つという点に着目すると，依頼種別が更に三つの条件に分かれているので，購買ステータスと依頼種別を組み合わせるためであることが推測できる。

そして，動作部には，更新用依頼ファイルに出力，動作 2)の二つの動作がある。動作 2)は，注 2)に「更新対象外依頼ファイルに出力」とある。

・空欄b：購買ファイルに同じ依頼IDをもつレコードがあり，購買ステータスが“見積り中”の規則である。空欄bの左側と右側の規則について条件部を確認すると，空欄 b と異なるのは依頼種別である。左側の規則は“登録”，右側の規則は“削除”に「Y」が定義されているので，残るは“変更”である。そして，この条件に該当する依頼ファイルのレコードは，更新用依頼ファイルに出力する。したがって，空欄bは（ア）が正解である。

・空欄 c　購買ファイルに同じ依頼 ID をもつレコードがあり，購買ステータスが“納品済”の規則である。空欄 c の左側の二つの規則について条件部を確認すると，依頼種別は“登録”と“削除”にそれぞれ「Y」が定義されているので，残るは“削除”である。そして，この条件に該当する依頼ファイルのレコードは，更新対象外依頼ファイルに出力する。これは“納品済”なので，削除することはできないという理由で，更新対象外とすることを意味している。したがって，空欄 c は（オ）が正解である。

表 A　出力するファイルの決定表（表 3 からの抜粋）

この条件は同じ

条件 1)		Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
購買ステータス	購買受付	Y	Y	Y	–	–	–	–	–	–	–	–	–
	見積り中	–	–	–	Y	Y	Y	–	–	–	–	–	–
	発注済	–	–	–	–	–	–	Y	Y	Y	–	–	–
	納品済	–	–	–	–	–	–	–	–	–	Y	Y	Y
依頼種別	登録	Y	–	–	(Y)	–	–	Y	–	–	(Y)	–	–
	変更	–	Y	–	–	(Y)	–	–	Y	–	–	(Y)	–
	削除	–	–	Y	–	–	(Y)	–	–	Y	–	–	(Y)
更新用依頼ファイルに出力		–	X	X	–	X	X	–	–	–	–	–	–
動作 2)		X	–	–	X	–	–	X	X	X	X	X	X

○印の条件が異なる　　空欄 b　　空欄 c

［設問 3］

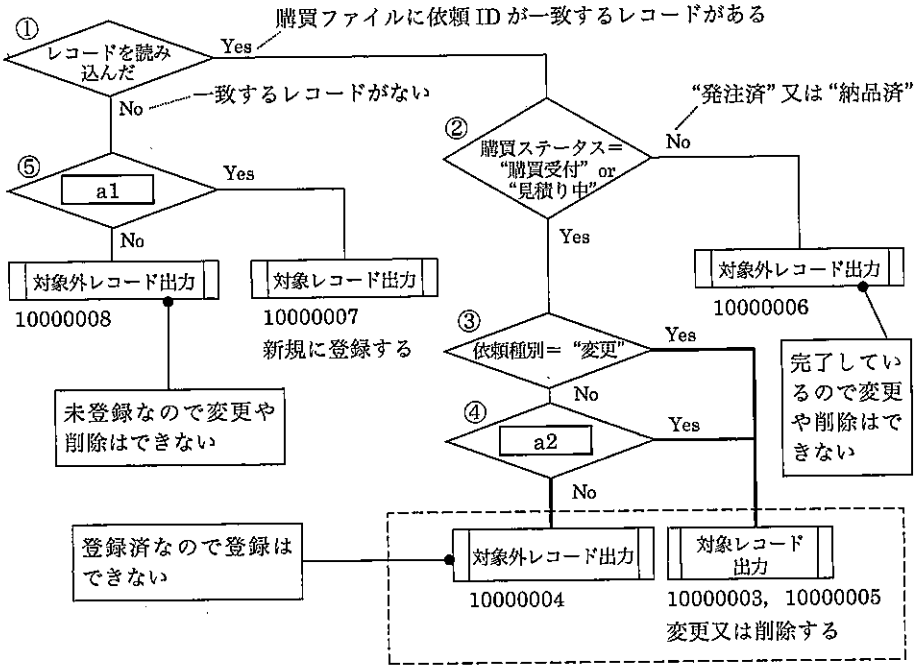
テスト用レコードでテストを実行した際のテスト結果を予測する。テスト結果は，流れ図の破線で囲んだ処理についてだけ答えることに注意する。破線で囲んだ処理は，空欄 a2 による分岐後の処理だが，対象レコード出力への分岐には「依頼種別＝“変更”」の条件判定で Yes となったレコードも含まれる。したがって，対象となるテスト用レコードは，次の条件を満たしていることになる（①，②とも Yes に分岐する）。

①購買ファイルに同じ依頼 ID をもつレコードが存在する
②購買ステータスが“購買受付”又は“見積り中”

この①，②に，条件分岐③，④，⑤を加えて示したものが表 B「テスト用レコードのトレース結果」と図 B の流れ図（図 2 からの抜粋）である。図 B 中の太線部分の分岐を通るレコードが空欄 d，e で処理され，表 B では網掛け部分が該当する。対象外レコード出力処理では依頼 ID が「10000004」のレコードが処理され，対象レコード処理では依頼 ID が「10000003 と 10000005」のレコードが処理される。したがって，空欄 d は（イ），空欄 e は（ア）が正解である。

表B　テスト用レコードのトレース結果

表5「依頼ファイルのテスト用レコード」		表4「購買ファイルのテスト用レコード」	トレース結果
依頼ID	依頼種別	購買ステータス	
10000003	削除	購買受付	①Yes→②Yes→③No→④Yes →対象レコード出力（削除実行）
10000004	登録	見積り中	①Yes→②Yes→③No→④No →対象外レコード出力（登録不可）
10000005	変更	見積り中	①Yes→②Yes→③Yes →対象レコード出力（変更実行）
10000006	削除	発注済	①Yes→②No→対象外レコード出力
10000007	登録	レコードなし	①No→⑤Yes→対象レコード出力
10000008	変更	レコードなし	①No→⑤No→対象外レコード出力



図B　購買ファイル更新可否チェック処理の流れ図（図2からの抜粋）

問6 プロジェクトのスケジュール作成（プロジェクトマネジメント）

(H30 秋・FE 午後問 6)

【解答】

［設問 1］ a－ウ，b－エ，c－オ，d－エ

［設問 2］ e－ウ，f－ク，g－オ（e，f は順不同）

【解説】

基本情報技術者試験の午後問題では，問 6 はプロジェクト管理の出題が多い。その中でもスケジューリング作成で，アローダイアグラムの問題が出題されたのは珍しい。平成 26 年春には問 7（経営戦略・企業と法務）で出題されている。

設問 1，2 のアローダイアグラムから，クリティカルパスや総所要日数などを答える問題は，図の所要日数を足していけばよいので，解答しやすい。アローダイアグラムから最早結合点時刻（最早開始日）と最遅結合点時刻（最遅開始日）を求め，クリティカルパスを算出する方法は必ず理解しておいてほしい。設問 2 のダミー作業を答える問題は，表 1 の作業内容をよく読み，図 2 のアローダイアグラムと照らし合わせながら考えなくてはならないので，多少時間を要するが，よく読めば関連が分かる。

平成 29 年までの，問題文を読んでいけばある程度解答できる問題から，平成 30 年は春，秋ともプロジェクト管理分野の技法を理解し，その技法を使用して解答する問題が出題されている。プロジェクト管理分野の技法を理解しておくことが必要である。

［設問 1］

まず，図 1 のアローダイアグラムから最早結合点時刻（最早開始日）を求める。出発点から順番に所要日数を足し，図 A のように最早結合点時刻を□の上段に記入する。結合点に矢印が二つ以上集まっているときには，数字が大きい方をとる。例えば，結合点 5 では，A，E の作業の所要日数は 3，B，E 及び B，F の作業の所要日数はともに 4 であり，この場合は 4 を選ぶ。この要領で，結合点 8 まで，最早結合点時刻を求める。次に最遅結合点時刻（最遅開始日）を求め，□の下段に記入する。結合点 8 からさかのぼって，所要日数を引いていき，結合点に矢印が二つ以上集まっているときには，今度は数字が小さい方をとる。

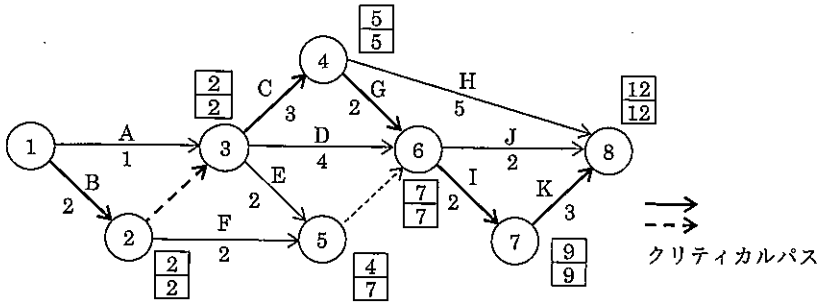


図 A

・空欄 a：クリティカルパスは最早結合点時刻と最遅結合点時刻の等しい（□の上段と下段の数字が等しい），最も余裕のない作業経路で，図 A の太矢線をつないだ「B, C, G, I, K」となる。したがって，正解は（ウ）である。

・空欄 b：全ての作業を完了するために必要な所要日数（総所要日数）は，最後の結合点 8 の「12」日である。したがって，正解は（エ）である。

・空欄 c：作業 J について，最早開始日（最早結合点時刻）と最遅開始日（最遅結合点時刻）求める。最早開始日は，作業 J 開始前の結合点 6 の日数である 7 日である。最遅開始日は，結合点 8 の総所要日数 12 から J の作業日数 2 日を引いた 10 日になる。最早開始日は「7 日」，最遅開始日は「10 日」，したがって，正解は（オ）である。

・空欄 d：余裕日数は 10−7=「3」日となる。したがって，正解は（エ）である。

［設問 2］

表 1 の作業一覧表から，図 2 のアローダイアグラムを作成し，それが途中であると設問文には示されている。そこで表 1 の作業名と作業項目を，図 2 のアローダイアグラムに当てはめたものが図 B である。

・空欄 e, f：図 2 に二つのダミー作業が欠けているのでそれを考える。アローダイアグラムでは，作業が A, B, C, I, J, K, L, M, N と，A, B, F, G, H, L, M, N と，A, B, C, D, E, L, M, N の経路が示されている。

まず，F, G, H の回付サービスについて考える。回付サービスについては，表 1 の作業名 C の作業内容に，「回付サービスに関しては，電子帳票システムとの連携要件を定義する」とある。作業一覧表の注 1）に，「連携要件とは，電子帳票システムとの連携に必要となる機能，データ項目及びインタフェースである」と示されている。回付サービスの連携要件については，作業名 F では，「連携要件は前提としない」，作業名 G は，「連携要件の適合性検証を実施し，導入する回付サービスを決定する」と示されている。したがって，作業名 C の作業の後に，作業名 G の作業で連携要件が必要となる。そこで作業名 C の後，作業名 G へ作業をつなげるために，「結合点 4 から結合点 5」（ウ）にダミー作業を追加する必要がある。

次に，作業名 I の外部設計の作業内容に「システム要件を基に電子帳票システムのソフトウェアの外部設計，ミドルウェアのパラメタ定義を行う」とある。作業名 E には「外部設計で定義したミドルウェアのパラメタ設定を行う」と示されている。そこで作業名 I の後，作業名 E へ作業をつなげるために，「結合点 7 から結合点 9」（ク）にダミー作業を追加する必要がある。

したがって，正解は（ウ）と（ク）である（順不同）。

・空欄 g：ダミー作業追加後の総所要日数は，アローダイアグラムの最早結合点時刻から求める（設問 1 と同様の方法を用いる）。結合点 13 での最早結合点時刻「200」がプロジェクト全体での総所要日数となる。したがって，正解は（オ）である。図 C では太矢線をつないだ経路になる。

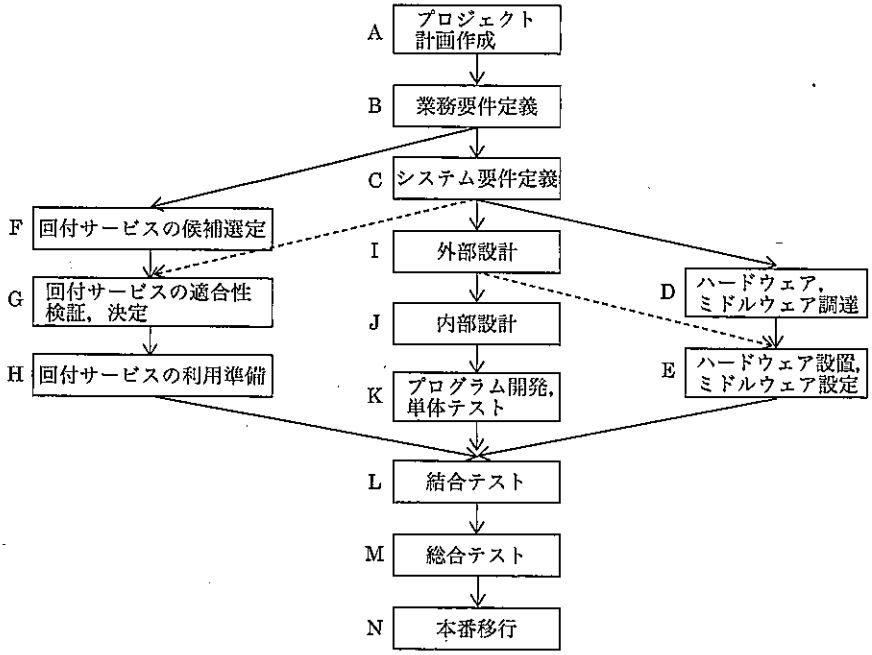


図 B

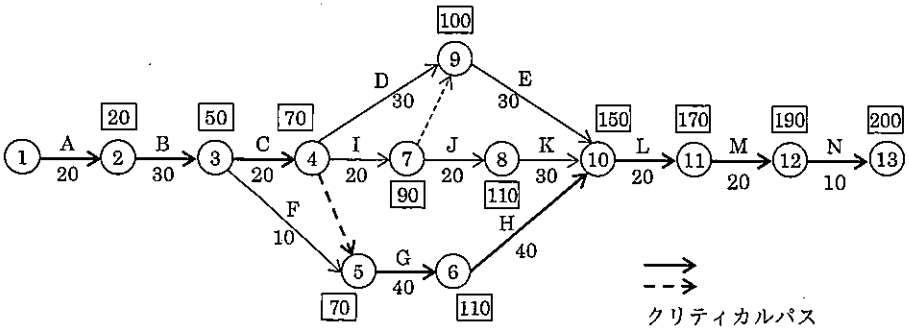


図 C

問7　広告制作業務の現状把握と改善（システム戦略）　(H30秋-FE 午後問7)

【解答】

［設問1］　ア

［設問2］　イ，ウ

［設問3］　a－イ，b－エ

【解説】

基本情報技術者の午後試験ストラテジ系の分野からは，専門用語などの知識を問う問題，論理的思考力を問う問題，及びそれらが混合する問題が出題されている。平成30年秋は論理的思考力を問う「システム戦略」分野から出題されており，広告制作会社において広告原稿を電子化することによる業務改善がテーマである。

設問1では，授受管理簿の内容について，設問2では，広告原稿の“紛失事故”が誘発されるおそれがある作業グループについて，設問3では，広告原稿を電子化することによる作業軽減について問われている。

［設問1］

この設問では，授受管理簿の一部が示された上で空欄部分を推測することが求められている。

〔制作部門での作業の流れ〕(1)④までの作業が完了しており，流れを追っていく。授受管理簿に記入しながら考えていくとよい。

① 授受管理簿の「通番」，「受付日」，「営業担当者」，「紙広告原稿枚数」及び「Web広告原稿枚数」には，実際に，該当項目が全て記入されている。通番6では「紙広告原稿枚数」は“－”となっており，紙広告原稿がないことを示している。

② 点検を行うので，授受管理簿に記入する項目はない。

③ ②の結果を受けて不備がある場合は，「返却事由」に“原稿不備”と記入して，(1)の作業を終える。通番6の作業では④まで実施されているので，不備がなかったと考えられる。したがって，本作業では，「返却日」，「返却事由」は空欄のままである。

④ 作業担当者を決定し，「審査担当者」，「紙広告作成担当者」及び「Web広告作成担当者」の欄に記入する。通番6では紙広告原稿がないので「紙広告作成担当者」の欄には“－”を記入する。

これらのことから，図Aのようになる。したがって，正解は，(ア)である。

通番	受付日	営業担当者	顧客名	紙広告原稿枚数	Web広告原稿枚数	返却日	返却事由	作業担当者		
								審査担当者	紙広告作成担当者	Web広告作成担当者
6	10月11日	佐々木	[redacted]	－	5枚			佐藤	－	渡辺

図A　10月11日における授受管理簿（通番6）

［設問2］

この設問では，〔制作部門での作業の流れ〕の中で，誘発されるおそれがある“紛失事故”について考察している。

「広告原稿をグループ間で受け渡すとき及び営業部門に渡すときに待ちが発生する場合がある。この場合に，一時的な保管が必要となって，“紛失事故”が誘発されるおそれがある」と記述されている。保管する可能性があるのは，広告原稿をグループ間で受け渡すとき，営業部門に渡すタイミングであることが分かる。また，受け渡すときに待ちが発生しない場合は保管の必要がない。

各作業において広告原稿を保管する可能性のある工程を洗い出してみる。

(1) 管理グループの作業1について確認する。

① 営業部門から広告原稿を受け取り，保管せず，すぐに処理している。

② 広告原稿の受渡しは発生しない。

③ 広告原稿に不備がある場合は，広告原稿を営業部門に渡すが，待ちは発生しない。

④ 広告原稿を審査担当者に渡すときに，待ちは発生しない。

①～④から，(1) 管理グループの作業1において“紛失事故”が誘発されるおそれはない。

(2) 審査グループの作業について確認する。

① 広告原稿の受渡しは発生しない。

② 広告原稿を管理グループに渡すときに，待ちは発生しない。

①，②から，(2) 審査グループの作業において“紛失事故”が誘発されるおそれはない。

(3) 管理グループの作業2について確認する。

① 審査グループでの審査結果が“不適”であれば，広告原稿を営業部門に渡すが，待ちは発生しない。

② 広告原稿を作成担当者に渡す。このとき，作成担当者が他の作業を行っている場合は，作業の終了を待って渡すことになる。この場合は渡すまでに待ちが発生し，広告原稿を保管する必要がある。このときに“紛失事故”が誘発されるおそれがある。

したがって，（イ）の「管理グループの作業2」が正解の対象となる。

(4) 作成グループの作業は解答群に含まれないので，確認を省略する。

(5) 管理グループの作業3について確認する。

① 紙広告とWeb広告について，受け取った広告原稿と作成済広告がそろっているかどうかを確認しているが，待ちの有無については明記されていない。

② 紙広告とWeb広告について，どちらかに不足がある場合は，必要なものが全てそろうまで待つ。その場合，既に受け取っている広告については保管する必要があり，このときに“紛失事故”が誘発されるおそれがある。

したがって，（ウ）の「管理グループの作業3」も正解の対象となる。

③　営業部門に広告原稿と作成済広告を渡すときに，待ちは発生しない。

(3)②及び(5)②の作業で保管が発生することが確認されたため，正解は（イ），（ウ）になる。

［設問３］

この設問では，広告原稿を電子化することによる，“紛失事故”が誘発されるおそれの低減，管理グループの作業軽減について考察している。

授受管理簿の通番４（図 B）に対応して管理グループが行った作業が，表１に整理されている。

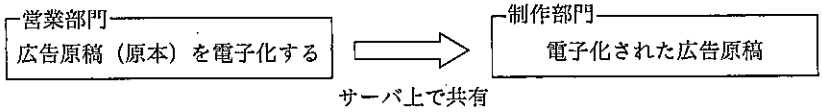
通番	受付日	営業担当者	顧客名	紙広告原稿枚数	Web 広告原稿枚数	返却日	返却事由	作業担当者		
								審査担当者	紙広告作成担当者	Web 広告作成担当者
4	10 月 6 日	松本	[redacted]	5 枚	－	10 月 10 日	作成完了	鈴木	田中	－

図 B　10 月 11 日における授受管理簿（通番 4）

広告原稿を電子化する時点の候補として，二つのポイントが挙がっている。

営業出口

・営業部門が制作部門に渡す直前に電子化する。
・広告原稿（原本）は営業部門がもっているので，制作部門での作業完了後に，広告原稿が制作部門から営業部門に手渡されることはない。



制作入口

・制作部門が営業部門から受け取った直後に電子化する。
・制作部門での作業完了後に，広告原稿（原本）が制作部門から営業部門に手渡される。

ここから，管理グループの作業における書類の受渡しについて確認していく。通番4にはWeb広告原稿がないのでWeb広告作成担当者との書類の受渡しは発生しないことに注意する。

(1) 管理グループの作業1について確認する。
① 営業部門から広告原稿を受け取っており，「広告原稿の受け（対営業部門）」が発生している。
② 書類の受渡しは発生しない。
③ 広告原稿に不備がある場合は，広告原稿を営業部門に渡すが，項番4の返却事由は「作成完了」であり，不備はない。したがって，書類の受渡しは発生しない。
④ 不備がないので広告原稿を審査担当者に渡す。したがって，「広告原稿の渡し（制作部門内）」が発生している。

(2) 審査グループの作業について確認する。
① 書類の受渡しは発生しない。
② 広告原稿と審査票を管理グループに渡すので，管理グループから見ると「広告原稿の受け（制作部門内）」と「審査票の受け（制作部門内）」が発生する。

(3) 管理グループの作業2について確認する。
① 審査グループでの審査結果が“不適”であれば，広告原稿を営業部門に渡すが，項番4の返却事由は「作成完了」であり，不備はない。したがって，書類の受渡しは発生しない。
② 紙広告原稿を紙広告作成担当者に渡すので，「広告原稿の渡し（制作部門内）」が発生する。

(4) 作成グループの作業について確認する。
① 管理グループへの書類の渡しは発生しない。
② 紙広告作成担当者から管理グループに対して作成済広告と広告原稿を渡す。したがって，管理グループから見ると「作成済広告の受け（制作部門内）」及び「広告原稿の受け（制作部門内）」が発生する。

(5) 管理グループの作業3について確認する。
①，② 書類を確認する。
③ 広告原稿と作成済広告を営業部門に渡すので，「広告原稿の渡し（対営業部門）」及び「作成済広告の渡し（対営業部門）」が発生する。

これらのことから，管理グループにおいて行われる作業のうち，受渡し作業は表Aのようになる。

午後解答

表A　管理グループが行った作業（受渡し）

作業	工程	件数	
広告原稿の受け（対営業部門）	(1)①	1	制作入口
広告原稿の渡し（対営業部門）	(5)③	1	
広告原稿の受け（制作部門内）	(2)②，(4)②	2	
広告原稿の渡し（制作部門内）	(1)④，(3)②	2	
作成済広告の受け（制作部門内）	(4)②	1	営業出口
作成済広告の渡し（対営業部門）	(5)③	1	制作入口
審査票の受け（制作部門内）	(2)②	1	
合計件数		9	

・営業出口：広告原稿を営業出口において電子化すると，管理グループが行う作業での広告原稿の受渡しは全て電子化されるので，手渡しによる作業は作成済広告の受渡しと審査票の受けだけになる。表Aから分かるように，その件数は「3」件である。
・制作入口：広告原稿を制作入口において電子化すると，営業出口に加え，広告原稿の営業部門との受渡しが手渡しになるので，件数は2件増えて「5」件になる。

したがって，空欄aの正解は（イ）であり，空欄bの正解は（エ）である。

問8　整数式の解析と計算（データ構造及びアルゴリズム）（H30秋-FE 午後問8）

【解答】

［設問1］　a－イ，b－エ
［設問2］　c－エ，d－ア
［設問3］　e－エ，f－イ，g－エ

【解説】

整数式を受け取り，括弧，演算子の優先順位に従い，解析及び計算を行う問題である。完成したプログラムが提示され，例を参考に読み進みながら全体を把握し，更に特性を一般化した場合や，処理の変更を考える。解析処理では整数式を三つの配列に設定し，計算処理では設定された値を用いて優先順位に従った計算を行う。処理に用いられる三つの配列の使用目的と具体的な値更新のタイミングに注意しながらプログラムの流れを追うようにする。

〔プログラムの説明〕(2)及び(4)から整数式は引数 Expression[]，整数式の文字数は引数 ExpLen で与えられ，整数式の値は Value[0]に設定する。〔プログラムの説明〕(6)には「受け取った整数式に誤りはないものとする」，「計算の過程で，あふれやゼロ除算は発生しないものとする」とあるため，本来の処理部分だけを考えればよい。

〔プログラム〕

プログラムの処理概要を確認し，行番号とともに示す。

```
（行番号）
   1  ○整数型関数: compute(文字型: Expression[], 整数型: ExpLen)
   2  ○文字型: Operator[100]
   3  ○整数型: OpCnt, Priority[100], Value[100]
   4  ○文字型: chr
   5  ○整数型: i, ip, nest
6～29  解析処理（詳細は〔プログラム（解析処理の部分）〕に示す）
30～56  計算処理（詳細は〔プログラム（計算処理の部分）〕に示す）
  57  ・return Value[0]
```

〔プログラム（解析処理の部分）の説明〕

図1「プログラム（解析処理の部分）を実行した直後の状態」から，配列の要素及び OpCnt が表す内容を示す。

Value[]：整数式の文字列の数字を出現順に数値に変換した値を格納。
Operator[]：演算子を数式の先頭から順番に格納。
Priority[]：それぞれの演算子の優先順位を表す値を格納。
OpCnt：Operator[]の配列要素数。すなわち整数式の演算子の数を格納。

〔プログラム（解析処理の部分）〕

```
（行番号）
     6   ・OpCnt ← 0
     7   ・Value[0] ← 0
     8   ・nest ← 0
     9   ■ i：0, i ＜ ExpLen, 1
    10   |  ・chr ← Expression[i]
    11   |  ▲ ('0' ≦ chr) and (chr ≦ '9')    /*数字 0～9 か? */
    12   |  |  ・Value[OpCnt] ← 10 × Value[OpCnt] ＋ int(chr)
    13   |  ▼
    14   |  ▲ (chr ＝'＋') or (chr ＝'－') or (chr ＝'×') or (chr ＝'÷')
    15   |  |  ・Operator[OpCnt] ← chr
    16   |  |  ▲ (chr ＝'＋') or (chr ＝'－')
①→  17   |  |  |  ・Priority[OpCnt] ← nest ＋ 1
②→  18   |  |  ▼  ・Priority[OpCnt] ← nest ＋ 2
    19   |  ▼
    20   |  ・OpCnt ← OpCnt ＋ 1
    21   |  ・Value[OpCnt] ← 0
    22   |  ▼
    23   |  ▲ chr ＝'('
③→  24   |  |  ・nest ← nest ＋ 10
    25   |  ▼
    26   |  ▲ chr ＝')'
④→  27   |  |  ・nest ← nest － 10
    28   |  ▼
    29   ■
```

プログラムの処理をブロックごとにまとめると，次のようになる。

・行番号 6～8：変数（OpCnt，Value[0]，nest）の初期設定。

・行番号 9～29：整数式を 1 文字ずつ処理し，整数式の長さ分（ExpLen）だけ繰り返す。添字 i を 0 から ExpLen－1 まで 1 ずつ増加させ，処理対象の文字を chr に設定する。

（1 文字ごとの処理）

・行番号 11～13：'0'から'9'の数字の場合の処理

初期値 0 に設定された Value[OpCnt]に連続する数字を加算する。複数桁の数字は 1 桁増えるごとに 10 倍になるため，Value[OpCnt]の数値を 10 倍してから処理対象数字を加算する。このとき，数字を整数値に変換する関数 int()を使用する。

・行番号 14～22：演算子の処理

'＋'，'－'，'×'，'÷'のとき，Operator[OpCnt]に文字を設定する。

'＋'，'－'の場合は nest に 1 を加算した値を設定し，'×'，'÷'の場合は 2 を加算した値を Priority[OpCnt]に設定する。

Value[OpCnt]，Operator[OpCnt]，Priority[OpCnt]の全てが設定されたので，OpCnt を 1 加算する。

次の数字の加算のために，Value[OpCnt]を初期値 0 に設定する。

・行番号 23～28：括弧の処理

'('の場合は nest に 10 を加算し，')'の場合は 10 を減算する。ここで，nest は括弧の深さを示す。最大値の nest が最も深い括弧の中を示し，最初に計算すればよいことが分かる。また，同じ括弧の深さの中の演算子は，優先順位によって nest ＋ 1，又は nest ＋ 2 となるので，Priority[]の値は等しくなる。


```
Priority[]      1      12   11     21      31      22     11   12     2
整数式        | 1 ＋ ( 6 ÷ 3 － ( 4 ＋ ( 1 0 ＋ 1 1 ) ÷ 7 ) ＋ 6 × 8 ) × 9 |
nest        0 →  10   →    20 → 30    →    20 → 10   →    0  →
```

図Ａ 括弧に対する nest と Priority[]の値の例

〔プログラム（計算処理の部分）〕

```
   (行番号)
      30  ■ OpCnt > 0
⑤→    31  |  ・ip ← 0
⑥→    32  |  ■ i：1, i < OpCnt, 1
⑦→    33  |  |  ▲ Priority[ip] < Priority[i]
      34  |  |  |  ・ip ← i
      35  |  |  ▼
      36  |  ■
      37  |  ・chr ← Operator[ip]
      38  |  ▲ chr ＝'＋'
      39  |  |  ・Value[ip] ← Value[ip] ＋ Value[ip ＋ 1]
      40  |  ▼
      41  |  ▲ chr ＝'－'
      42  |  |  ・Value[ip] ← Value[ip] － Value[ip ＋ 1]
      43  |  ▼
      44  |  ▲ chr ＝'×'
      45  |  |  ・Value[ip] ← Value[ip] × Value[ip ＋ 1]
      46  |  ▼
      47  |  ▲ chr ＝'÷'
      48  |  |  ・Value[ip] ← Value[ip] ÷ Value[ip ＋ 1]
      49  |  ▼
      50  |  ■ i：ip＋1, i < OpCnt, 1
      51  |  |  ・Value[i] ← Value[i ＋ 1]
      52  |  |  ・Operator[i － 1] ← Operator[i]
      53  |  |  ・Priority[i － 1] ← Priority[i]
      54  |  ■
      55  |  ・OpCnt ← OpCnt － 1
      56  ■
```

・行番号 30～56：設定した演算子がなくなるまで計算処理を繰り返す。
　OpCnt が 0 より大きい間処理を繰り返し，行番号 55 で 1 を減算する。
　行番号 31 で繰り返すたびに ip を 0 に初期化する。

（繰返し処理）

・行番号 32～36：Priority[]の要素の最大値を示す添字を ip に求める。値が等しいときは変更せず，前の値をそのまま最大値とする。ip には優先順位が最も高い演算子の添字が求められるため，その演算子から計算をする。
　添字 i を 1 から OpCnt より小さい間 1 ずつ増加させ，Priority[0]から順に比較し，Priority[i]が大きければ i を ip に設定する。

・行番号 37：chr に最も優先順位が高い演算子 Operator[ip]を設定する。

・行番号 38～49：chr の演算子に従って条件分けし，Value[ip]と Value[ip ＋ 1]の計算結果を Value[ip]に設定する。

・行番号 50～54：計算済みの配列要素に後ろの要素を前へシフトする処理
　演算子 Operator[ip]の計算が終了したので，ip より後ろで OpCnt より小さな Operator[]と Priority[]の要素を一つずつ前へ移動する。Value[ip]には計算結果が求められているので，Value[ip ＋ 2]から Value[OpCnt]までの要素を一つずつ前へ移動する。
・行番号 55：配列要素を一つずつ前へ移動したので，OpCnt を 1 減算する。

ここまでの解析処理，計算処理について，具体例を用いて配列の内容がどのように変わっていくか確認していく。

例 1： 2×（34－（5+67）÷8）

（解析処理の部分）
行番号 9～29 のループごとの処理について，「行」に示す行番号が実行された後の各変数の値を示す。網掛け部分は更新された箇所を示す。

ExpLen　15

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
Expression[]	2	×	(	3	4	－	(	5	+	6	7	)	÷	8	)

行	chr	OpCnt	nest	添字	0	1	2	3	4
6～8		0	0	Value[]	0				
9		0	0	i＜ExpLen が真　ループに入る					

i＝0

12	2	0	0	chr が数字　行番号 11 の条件が真 Value[0]＝10×Value[0]＋2＝10×0＋2＝2					
				Value[]	2				

i＝1

18	×	0	0	chr が演算子　行番号 14 の条件が真 行番号 16 の条件が偽 Priority[0]＝nest＋2＝0＋2＝2					
				Operator[]	×				
				Priority[]	2				
21	×	1	0	Value[]	2	0			

i＝2

24	(	1	10	chr が括弧　行番号 23 の条件が真 nest＋10＝0＋10＝10					

午後解答

i=3

行	chr	OpCnt	nest
12	3	1	10

添字	0	1	2	3	4
Value[]	2	3			

chr が数字　行番号 11 の条件が真
Value[1]＝10×Value[1]＋3＝10×0＋3＝3

i=4

12	4	1	10

Value[]	2	34			

chr が数字　行番号 11 の条件が真
Value[1]＝10×Value[1]＋4＝10×3＋4＝34

i=5

17	−	1	10

Operator[]	×	−			
Priority[]	2	11			

chr が演算子　行番号 14，16 の条件が真
Priority[1]＝nest＋1＝10＋1＝11

21	−	2	10

Value[]	2	34	0		

i=6

24	(	2	20

chr が括弧　行番号 23 の条件が真
nest＋10＝10＋10＝20

i=7

12	5	2	20

Value[]	2	34	5		

chr が数字　行番号 11 の条件が真
Value[2]＝10×Value[2]＋5＝10×0＋5＝5

i=8

17	＋	2	20

Operator[]	×	−	＋		
Priority[]	2	11	21		

chr が演算子　行番号 14，16 の条件が真
Priority[2]＝nest＋1＝20＋1＝21

21	＋	3	20

Value[]	2	34	5	0	

i=9

12	6	3	20

Value[]	2	34	5	6	

chr が数字　行番号 11 の条件が真
Value[3]＝10×Value[3]＋6＝10×0＋6＝6

i=10

12	7	3	20

Value[]	2	34	5	67	

chr が数字　行番号 11 の条件が真
Value[3]＝10×Value[3]＋7＝10×6＋7＝67

i＝11

行	chr	OpCnt	nest	添字	0	1	2	3	4
27	)	3	10	chr が括弧　行番号 26 の条件が真 nest－10＝20－10＝10					

i＝12

行	chr	OpCnt	nest						
18	÷	3	10	chr が演算子　行番号 14 の条件が真 行番号 16 の条件が偽 Priority[3]＝nest＋2＝10＋2＝12					
				Operator[]	×	－	＋	÷	
				Priority[]	2	11	21	12	
21	÷	4	10	Value[]	2	34	5	67	0

i＝13

行	chr	OpCnt	nest						
12	8	4	10	chr が数字　行番号 11 の条件が真 Value[4]＝10×Value[4]＋8＝10×0＋8＝8					
				Value[]	2	34	5	67	8

i＝14

行	chr	OpCnt	nest	
27	)	4	0	chr が括弧　行番号 26 の条件が真 nest－10＝20－10＝0

i＝15

行	chr	OpCnt	nest	
9	)	4	0	i＝15＜ExpLen が偽　ループを抜ける

〔プログラム（解析処理の部分）〕実行後の各配列の内容は，次のようになる。

Value[]	2	34	5	67	8
Operator[]	×	－	＋	÷	
Priority[]	2	11	21	12	

OpCnt	4

〔プログラム（計算処理の部分）〕

最も外側の繰返しを 1 回実行するときの変数の値をトレースする。

次に，「行」に示す行番号が実行された後の各値を示す。太線で囲んだ部分が次のループ処理の計算対象，網掛け部分が更新された箇所を示す。

行	OpCnt	ip	i	chr	添字	0	1	2	3	4
30	4				OpCnt＞0 が真　ループに入る					
31	4	0			Value[]	2	34	5	67	8
					Operator[]	×	－	＋	÷	
					Priority[]	2	11	21	12	

行	OpCnt	ip	i	chr	添字	0	1	2	3	4
32	4	0	1		i＜OpCnt が真　ループに入る					
33	4	0	1		Priority[0]＜Priority[1]が真					
34	4	1	1		i を ip へ設定 行番号 32 へループ					
32	4	1	2		i＜OpCnt が真　ループに入る					
33	4	1	2		Priority[1]＜Priority[2]が真					
34	4	2	2		i を ip へ設定 行番号 32 へループ					
33	4	2	3		Priority[2]＜Priority[3]が偽					
32	4	2	4		i＜OpCnt が偽　ループを抜ける					
37	4	2		+	Operator[2]を chr に設定					
39	4	2		+	行番号 38 が真 Value[2]＝Value[2]＋Value[3]＝5＋67＝72	Value[]	2	34	72	67	8
50	4	2	3	+	i＝ip＋1　i＜OpCnt が真　ループに入る					
53	4	2	3	+	Value[4]を Value[3]に設定 Operator[3]を Operator[2]に設定 Priority[3]を Priority[2]に設定	Value[]	2	34	72	8 ⇐ 8	
						Operator[]	×	−	÷ ⇐ ÷		
						Priority[]	2	11	12 ⇐ 12		
					行番号 50 へループ						
50	4	2	4	+	i＜OpCnt が偽　ループを抜ける					
55	3	2		+		Value[]	2	34	72	8	8
						Operator[]	×	−	÷	÷	
						Priority[]	2	11	12	12	
					OpCnt＝3 なので，斜線部分は処理対象外						

最も外側のループ行番号 30～56 を引き続き実行した後の変数の値を示す。

2 回目

55	2	2		÷	Priority[2]が最大値， Value[2]＝Value[2]÷Value[3]＝72÷8＝9 行番号 50 が偽　シフトは行わない

Value[]	2	34	9
Operator[]	×	−	
Priority[]	2	11	

3 回目

行	OpCnt	ip	i	chr
55	1	1		−

添字	0	1	2	3	4
Value[]			2	25	
Operator[]			×		
Priority[]			2		

Priority[1]が最大値,
Value[1]＝Value[1]−Value[2]＝34−9＝25
行番号 50 が偽　シフトは行わない

4 回目

行	OpCnt	ip	i	chr
55	0	0		×

添字	0	1	2	3	4
Value[]	50				
Operator[]					
Priority[]					

Priority[0]が最大値で,
Value[0]＝Value[0]×Value[1]＝2×25＝50
行番号 50 が偽　シフトは行わない

5 回目で OpCnt＝0 となり，行番号 30 の OpCnt＞0 が偽となるため，ループを抜け終了する。Value[0]に計算結果の 50 が求められる。

［設問 1］

・空欄 a：演算順序を示す nest に増減させる定数が 10 以外でも常に正しい演算順序が保証されることを実際の数値で確認する。整数式は図 1 の例を用いて解答群の数値を定数に当てはめて考える。OC は OpCnt, V は Value[], O は Operator[], P は Priority[]の値を示す。OC，ip，V，O，P の値は各ループで行番号 38 実行前の，計算処理を行う前の状態を示す。太線で囲んだ部分がそのループの添字 ip で示される演算子の計算対象となる。

'('と')'の 2 通りの場合で nest を増減させて区別するため，定数は 2 以上と予想できるが，まず，定数が 1 のときを考え，nest の値と整数式，プログラム実行状態を見ていくと，次のようになる。

Priority[]		2		2		3		3	
整数式		2	×	(3	4 −	(5	+ 6	7)	÷ 8)
nest	0 →	1	→	2	→	1	→	0	

1 回目

OC	4					
ip	2					
	0	1	2	3	4	
V	2	34	5	67	8	
O	×	−	+	÷		
P	2	2	3	3		

2 回目

	3				
	2				
	0	1	2	3	
V	2	34	72	8	
O	×	−	÷		
P	2	2	3		

3 回目

	2			
	0			
	0	1	2	
V	2	34	9	
O	×	−		
P	2	2		

4 回目

	1		
	0		
	0	1	
V	68	9	
O	−		
P	2		

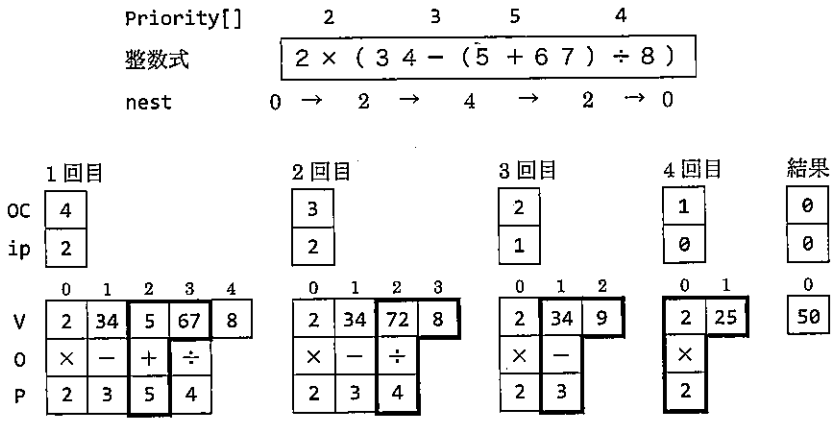
結果

	0
	0
	0
V	59

計算処理 3 回目に外括弧内の 34−9 を行うべきところが Priority[0]と括弧内の演算の Priority[1]が同じ値となり，2×34 を先に計算し，正しい計算順序が保証されない。

次に，定数が 2 のとき，nest の値と整数式，プログラム実行状態を見ると，次のようになる。2 以上であれば，括弧の内側の演算子の Priority[]は括弧の外側の演算子の Priority[]より大きくなるため正しい計算順序が保証される。したがって，「2 以上」（イ）が正解である。

「1 以上」（ア），「11 以下」（ウ），「12 以下」（エ）はともに 1 を含むため正しい計算順序が保証されない。



・空欄 b：空欄 a では具体例で考えたが，空欄 b は一般的な考え方に拡張された問題である。行③及び④は括弧'(', ')'のときの nest の定数分増減処理である。〔プログラム（計算処理の部分）〕行番号 32，33 で既に確認したように，Priority[]の値が大きい演算子から計算するため，Priority[]の値は一つ深い括弧の中の演算子がその外側の演算子より大きい値となるように設定しなければならない。

nest に加減算する定数と，演算子'＋', '−'のとき加算する値 priLow，演算子'×', '÷'のとき加算する値 priHigh の大小関係は図 B のようになる。

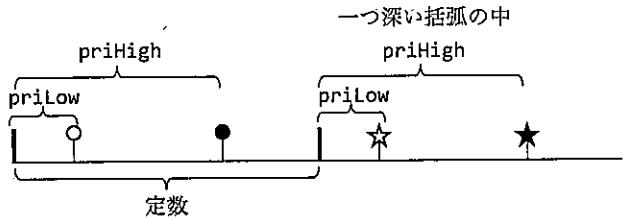


図 B　増減する定数，priLow，priHigh の関係

○が演算子'＋', '－'の Priority[]の値，●が演算子'×', '÷'の Priority[]の値を示し，☆が一つ深い括弧の中の演算子'＋', '－'の Priority[]の値，★が一つ深い括弧の中の演算子'×', '÷'の Priority[]の値を示す。演算子の優先順序から priHigh＞PriLow である。

この関係を常に成り立たせるためには，

☆の値＞●の値

とならなければならない。すなわち，

定数＋priLow ＞ priHigh

したがって，定数 ＞ priHigh － priLow

この関係から，定数は priHigh－priLow＋1 以上でなければならない。したがって，空欄 b には，「priHigh－priLow＋1 以上」（エ）が入る。

［設問2］

・空欄 c：計算する演算子を決定するのは，⑦の比較によって ip を設定する部分である。Priority[ip]＝Priority[i]の場合は条件を満足せず ip の値は更新されないため，最初の値すなわち左から順に実行する。

これに対し，右から順に実行する場合は，等しい場合も変換して，ip に i を設定するようにすればよい。したがって，（エ）の「行 ⑦ の内容が Priority[ip]＜Priority[i] か Priority[ip]≦Priority[i] で決まる」を選べばよい。

（ア）は Priority[0]との比較が行われない。（イ）は i＝OpCnt のとき，Priority[ ]が処理対象外の範囲を処理する。（ウ）は i を 1 から OpCnt－1 まで変化させる代わりに，OpCnt－1 から 1 まで 1 ずつ減じて変化させるため同じ処理となる。

・空欄 d：演算を右から順に実行すると，それぞれのケースは表 A のような式に書き換えられる。

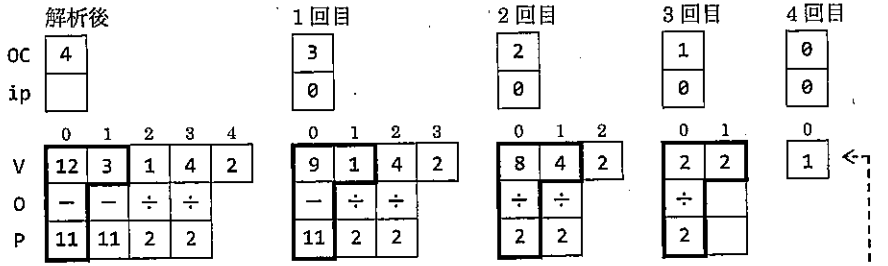
表 A

	左から順に実行	右から順に実行
ケース 1	(12＋3＋1)×4×2＝128	(12＋(3＋1))×(4×2)＝128
ケース 2	(12＋3＋1)÷4÷2＝2	(12＋(3＋1))÷(4÷2)＝8
ケース 3	(12－3－1)×4×2＝64	(12－(3－1))×(4×2)＝80
ケース 4	(12－3－1)÷4÷2＝1	(12－(3－1))÷(4÷2)＝5

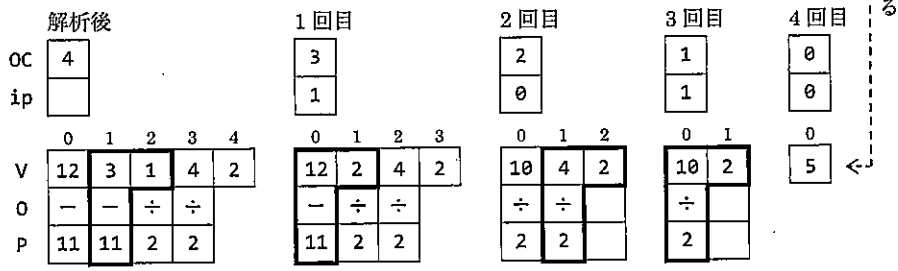
加算と乗算は順序が変わっても値は変わらないが，減算，除算は計算結果が順序によって変わるため，このプログラムでは，「ケース 1」だけが左右どちらから実行しても結果が等しくなる。したがって，（ア）が正解である。

次に，ケース 4 の左右から実行した計算部分の行番号 30～56 の繰返し処理を終えた計算処理後の配列を示す。OC は OpCnt，V は Value[]，O は Operator[]，P は Priority[]の値を示す。太線で囲んだ部分が次のループの添字 ip で示される演算子の計算対象となる。

（ケース４　左から）

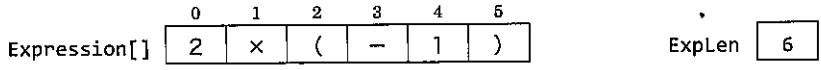


（ケース４　右から）



［設問３］

負数を扱うときにプログラムでポイントとなる点は，行番号７及び行番号21の Value[0]，Value[OpCnt]に０を設定する処理である。整数式２×（−１）を解析処理，計算処理した場合を次に示す。



行	chr	OpCnt	nest	添字	0	1	2	3	4
6〜8		0	0	Value[]	0				

i=0

12	2	0	0	chr が数字　行番号 11 の条件が真 Value[]	2				

i=1

18	×	0	0	chr が演算子　行番号 14 の条件が真 行番号 16 の条件が偽 Operator[]	×				
				Priority[]	2				
※ 21	×	1	0	Value[]	2	0			

i=2

行	chr	OpCnt	nest	添字	0	1	2	3	4
24	(	1	10	chr が括弧　行番号 23 の条件が真					

i=3

17	−	1	10	chr が演算子　行番号 14, 16 の条件が真					
				Operator[]	×	−			
				Priority[]	2	11			
21	−	2	10	Value[]	2	0	0		

i=4

12	1	2	10	chr が数字　行番号 11 の条件が真					
				Value[]	2	0	1		

i=5

27	)	2	0	chr が括弧　行番号 26 の条件が真					

i=6

9	)	2	0	i=6<ExpLen が偽　ループを抜ける					

〔プログラム（解析処理の部分）〕実行後，〔プログラム（計算処理の部分）〕行番号 30〜56 の繰返し処理を終えた後の配列の値を示す。

	解析後			1 回目		2 回目
OpCnt	2			1		0
ip				1		0
	0	1	2	0	1	0
Value[]	2	0	1	2	−1	−2
Operator[]	×	−		×		
Priority[]	2	11		2		

・空欄 e：負数を表す演算子'−'が格納されている ip=1 のときの Operator[ip]に対応する Value[ip]には i=1 の表の※の部分で 0 が設定されている。
　負数を表す演算子'−'の計算は Value[ip]と Value[ip ＋ 1]で行われるため，0−(符号の次の数値) を計算することになり，正しい結果が求められる。したがって，空欄 e には（エ）の「正しい値を返す」が入る。

空欄 f，g：〔プログラム（解析処理の部分）〕実行後の各配列の内容から空欄 f の Value[1]には（イ）の「0」が，空欄 g の OpCnt には（エ）の「2」が入る。

問9　鉄道模型における列車の運行シミュレーション（C）（H30秋-FE 午後問9）

【解答】

［設問1］　a－カ

［設問2］　b－ウ，c－ア，d－オ

［設問3］　e－イ，f－ア

【解説】

鉄道模型の運行シミュレーションを題材にした問題である。

鉄道模型はレールに電気を流すことで，列車を走行させる仕組みになっている。区間に分けて通電状態を制御する本問のようなプログラムや，ポイントの切替えを自動的に行うプログラムを利用することによって，自動運行が可能になる。

この問題を解く上で用語として登場する，「区間」，「入口」，「出口」，「始発駅」，「終着駅」の意味を正しく把握する必要がある。〔列車を進行させるルール〕における定義付け以外にも，図1や表1，及び〔プログラム1の説明〕にある構造体block_infoの定義を見ると，次の要点が把握できる。

・図1の路線構成の上を列車は左から右にだけ移動する。
・駅は始発駅と終着駅だけで，中間駅はない。
・区間情報はプログラム上block_infoという構造体で表現され，分岐をもつ単方向リストになっている。

［設問1］

〔プログラム1〕で示されたコードの中でも関数set_signals内の空欄aについて解答する。

関数set_signalsは，〔プログラム1の説明〕(3)の機能に「全ての区間の信号機それぞれについて，区間内に列車がいるときは表示を赤に，列車がいないときは表示を緑にする」とあるため，目的は非常に単純であり，把握しやすいと思われる。区間に列車がいる場合，いない場合というのは，区間を示す構造体block_infoのメンバであるtrainがNULLかどうかで判定できる。そして，空欄aの1行上で，block = &block[i]としていることから，変数blockが，そのとき見ている区間を示す構造体のアドレスであることが分かる。このため，block->trainがNULLであれば緑（定数名はGREEN）をセットすればよい。したがって，「block->train == NULL」（カ）が正解である。

［設問2］

関数proceedを題材にした設問である。ただし〔プログラム2〕を見ていくと，関数find_blockが重要であると分かる。関数find_blockは〔プログラム2の説明〕(2)の機能に「引数blockで示す区間が，引数destで示す区間に最終的に到達できる経路上にあるかどうかを判定する」と記述されている，これを図1や表1と照らし合

わせて考えると，ある区間の出口にその次区間が二つあるときに，どちらが終着駅に向かう区間なのかを判定するために使われる関数だと読み取ることができる。そして，区間と終着駅を引数として渡し，到達できるかどうかを1か0という数値で答える関数である。

・空欄b：このif文の結果が真のときの処理内容を見ると，「block->train = NULL;」と「continue;」となっている。一方，空欄bの3行上にあるif (block->train == NULL)で始まるif文の処理を見ると，そちらが対象区間に列車がいないときの処理であると分かる。すなわち対象区間に列車がいる場合，かつ空欄bの式が成立するとき（真になるとき）にblock->trainをNULLにしている。対象区間に列車がいて列車がその区間からいなくなるのは，列車が次区間に移動するときと考えられる。

しかし，その部分の処理は，空欄bの4行下にあるfor (j = 0; j < 2; j++)で始まるif文のブロック内で行われている。

このため，そのほかのケースで列車が区間からいなくなるケースを思い出す必要がある。すると，〔列車を進行させるルール〕(6)の②（ア）に「終着駅を出口とする区間に列車がいる場合は，(4)に従って列車を進行させ，路線上から取り除く」という記述が該当すると分かる。すなわち該当区間の出口と，そこにいる列車の終着駅が同じであればよい。列車の終着駅は構造体block_infoのアドレスという形でtrain->destに格納されているが，出口に終着点をもつ区間のアドレスがその列車のtrain->destに格納されている。このため，対象区間であるblockと，そこにいる列車の終着駅block->train->destが一致する場合は，列車を進行させ，その区間から列車を取り除く。したがって，「block == block->train->dest」（ウ）が正解である。

・空欄c：この空欄が属するif文ブロック内は列車を次区間に移動させる処理である。block->next[j]が列車の進行先となる次区間である。選択肢には制御文である，break，continue，returnがある。returnを入れてしまった場合は関数proceedから抜けてしまうので，残りの区間の処理ができなくなり，ふさわしくない。次にbreakかcontinueだが，進行先の区間への移行が済んだためbreakによって，「for (j = 0; j < 2; j++)」のループを抜ける処理がふさわしい。したがって，「break」（ア）が正解である。

一見すると，無駄な処理が増えることになるが，continueでも正しい動作をするようにも思われる。しかし，jが0の時点で列車を次区間に移動させたとすると，jが1のときにif (find_block(block->next[j], block->train->dest) == 1)を評価するタイミングでblock->trainがNULLになるためアクセス違反になり，プログラムが異常終了することが想定される。このため，「continue」（イ）は不正解である。

午後解答

・空欄 d：関数 find_block 内で，関数 find_block 自身を再帰的に呼び出す際の引数が問われている。ここで次区間となる block->next の添字としては i が使われている。すると，次区間 block->next[i]が終着駅 dest に到達できるかどうかを再帰的に呼び出していけばよいと分かる。したがって，「block->next[i], dest」（オ）が正解である。

［設問3］

図1で示された列車の位置から，プログラム3を実行し終えた際の列車4のいる区間とその次区間についてのトレース力が問われている。

関数 start 内では，まず，関数 set_signals が最初に呼ばれ，列車の位置に合わせて，信号の初期状態がセットされる。続いて for (i = 0; i < 4; i++)の中で関数 proceed と関数 set_signals が呼ばれている。つまり，列車の進行と信号の再セットを4回繰り返されることが分かる。ここまでが分かれば，プログラムコードは見なくてもトレースができる。

なお，関数 proceed は，1. 終着駅に近い方の区間から列車を進行するのか，2. 始発駅に近い方の区間から列車を進行するのか，気になると思われる。2. の場合はいわゆる渋滞を引き起こしてしまう。この点について，［プログラム2］を見ると，for (i = nblocks - 1; i >= 0; i--) とあるので，図1と照らし合わせると，1. の考え方に近い，区間番号が大きい方から処理していくことが分かる。また，信号の一斉設定は全ての列車進行が済んでから行われるため，注意が必要である。

1回目：列車0が終着駅に到着し，取り除かれる
　　列車1が区間5に進行
　　列車2が区間4に進行したいが，区間4の信号が赤なので進行しない
　　列車3が区間2に進行
　　※ここで区間10の列車4の進行先も区間2だが，区間1から進行する列車3が優先される。
2回目：列車1が終着駅に到着し，取り除かれる
　　列車2が区間4に進行
　　列車3が区間3に進行したいが，区間3の信号が赤なので進行しない
　　列車4が区間2に進行したいが，区間2の信号が赤なので進行しない
3回目：列車2が区間6に進行
　　列車3が区間3に進行
　　列車4が区間2に進行したいが，区間2の信号が赤なので進行しない
4回目：列車2が終着駅に到着し，取り除かれる
　　列車3が区間4に進行
　　列車4が区間2に進行

これでプログラム3は終了する。すると，列車4は区間2にいて，次区間は区間3の一つだけであり，区間3の信号は緑（列車はいない）。したがって，空欄 e は「2」（イ），空欄 f は「一つあり，その信号機は緑を表示している」（ア）が正解である。

問10　社内資格の保有状況の管理（COBOL）　(H30秋-FE 午後問10)

【解答】

[設問1]　a－ウ，b－エ，c－イ

[設問2]　d－エ，e－エ

[設問3]　ア

【解説】

従業員の社内資格の保有状況を管理するプログラムである。プログラムは索引ファイル，SORT文を利用しているが，テーマは一重のコントロールブレイク処理と1次元の表操作である。どれもCOBOLでは基本処理の一つであり，理解していなくてはならない。合格点が取れるかどうかは，変更前のプログラムの全体構造と内容を正確に理解できたかどうかによる。

空欄aが間違えやすいが，設問2はPERFORM文の終了条件，設問3は上期，下期の意味が理解できていれば解答できるだろう。

変更前のプログラム全体の流れ，処理概要は次のとおりである。

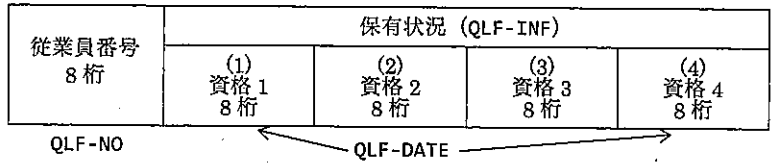
① 合格ファイル（PAS-FILE）を整列併合用ファイル（SRT-FILE）に引き渡し，従業員番号（SRT-NO）の昇順に整列する。合格ファイルのレコードは順不同に作成されており，1人の従業員が複数の資格試験に合格した場合は，保有資格ファイルの1レコードにまとめて合格日を格納しなければならない。そのためは，合格ファイルのレコードを従業員番号で整列し，各レコードをコントロールする必要がある。

② 整列併合用ファイル（以下，整列ファイルという）からレコード（合格ファイルとレコード様式が同じ）を引き取り，あらかじめ入力しておいた資格保有ファイルの保有状況欄に合格日を格納する。これを従業員番号が変わるまで繰り返す。

③ 一つの従業員番号の合格日の格納処理を終えたら，次の従業員番号のレコード処理へ移る。

④ ②，③を整列ファイルのレコードがなくなるまで繰り返す。

保有資格ファイルのレコード様式と意味は図Aのとおりである。



注記：資格1～4（QLF- DATE）へは，資格種別（1～4）を添字として格納する。

図A　保有資格ファイルのレコード様式と意味

[設問1]

・空欄a：整列ファイルからのレコード入力が終了したときの処理である。行番号32はSET文を利用して，行番号31のPERFORM文を終了に導く処理であることがと分かる。このPERFORM文の終了をもって全ての処理を終えるが，行番号33はSET文（ファイル処理の終了）の処理の後に必要な処理があることを示している。

プログラム全体の構造は，冒頭で①～④として解説したが，同じ従業員番号をもつレコードを一つにまとめる必要があるので，一重のコントロールブレイク処理となる。コントロールキー（従業員番号）の保存は行番号42で行っている。整列ファイルからの最後のレコードを処理した後，その従業員番号をもつ保有資格ファイルのレコードの出力（書換）が必要である。書換処理は，REWRITE文があるので，行番号45～48で行っている。したがって，手続WRI-PROCを実行するための「PERFORM WRI-PROC」の（ウ）か，「REWRITE QLF-REC」の（エ）が該当することが分かる。どちらも最後に編集したレコードを出力する内容なので，整列ファイルからのレコード入力が終了した場合では同じ機能を有する。

ここで，行番号46のIF文の意味を考えてみると，行番号21で定義されたCR-NOの初期値は空白（SPACE）であり，最初の1件目のレコードを入力した際に行番号47のREWITE文を回避する内容であると分かる。2件目以降のレコードが入力されると，行番号42でCR-NOには従業員番号が格納されるため，それ以降は空白（SPACE）になることはない。そのため，整列ファイルからのレコード入力が全て終了したときの書換処理では（ウ）でも（エ）でもよいことになる。

そこで，1件目のレコード入力に着目してみると，レコードが存在するなら，行番号47を回避して正しく動作することが分かる。では，レコードが1件も存在しない場合はどうかというと，行番号32のRETURN文が実行された後，AT END句によって空欄aが実行されるが，（ウ）ではCR-NOに空白が設定されているままなので，行番号47のREWRITE文は実行されず終了できる。しかし，（エ）では資格保有ファイルからの入力がないまま，書換処理を実行してしまい，INVALID指定もないからエラーとなり，正しい動作にならない。したがって，（ウ）が正しいことが分かる。

・空欄b，c：手続UPD-PROCの処理内容なので，一連で考える。行番号38のIF文の条件が真の場合は，行番号39～42の文を実行した後に行番号44の文を実行する。逆に偽であれば，行番号44の文だけを実行して終了する。

冒頭の①～④で解説したように，整列ファイルからのレコードには，同一の従業員番号が複数発生し得るから，資格保有ファイルへのレコードの書換処理は，従業員番号が変わったタイミングとなる。そのためには，直前に入力したレコードと現レコードの従業員番号が等しいかどうかを判定する必要があり，

その条件判定が行番号38の内容であると分かる。従業員番号が等しい場合は，複数の資格に合格した従業員がいたということだから，従業員番号が変わるまで合格日の格納を繰り返す。

従業員番号の保存領域は，行番号21の定義及び行番号42から，CR-NOと理解できる。従業員番号が変わったら，現レコードの書換（出力）処理を行い，次のレコードを資格保有ファイルから入力する。この一連の作業が行番号39～42の内容であると分かる。

したがって，空欄bは従業員番号が変わったという条件「SRT-NO NOT = CR-NO」の（エ）が正解である。（ア），（ウ）は従業員番号の比較内容ではなく，（イ）は従業員番号が同じという条件なので誤りである。

空欄cは合格日の格納処理であり，整列ファイルのレコードにおける合格日はSRT-DATE，格納先である保有状況は1次元の表で，添字（SRT-CD）を用いて格納すればよいから，「MOVE SRT-DATE TO QLF-DATE(SRT-CD)」の（イ）が正解である。

（ア）は表の要素を右に一つ移動しているだけであり，（ウ），（エ）も合格日の格納にはならないので誤りである。

午後解答

［設問2］

変更内容は，社内資格の四つを全て取得した従業員を表彰するためであり，社内資格を四つ全て取得したかどうかの判定処理が必要となる。

行番号21と22の間に追加されたW-CNTは，空欄dで追加されたPERFORM文のUNTIL条件に，初期値を1として4回繰り返すこと（W-CNT > 4）の記述があるので，社内資格を四つ全て取得したかどうかの回数判定を行うための変数であることが分かる。

・空欄d：社内資格の四つを全て取得している場合，保有状況欄には合格日が全て格納されている。一方，取得していない資格がある場合は，保有状況欄の該当要素に0が格納されている。したがって，社内資格を四つ全て取得（合格）しているかどうかの判定処理は，保有状況欄の要素に0が格納されているか否かで判定できる。一つでも0が格納されていれば，社内資格の四つを全て取得していない。この判定処理は表の要素数だけ行えばよく，その条件が「W-CNT > 4」の意味である。

また，「QLF-DATE(W-CNT) = ZERO」は，該当要素に0が格納されており，取得していない資格があることを意味している

これらの判定は，一つの従業員番号の合格日の格納処理が全て終わった後でなければならないから，資格保有ファイルへの書換処理を行えるタイミングとなる。その視点で空欄dの解答群を考察すると，次のようになる。

ア：行番号33と34の間……整列ファイルからのレコード入力が全て終えたタイミングなので，正しくない。

イ：行番号35と36の間……整列ファイルからのレコード入力処理を繰り返し

ているタイミングで，保有状況欄への格納が全て終了していないため，正しくない。

ウ：行番号 42 と 43 の間……行番号 40～42 の処理は一つの従業員番号の資格保有ファイルへの書換処理が終わり，次の従業員番号の合格日格納処理を行うための前処理である。このタイミングは次の新しいレコードを入力した直後なので，正しい判定はできない。

エ：行番号 46 と 47 の間……行番号 47 は書換処理であるから，このタイミングで行えば，全ての要素について 0 が格納されているかどうかの判定が可能である。

したがって，正解は（エ）である。

・空欄 e：追加された PERFORM 文の UNTIL 条件には，①，②の二つがある。

① W-CNT > 4

② QLF-DATE(W-CNT) = ZERO

①は四つの要素を全て判定したかどうかであり，②は取得していない資格があった場合である。このどちらかの条件で PERFORM 文は終了する。

PERFORM 文を抜けた後にある IF 文の内容を見ると，条件が真のときに表示（DISPLAY）しているので，これは社内資格の四つを全て取得している場合であり，①が該当する。したがって，正解は「W-CNT > 4」の（エ）である。

（ア）は添字（CR-NO）が正しくない。（イ）は取得していない資格があった場合である。（ウ）は「ALL ZERO」を用いているが，四つの要素に 0 と合格日が混在していても条件を満たすので誤りである。

［設問３］

行番号 44 と 45 の間に追加された ADD 文は，1 次元の表の要素 W-NUM に，資格種別（SRT-CD）を添字として集計（＋1）している。このことは，各試験に合格した従業員の数を意味している。また，プログラムの実行は 10 月初めなので，集計内容は上期が該当することが分かる。このことを念頭に，各選択肢を吟味していけばよい。

ア：特に誤りの箇所は見当たらない。

イ：受験した従業員との記述があるが，合格した内容の処理なので誤りである。

ウ：（ア）とほぼ同じような内容であるが，「上期までに」に注意する。〔プログラムの説明〕に，資格保有ファイルの管理は，上期（4 月～9 月）と下期（10 月～翌年 3 月）の 2 期から成り，1 期分を蓄積すると記述されているので，過去からの蓄積ではないことが分かる。したがって，「上期までに」は誤りである。

エ：「合格していない」との記述があるので，誤りである。

したがって，正解は（ア）である。

問 11　書式を表すひな形への置換表の適用による文書の作成（Java）

（H30 秋-FE 午後問 11）

【解答】

［設問 1］　a－ウ，b－ウ，c－オ，d－ウ，e－エ

［設問 2］　f－オ

【解説】

文書の書式を表すひな形に置換表を適用して，出力文書を得るプログラムの問題である。問題文ではフラグメント（Fragment）が使われており，〔プログラムの説明〕(3)の後には「ひな形を，0 個以上の置換指示と 0 個以上の置換指示以外の部分が連なる文字ストリームとして扱う。個々の置換指示及び個々の置換指示以外の部分をフラグメントと呼ぶ」と記述されているように，ひな形となる文字データの中の，いわば句に当たる。本問ではこれを扱うためのインタフェース Fragment が定義されている。ちなみに，この問題のフラグメントは，Android で頻繁に利用されるクラス Fragment とは関係がない。

この問題におけるフラグメントとしては，前述のように置換指示と置換指示以外が定義されている。具体的には，図 1 のひな形の<名前>，<明細>，<合計>が置換指示のフラグメントである。また，「様」，「様のお買い上げ明細は次のとおりです。」，「------（以下略）」，「合計：」，「円」が置換指示以外のフラグメントとなる。なお，置換指示以外のフラグメントは，ひな形の文字ストリームに置換指示が現れるところまでで一つのフラグメントとなる。

問題文の要点を次に示す。

- フラグメントを表すインタフェース Fragment が定義されている。
- インタフェース Fragment の実装クラスとして，置換指示を表すクラス Replacer，及び置換指示以外を表すクラス PassThrough が定義されている。
- 問題で定義されている機能を実装するために，クラス TemplateParser，クラス Template，クラス ReplacementTableParser が定義されている。
- クラス TemplateParser には，ひな形の文字ストリームからフラグメントのリストを構築し，クラス Template のインスタンスを生成して返す機能を提供するメソッド parse が定義されている。クラス Template は，ひな形をフラグメントのリストとして保持する。
- ひな形に置換表を適用して出力文書を文字列で返すメソッド apply が定義されている。
- クラス ReplacementTableParser には，置換表を表す文字ストリームから，キー名称とそれに対応する文字列のマップを構築して返す機能を提供するメソッド parse が定義されている。なお，クラス ReplacementTableParser は仕様の定義だけで，プログラムは記載されていない。
- クラス TemplateTester はテスト用のプログラムである。

なお，設問2では，エスケープシーケンス文字がテーマになっており，知識がないと混乱したかもしれない。それ以外は問題で示されているアルゴリズムも空欄の構文も理解できたと思われる。

本問では，ファイルを扱うクラス FileReader が利用されているが，使われている機能は簡単なものである。また，基本情報技術者試験の午後問題の冊子には「Java プログラムで使用される API の説明」が収録されているので，仕様を確認できる。

［設問1］

プログラムの穴埋め問題である。空欄 a は，インタフェースを実装するクラスの宣言構文を知っていれば解答できる。空欄 b，c は問題文とプログラムをよく読み，フラグメントのリストに追加する適切なクラスのインスタンスは何型であるか考えて解答する。空欄 d，e は，コンストラクタやメソッドを呼び出す際の，仮引数と実引数の型に関する知識があれば解答できる。

・空欄 a：〔プログラム 2〕のクラス Replacer，及び〔プログラム 3〕のクラス PassThrough のクラス宣言構文に空欄 a がある。空欄の右側に記述されているのは，インタフェース Fragment である。クラス宣言構文でインタフェース名の直前に記述されるのは，Java 言語のキーワードである「implements」である。したがって，（ウ）の「implements」が正解になる。

・空欄 b：クラス TemplateParser のメソッド parse 内の switch 文で，reder.read() で読み込んだ文字が '<' だった場合に，fragmentList にインスタンスを追加するメソッド add の引数に記述された new 演算子に続く構文である。まず，〔プログラムの説明〕には「メソッド parse は，ひな形を表す文字ストリームからフラグメントのリストを構築し」とある。reder.read()がストリームから文字を読み込む処理である。読み込んだ値は変数 c に格納している。空欄 b を含む switch 文の式には，この変数 c が記述されており，読み込んだ文字ごとに処理を振り分けている。

次に変数 fragmentList に着目する。変数 fragmentList は，List<Fragment> 型として宣言されており，Fragment 型のインスタンスを格納するリストである。空欄 a で見たように，Fragment 型のインスタンスとして生成できるのは，インタフェース Fragment を実装したクラス Replacer，又はクラス PassThrough のいずれかである。空欄 b が含まれる case ではどちらのクラスのインスタンスを生成するべきか考える。空欄 b が実行されるのは，変数 c の値が '<' の場合である。文字 '<' は，ひな形でキー名称の範囲の開始を示す文字である。見方を変えると，置換指示以外の部分の終了である。このことから，fragmentList に追加するのはクラス PassThrough のインスタンスであればよいことが分かる。

クラス PassThrough のコンストラクタの引数について考える。まず，コン

ストラクタの宣言構文を見ると，引数は StringBuilder 型であることが分かる。メソッド parse 内で，StringBuilder 型の変数は buf だけなので，この時点でコンストラクタの実引数は変数 buf であればよい。確認のため，変数 buf に格納される値について見ていく。変数 c の値がキー名称の範囲の開始と終了を示す '<' と '>' 以外の場合，default に処理が振り分けられている。この場合，変数 buf を利用してメソッド append を呼び出し，読み込んだ文字を順次追加している。この処理で，変数 buf に一括りの置換指示以外のテキストが格納されていることになる。これらのことから，変数 buf を実引数とするクラス PassThrough のコンストラクタの呼出しの構文になればよい。したがって，（ウ）の「PassThrough(buf)」が正解になる。

・空欄 c：空欄 b と同様にクラス TemplateParser のメソッド parse 内の処理であるが，空欄 c では reader.read()で読み込んだ文字が '>' だった場合に追加するインスタンスを生成する new 演算子に続く構文である。文字 '>' は，キー名称の範囲の終了を示す文字である。変数 c の値が '>' の場合，変数 buf にはキー名称のテキストが格納されていることになる。このことから，変数 fragmentList に追加されるのはクラス Replacer のインスタンスであればよいことが分かる。したがって，（オ）の「Replacer(buf)」が正解になる。

・空欄 d：クラス TemplateParser のメソッド parse の戻り値である。メソッド parse の宣言構文を見ると，戻り値は Template 型である。メソッド parse 内の空欄 d 以外の場所で Template 型のインスタンスを生成している箇所はない。このことから，空欄 d は Template 型のインスタンスを生成する構文であればよいことが分かる。クラス Template のコンストラクタの宣言文を見ると，引数の型は List<Fragment>型となっている。メソッド parse 内で，List<Fragment>型であるのは，変数 fragmentList だけであり，実引数として変数 fragmentList を渡してコンストラクタを呼び出す構文であればよいことが分かる。したがって，（ウ）の「new Template(fragmentList)」が正解になる。

・空欄 e：クラス Template のメソッド apply 内の処理の一部である。空欄 e は，インタフェース Fragment で定義されているメソッド replace を呼び出す際の実引数である。インタフェース Fragment のメソッド replace の宣言文を見ると，引数の型は Map<String, List<String>>と宣言されている。メソッド apply 内で，Map<String, List<String>>型で宣言されているのは，仮引数の table だけである。これらのことから変数 table を実引数とすればよいことが分かる。したがって，（エ）の「table」が正解になる。

［設問2］

クラス TemplateParser のαの位置に挿入するコードの空欄の穴埋め問題である。挿入するコードは switch 文の case で変数 c の値が '\\' の場合の処理である。

まず，char 型のリテラルは原則として，シングルクォーテーション内に 1 文字しか記述できないはずであるのに，'\\' と 2 文字記述されていることに注意が必要である。\（バックスラッシュ）は，\ と次に続く 1 文字の組でエスケープシーケンス文字と呼ばれる，特殊な文字を表現する場合に用いられる文字である。代表的な例は '\n' で，\ と n で 1 文字と見なし，改行を表す文字になる。このように，\ は通常，次に続く文字と組み合わされて解釈されてしまう。このため，\ を単なる文字として扱いたい場合，'\\' とバックスラッシュを 2 文字続けて表現する。これらのことを踏まえて挿入されるコードを見ると，変数 c の値が \（バックスラッシュ 1 文字）だった場合の処理であることが分かる。なお，Windows の日本語版など環境によっては¥で表示される。

設問文から追加するコードの仕様を確認すると「\ に続く文字（\ が複数個連続するときは奇数個目に続く 1 文字）は，置換指示以外の部分やキー名称の一部として扱われる」とある。空欄 f は変数 c の値が '\\' の場合に実行されるので，\ に続く 1 文字を読み込み，変数 buf に対しメソッド append を呼び出して追加すればよい。これをコードで記述すると「buf.append((char) reader.read());」となる。なお，インタフェース Reader のメソッド read の戻り値の型は int 型であるが，クラス StringBuilder のメソッド append の引数は char 型であるため(char)で明示的にキャストしている。

次に，挿入されるコードの位置に着目する。挿入位置αは switch 文の case の default の前である。このため，前述のコードに加えて「break;」を記述しないと，default に記述している処理も実行されてしまうことになる。つまり「buf.append((char) reader.read()); break;」であればよい。なお，;で区切られた複数のステートメントを 1 行に記述しても構文上問題はない。

したがって，（オ）の「buf.append((char) reader.read()); break」が正解になる。

問12　日数の計算（アセンブラ）　(H30秋-FE 午後問12)

【解答】

［設問1］　a－イ，b－カ，c－イ，d－カ

［設問2］　e－エ，f－ウ

【解説】

1970年1月1日（基準日）から，指定された日付（年，月，日）までの日数を求めるプログラムについて考える問題である。論理演算や指標レジスタを利用したアドレス操作がどのような目的で使われているかを考えながらプログラムを読み解いていく。空欄はコメントを参考にしたり，〔プログラムの説明〕(2)のうるう年判定を確認したりしながら解答する。

〔プログラムの説明〕から，使用される副プログラムに渡される値と設定される値を次にまとめる。

・副プログラム DAYOFFST（プログラム1）

+0	+1	+2
年	月	日

GR2：↵

GR0：基準日（1970年1月1日）から渡された日付までの日数を設定する。

・副プログラム LEAPYEAR（プログラム2）

GR2：年

GR0：渡された年が平年の場合には0，うるう年の場合には1を設定する。

・副プログラム DIVISIBL

GR2：整数値

GR3：整数値

GR0：GR2÷GR3が割り切れる場合には1，割り切れない場合には0を設定する。

〔プログラムの説明〕(1)に，プログラム1は副プログラム DAYOFFST であり，日付の年，月，日には誤りがなく，「基準日から65535日目までの日付で与えられる」。日数は「符号のない数値（0～65535）」とするため計算には論理加算（ADDL 命令），論理減算（SUBL 命令）を使うことになる。

［設問1］

〔プログラム1〕から確認していく。まず，行番号25，26に注目すると，コメントに「ACCMDAYS は，平年の各月1日の1月1日からの日数」とある。定数として ACCMDAYS が定義されており，次のようになる（図A参照）。

1月1日……0日
2月1日……0日＋31日（1月）
5月1日……31日（1月）＋28日（2月平年）＋31日（3月）＋30日（4月）＝120日

	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+10	+11
ACCMDAYS	0	31	59	90	120	151	181	212	243	273	304	334
月	1	2	3	4	5	6	7	8	9	10	11	12

図A　ACCMDAYSの値と月，アドレスの関係

・行番号1，2：レジスタ内容の退避。
・行番号3：(GR2)＋0番地の内容（年）をGR5に格納。
・行番号4：(GR2)＋1番地の内容（月）をGR3に格納。
・行番号5：(GR2)＋2番地の内容（日）をGR1に格納。
・行番号6：GR1に日数を求めるため，GR1から基準日分の1日を引く。
・行番号7～13：コメントから，渡された月日の1月1日からの日数をGR1に求める。
　行番号8：(GR4)－1番地の内容をGR1に加算。
　行番号9，10：GR3が3より小さければSKIPへ分岐。1月，2月はうるう年を考慮しなくてよいため。
　行番号11：GR5（年）をGR2に格納。副プログラムLEAPYEARの実行準備。
　行番号12：副プログラムLEAPYEARを実行。
　行番号13：GR1に，うるう年判定後のGR0に設定された1か0を加算。

行番号13で月日の算出が終了し，以降は年の処理となる。

・空欄a：行番号8のADDL命令には値を設定していないGR4が指標レジスタとして使用されているので，行番号7はGR4の設定と考えられる。また，1月1日からの日数を求めるために定数ACCMDAYSが定義されている。渡された月の1日までの日数を示すデータは図Aから，次の番地になる。
　　ACCMDAYSのアドレス＋月（GR3の値）－1番地
　行番号8の「ADDL　GR1,-1,GR4」のアドレス指定部分の番地には－1を，指標レジスタにはGR4を用いて(GR4)－1番地を指定している。GR1に1月1日からの日数（平年）を加算するには，GR4に行番号7でACCMDAYSのアドレス＋月数（GR3の値）を設定すればよい。したがって，空欄aには「LAD GR4,ACCMDAYS,GR3」（イ）が入る。

・行番号14～21：年部分の日数加算処理。1970年から渡された年までの日数をループしながら加算。
　行番号14：（ラベルSKIP）GR2に1970を格納。
　行番号15：（ラベルLOOP）GR5とGR2を比較。

行番号16：ループを抜ける分岐命令。
行番号17：副プログラムLEAPYEARを実行。
行番号18：GR0に，平年の365を加算，うるう年なら366を加算。
行番号19：日数合計GR1にGR0を加算。
行番号20：年GR2に1を加算。
行番号21：行番号15　LOOPへ分岐。
行番号22：（ラベルBREAK）GR0に，求めた日数GR1を格納。
行番号23：（ラベルEXIT）退避したレジスタの内容を復元。

・空欄b：行番号15～21のループを抜ける条件分岐命令が入ると考えられる。GR2に初期値1970を設定し，行番号20で1加算しているので，GR2は1970年から与えられた年まで増分1で順に増加する。行番号15でGR5と比較しているが，GR5に設定された年については月日の計算で終了しているので，GR5の1年前まででよい。すなわち，GR2<GR5の間，ループし，GR2=GR5となった時点で終了する。したがって，空欄bには，「JZE　BREAK」（カ）が入る。

〔プログラム2〕はうるう年を判定するプログラムである。〔プログラムの説明〕(2)の①～④を順に判断し，うるう年かどうかを判定する。

・行番号1，2：レジスタ内容の退避。
・行番号3：GR0－GR0を行い，GR0を0（平年）で初期化。
・行番号4：GR2（年）をGR3に格納。
・行番号5，6：(2)の①の判定。
　行番号5：GR3と3=0000 0000 0000 0011をAND演算。
　行番号6：GR3が4で割り切れない場合はFINへ分岐。
・行番号7～10：(2)の②の判定。
　行番号7：GR3に100を格納。
　行番号8：副プログラムDIVISIBLの実行。GR2が100で割り切れるか判定。
　行番号9：100で割り切れない場合は副プログラムDIVISIBLでGR0に0を設定して戻り，うるう年となるため副プログラムLEAPYEARの結果としてGR0に1を設定し，割り切れる場合はGR0に1を設定して戻り，次の判定へ進むためGR0に0を設定する。
　行番号10：GR0が0でなければFINへ分岐。
・行番号11，12：(2)の③，④の判定。
　行番号11：GR3に400を格納。
　行番号12：副プログラムDIVISIBLの実行。GR2が400で割り切れるか判定。
・行番号13：（ラベルFIN）退避したレジスタの内容を復元。GR2が400で割り切れる場合はうるう年で，副プログラムDIVISIBLでGR0に1を設定して戻り，副プログラムLEAPYEARの結果のGR0も1である。それ以外は平年と

午後解答

するため割り切れない場合は GR0 に 0 を設定して戻り，副プログラム LEAPYEAR の結果の GR0 も 0 である。副プログラム DIVISIBL からの GR0 の値がそのまま副プログラム LEAPYEAR の結果となるため終了できる。

・空欄 c：(2)の①は「年が 4 で割り切れない場合，平年とする」である。年は行番号 4 で GR3 に格納されているので，GR3 が 4 で割り切れなければ副プログラム LEAPYEAR の結果として GR0 に 0（平年）を設定してプログラムを終了すればよい。2 進数の各桁と 10 進数の対応は最下位ビットから上位ビットに向かってビット 0 が 1，ビット 1 が 2，ビット 2 が 4，ビット 3 が 8……であり，ビット 2～ビット 15 は全て 4 の倍数なので 4 で割り切れる。したがって，4 で割り切れるかどうかは最下位 2 ビットで判断できる。最下位 2 ビットが 00 ならば割り切れ，00 でない，すなわち 1＝01，2＝10，3＝11 ならば割り切れない。行番号 5 の GR3 と 3＝0000 0000 0000 0011 との AND 命令でビット 15～ビット 2 まではビットごとに 0 との，最下位 2 ビットはビットごとに 1 との AND 演算となる。0 との AND 演算は GR3 の対応するビットの値が 0 でも 1 でも 0 となり，1 との AND 演算は GR3 の対応するビットの値が 0 ならば 0，1 ならば 1 と GR3 のビットと同じとなる。したがって，GR3 にはビット 15～ビット 2 までが全て 0 で最下位 2 ビットは元と同じビットが設定され，GR3 の最下位 2 ビットだけが取り出される。年を格納した GR3 が 4 で割り切れない場合は最下位 2 ビットが 00 でないので，AND 命令実行後の GR3≠0 の場合は平年である。行番号 3 で GR0 には平年を示す 0 が格納されているので，そのまま終了すればよい。したがって，空欄 c には，「JNZ　FIN」（イ）が入る。

・空欄 d：(2)の②は「年が 4 で割り切れ，かつ 100 で割り切れない場合，うるう年とする」である。行番号 7 が実行されるのは 4 で割り切れる場合だけなので，100 で割り切れなければ副プログラム LEAPYEAR の結果として GR0 に 1（うるう年）を設定して終了し，割り切れれば 0（平年）を設定して次の判定に進む。行番号 10 で空欄 d の結果が 0 でなければラベル FIN へ分岐する。

行番号 8 の副プログラム DIVISIBL では，〔プログラムの説明〕(3)にあるように「割り切れない場合は 0 を，割り切れる場合は 1 を，GR0 に設定」しているので，空欄 d では，副プログラム DIVISIBL の結果としての GR0 が 0（割り切れない）なら副プログラム LEAPYEAR の結果として GR0 には 1（うるう年）を，1（割り切れる）なら 0（平年）を設定する演算を選べばよい。選択肢は GR0 と#FFFE＝1111 1111 1111 1110，1＝0000 0000 0000 0001 との AND，OR，XOR 演算である。特定位置のビットを反転し，それ以外のビットは元のままとするには，反転したいビット位置に 1 を，元のままにするビット位置には 0 を設定した値との XOR 演算を行えばよい。GR0 が 0 のときは 1 に，1 のときは 0 にするには，最下位ビットだけをビット反転すればよい。

したがって，空欄 d には「XOR　GR0,=1」（カ）が入る。選択肢の演算を次

に示す。

	GR0＝0	GR0＝1
ア	0000 0000 0000 0000 AND 1111 1111 1111 1110 0000 0000 0000 0000	0000 0000 0000 0001 AND 1111 1111 1111 1110 0000 0000 0000 0000
イ	0000 0000 0000 0000 AND 0000 0000 0000 0001 0000 0000 0000 0000	0000 0000 0000 0001 AND 0000 0000 0000 0001 0000 0000 0000 0001
ウ	0000 0000 0000 0000 OR 1111 1111 1111 1110 1111 1111 1111 1110	0000 0000 0000 0001 OR 1111 1111 1111 1110 1111 1111 1111 1111
エ	0000 0000 0000 0000 OR 0000 0000 0000 0001 0000 0000 0000 0001	0000 0000 0000 0001 OR 0000 0000 0000 0001 0000 0000 0000 0001
オ	0000 0000 0000 0000 XOR 1111 1111 1111 1110 1111 1111 1111 1110	0000 0000 0000 0001 XOR 1111 1111 1111 1110 1111 1111 1111 1111
カ	0000 0000 0000 0000 XOR 0000 0000 0000 0001 0000 0000 0000 0001	0000 0000 0000 0001 XOR 0000 0000 0000 0001 0000 0000 0000 0000

［設問2］

副プログラム DAYOFFST に，基準日から 65536 日目以降 32767 年 12 月 31 日までの日付を与えた場合には，あふれが発生する。この場合にフラグレジスタ FR の OF が 1 の状態で呼出し元に戻るようにプログラムを変更する問題である。

・空欄 e，f：副プログラム DAYOFFST で日数が 65536 を超える可能性がある命令は，行番号 18 で GR0 に設定された平年なら 365 日，うるう年ならば 366 日を 1 年ごとに GR1 に加算する行番号 19 の ADDL 命令である。ADDL 命令は結果が 65535 を超えるとオーバフローになり，フラグレジスタ FR の OF に 1 を設定する。FR の OF が 1 のとき分岐する命令は JOV である。分岐先の行番号 23 の RPOP 命令，行番号 24 の RET 命令は通常フラグレジスタ FR の OF を変化させないためフラグレジスタ FR の OF が 1 の状態で呼出し元に戻る。

したがって，「プログラム 1 の行番号 19 の行の直後に JOV 命令を挿入し，ラベル EXIT の行に分岐する」となり，空欄 e には「19」(エ) が，空欄 f には「JOV」(ウ) が入る。

問13　待ち時間の状況などの分析（表計算）　(H30秋・FE午後問13)

【解答】

［設問1］　a－正解なし（IPAが発表）
［設問2］　b－オ，c－イ，d－イ
［設問3］　e－エ，f－エ，g－ウ

【解説】

表計算のワークシートで窓口の待ち行列の分析を行うことがテーマとなっている。分析に必要なデータから分析結果を表示するとともに，条件を変えた場合の待ち行列の状況をシミュレーションして業務改善に生かすことを目的としている。

本問では，図1「ワークシート“来店状況”の例」（以下，“来店状況”），図2「ワークシート“分析”の例」（以下，“分析”）で示された2種類のワークシートが利用されている。設問1，設問2では，これらの表のセルに入力する式で使用する関数やセルの参照方法について出題されている。出題される頻度が高い関数についてはよく理解しておくことが重要である。加えて，関数の組合せや関数を入れ子にして使用するケースも出題されており，このような関数の使い方にも慣れておく必要がある。

設問3では，本問の目的である窓口数という条件を変えて，待ち行列状況のシミュレーションをマクロで処理する内容となっている。ワークシート内のセル間の値を参照・代入する処理を自動化させる内容である。解答に際しては個々の処理の具体的な内容について問題文に沿って適切に把握していくことが大切である。

［設問1］

“来店状況”のセルI3「待ち人数」を求めるための式が問われている。問題文から，論理式と条件を正しく把握する。ここで取り上げられている関数“条件付個数”については問題文でも説明されているが，しっかり理解しておくことが望まれる。

・空欄a：“来店状況”のセルI3に設定する式の一部を解答する。列I（セルI3～I252）には「待ち人数」が表示されている。待ち人数は，〔ワークシート：来店状況〕の(9)に「当該行の受付番号よりも小さい受付番号をもつ窓口利用者のうち，当該行の受付時刻にまだサービスが開始されていない人数」とある。「サービスが開始されてない」は「開始時刻が，当該行の受付時刻より遅い時刻」と読み換えることができる。受付番号は3行目を1とし4行目以降に昇順に入力されるので，待ち人数は「当該行より上の行の中で，当該行の受付時刻より開始時刻が遅い（＝大きい）行の数」と考えればよい。開始時刻はE列であり，先頭行のセルであるE3から当該行（E3セルには自身のセルを指定する）を範囲とする。このとき，セルI3に入力した式をセルI4～I252に複写することも考慮してセルの式を指定する必要がある。すなわち，先頭行を固定してE$3とし，E$3:E3をセル範囲とする。そうすることで，セルI4ではE$3:E4というよう

に，以下，セル範囲が拡大していく。そして条件を当該行の受付時刻（B列）より大きい「>B3」として関数“条件付個数”を呼び出す式となる。したがって，（ウ）が正解に近い。しかし，平成30年11月5日に，IPAから，この設問は成立しないことが発表されている。

> 平成30年10月21日に実施いたしました基本情報技術者試験午後問題問13設問1において，誤りがあることが判明いたしました。
> 当該設問については，本文と，正解としている選択肢ウの内容が合致していないことから成立しないと判断し，問13を選択した受験者全員について，設問1を正解として取り扱うこととしました。

出典　https://www.jitec.ipa.go.jp/1_00topic/topic_20181105.pdf

（ウ）の選択肢に従って考えると，当該受付に関する行自体の開始時刻も受付時刻と比較してしまい，受付時刻から少し遅れてサービスを開始した行もカウントしてしまうため，適切ではない。本来，E3の行はその日の先頭で受付けた行なので，それより前の受付があるはずもなく，I3は固定値で0でもよい。このため，（ウ）を修正し，「条件付個数(E$3:E3),>B4」とし，

IF(A3=null,null,条件付個数(E$3:E3),>B4)

をI4に設定した上で，I5～I252に複写する必要があるといえる。

他の選択肢も確認しておこう。

ア：セル範囲が，当該行～最終行となってしまい，当該行より後に受付けた行の開始時刻を比較対象としてしまっているため，誤りである。

イ：条件が当該行の受付時刻より前にサービスを開始した行となっているため，誤りである。

エ：セル範囲にG列が指定されており，終了時刻を比較対象としているため，誤りである。

オ：セル範囲と条件式がともに誤りである。

カ：（エ）と同様であり，誤りである。

［設問2］

“分析”の各セルに求める結果を表示するために入力する式が問われている。ここでは，関数の定義と引数を理解した上で，引数として指定する条件を正しく導き出す理解力が求められている。問題文の説明から，条件を正しく把握することが重要である。

・空欄b：“分析”のセルC4に設定する式を解答する。列C（セルC4～C12）は「当該行の時間帯の受付人数」である。(1)には「各時間帯の始まりの時刻（毎時の00分）と終わりの時刻（毎時の59分）」と記述されており，“分析”の受付時間帯の「から」，「まで」が該当する。“来店状況”のデータから次のように算

出する。

“来店状況”の各行の受付時刻が,

①「当該時間帯の「まで」の時刻（B列）以下となるデータ数から,「から」の時刻（A列）より小さいデータ数を減算した結果」

又は

②「当該時間帯の「から」の時刻（A列）以上となるデータ数から,「まで」の時刻（B列）より大きいデータ数を減算した結果」

で表される式を導き出す。

このような範囲内のデータ数を求める場合，ここでも関数“条件付個数”を使用する。セル範囲は“来店状況”の“B$3:B$252”である。解答群の中では,（オ）が①に該当する。

条件付個数(来店状況!B$3:B$252,≦B4)－条件付個数(来店状況!B$3:B$252,＜A4)

したがって,（オ）が正解である。

他の選択肢も確認しておこう。

ア：開始時刻が「10:00」など毎時の00分の受付が含まれないため，誤りである。また，A5（次の行の開始時刻）を指定すると行12のとき，次の行の開始時刻が空白であるため，正しい結果とならない。

イ：開始時刻が「10:00」など毎時の00分の受付が含まれないため，誤りである。

ウ：結果がマイナスになる。

エ：結果がマイナスになる。

・空欄c：“分析”のセルI4の，種別分析におけるサービス種別ごとの「平均サービス時間」を求める式を解答する。この場合は,“来店状況”の列F，セルの範囲としては“F3:F252”のサービス種別ごとの合計を求め，それをサービス種別ごとの件数で除算すればよい。サービス種別ごとの合計を求めるには,関数“条件付合計”を使用する。第1引数の検索セル範囲は“来店状況”の列Cの，セル範囲“C3:C252”，第2引数の検索条件は,“分析”の種別分析のセルF4，そして第3引数の合計のセル範囲には“来店状況”の列Fの,セル範囲“F3:F252”を指定する。入力した式をセルI5～I6に複写するため，ここでもセル範囲の行には絶対参照を指定し，条件付合計(来店状況!C$3:C$252, ＝F4, 来店状況!F$3:F$252)となる。これを，件数である“分析”のセルG4で除算する。

条件付合計(来店状況!C$3:C$252,＝F4,来店状況!F$3:F$252)／G4

したがって,（イ）が正解である。

他の選択肢も確認しておこう。

ア：“来店状況”の全データのサービス時間の合計を，サービス種別ごとの件数と構成比率を掛け合わせた値で除算しており，サービス種別ごとの平均サービス時間が求められるわけではないため，誤りである。

ウ：サービス種別ごとのサービス時間の合計を求めているところは正しいが，全体の件数で除算しているため，誤りである。
エ：関数“条件付合計”の第 1 引数と第 3 引数が逆になっており，誤りである。
オ：(エ) と同様である。かつ全体の件数で除算しており，誤りである。
カ：サービス時間全体の平均値に構成比率を掛け合わせており，正しい平均サービス時間とはならないため，誤りである。

・空欄 d：“分析”のセル G10 の，待ち時間分析における待ち時間順位に該当する「受付番号」を求める式を解答する。この場合は，“来店状況”の列 J の，セルの範囲としては“J3:J252”の順位と一致する行の受付番号である列 A の値を求めることになる。このような場合には，関数“垂直照合”あるいは“照合検索”を使用する。関数“表引き”を用いるケースも考えられる。ただし，この場合は，検索のセル範囲が抽出のセル範囲より右側にあり，検索のセル範囲から，左端から何列目と指定する抽出する列範囲の指定ができない。このため，関数“照合検索”を用いることになる。第 1 引数の検索値は，“分析”の待ち時間分析の順位であるセル F10，第 2 引数の検索セル範囲は“来店状況”の列 J，セル範囲“J3:J252”，そして第 3 引数の抽出のセル範囲には“来店状況”の列 A，セル範囲“A3:A252”を指定する。入力した式をセル G11～G14 に複写するため，セル範囲の行には絶対参照を指定する。

照合検索(F10,来店状況!J$3:J$252,来店状況!A$3:A$252)

したがって，(イ) が正解である。

他の選択肢も確認しておこう。

ア：関数“照合検索”の第 2 引数と第 3 引数が逆であり，誤りである。
ウ，エ：この場合は関数“垂直照合”では正しい結果が得られないため，誤りである。
オ：ここでは，関数“表引き”と入れ子になっている，関数順位について確認すると，その不整合が分かる。

まず，関数順位は第 1 引数に順位を求める対象の計算式 (セルを含む)，第 2 引数に範囲，第 3 引数は順位を昇順：0 で求めるか，降順：1 で求めるかを指定する。

これを踏まえると，第 1 引数には F10 を渡しているが，これは“分析”の例にある「待ち時間分析」部分の「順位」の値であり，F10 では「1」が指定されている。一方，範囲には，“来店状況”の例の H$3~H$252，つまり「待ち時間」が指定されている。このため，関数順位の呼出し自体が意味をなしていないといえる。誤りである。

カ：(オ) と同様に関数“表引き”を用いている。この場合は，表引きの第 1 引数及び第 2 引数で使用している関数“照合一致”の引数がともに正しくないため，誤りである。

［設問３］

マクロに関する設問である。処理自体は比較的分かりやすいといえるが，配列を含め変数が多いことから，問題文の条件とプログラムを読み進めながら，変数の意味とその値の変化を把握することがポイントになる。

なお，〔マクロ：queue_simulation の説明〕（以下，マクロの説明）の(1)にあるように「各窓口でサービスを受けている窓口利用者へのサービスが終了する時刻」を前者終了時刻という。

まず，変数と配列を整理しておく。

・work_line：“来店状況”の当該処理行が設定される変数
・end_time[9]：各窓口の前者終了時刻を格納する配列
・min_time：前者終了時刻の最小の値を格納する変数
・min_no：前者終了時刻の最小の値をもつ窓口番号を格納する変数
・sw：メインループの終了判定フラグとして使用される変数
・i：窓口のループで使用されるカウンタ変数

続いて，マクロの説明の構造の概要を確認する。(1)～(3)の項番は，マクロの説明の項番と一致する。

(1) 初期処理ループで，配列 end_time の初期化を行う。初期化する範囲である窓口数はセル B1 の値を使用する。
(2) 変数の初期化
(3) メイン処理ループで全データの処理を行う。
　① 窓口を決定するための検索処理ループ……条件が空欄 e
　　初期設定として，窓口番号（変数 min_no）と前者終了時刻の最小の値（変数 min_time）の初期値を設定する。続いて，前者終了時刻の最小の値を求めるループに入り，前者終了時刻の中で最小の値をもつ要素を探し，その添字の値を窓口番号として処理行の列 D のセルに格納する。前者終了時刻が等しい窓口が複数あるときは，窓口番号が最小の窓口を選択する。……空欄 f
　② 条件分岐（IF-Else 構造）において，(3)①で求めた当該行の受付時刻が前者終了時刻の最小の値（変数 min_time）より小さい場合は受付時刻を，そうでなければ前者終了時刻の最小の値を，当該行の開始時刻（列 E のセル）に代入する。
　③ (3)①で決定した窓口の前者終了時刻すなわち，配列 end_time の当該窓口に対応する要素に，処理行の終了時刻（列 G）の値を代入する。……空欄 g
　④ 処理行，すなわち変数 work_line に 1 を加え，“来店状況”の受付番号が空白であればメインループを終了する。

・空欄 e：マクロの説明(3)①における，「前者終了時刻の中で最小の値をもつ要素を探し」という条件式を示す式を解答する。各窓口の前者終了時刻は配列 end_time[9]に格納されており，ループ内での現時点での最小の値は変数 min_time に格納されているため，求める式は「min_time ＞ end_time[i]」

となる。したがって，正解は（エ）である。

他の選択肢も確認しておこう。

ア：最小の値を求めるための演算子が逆であり，誤りである。

イ：min_time と end_time[min_no]はともに最小の値が格納されているため，誤りである。

ウ：相対（A2, work_line, 1）は処理行の受付時刻であるため，誤りである。

オ：（イ）と同じく誤りである。

カ：（ウ）と同じく誤りである。

・空欄 f：マクロの説明(3)①における，「最小の値をもつ要素を探し，その添字の値を窓口番号として処理行の列 D のセルに格納する」という処理の「添字の値」を示す式を解答する。ループカウンタは変数 i であるが，変数 i はループの最終値となっている。最小の値を求めた際に要素の添字は変数「min_no」に格納されているため，正解は（エ）である。

他の選択肢も確認しておこう。

ア：end_time[i]は，ループ終了時には，i は窓口数＋1 であり，end_time[i]は未設定である。誤りである。

イ：end_time[min_no]は，窓口番号ではなく最小の値である。誤りである。

ウ：ループカウンタであり，ループ終了時には窓口数＋1 である。誤りである。

オ：min_time は最小の値を格納する変数である。誤りである。

カ：work_line は当該処理行を示す変数である。窓口番号とは無関係であり，誤りである。

・空欄 g：マクロの説明(3)③における，(3)①で決定した窓口の前者終了時刻の最小の値をもつ配列 end_time 要素に，処理行の終了時刻（列 G）の値を代入する式を解答する。配列要素の添字は変数 min_no に格納されており，配列 end_time の当該窓口に対応する要素は，配列 end_time[min_no]で示される。次に，処理行の終了時刻（列 G）は“来店状況”の該当するセルを相対参照する式として表すと，相対(A2, work_line, 6)になる。

```
end_time[min_no] ← 相対(A2, work_line, 6)
```

したがって，正解は（ウ）である。

他の選択肢も確認しておこう。

ア：min_time は終了時刻でなく，開始時刻である。誤りである。

イ：(A2, work_line, 4)は，終了時刻でなく，開始時刻の列を指す。誤りである。

エ，オ，カ：配列 end_time の添字が work_line である点が，そもそも誤りである。

●平成30年度秋期
午後問題 IPA発表の出題趣旨と採点講評

問1

出題趣旨
情報セキュリティ事故発生時には，被害の拡大を防止するとともに，原因を速やかに特定して適切な再発防止策を施す必要がある。中でも，外部からの攻撃によるセキュリティ事故の場合は，事故の原因となり得る脆弱性の内容や攻撃の手口を理解しておくことは，攻撃からの防御，原因の特定及び再発防止策の実施において重要である。 本問は，インターネットを経由した攻撃を題材に，公開しているWebサイトで情報セキュリティ事故が発生した場合の，原因の特定と適切な対策の立案を主題としている。 本問では，Webサイトへの攻撃手法の理解及び事故を踏まえたシステム面での適切な対策を立案する能力を評価する。

採点講評
問1では，インターネットを経由した攻撃を題材に，公開しているWebサイトで情報セキュリティ事故が発生した場合の，原因の特定と適切な対策の立案について出題した。 設問1のSQLインジェクションの脆弱性を悪用する攻撃については，正答率は高く，よく理解されていた。 設問2では，aの正答率は平均的で，おおむね理解されていた。アやイと誤って解答した受験者が見受けられた。インターネットを経由した攻撃の種類によって対策方法が異なるので，具体的な対策について理解しておくことが重要である。bとcの正答率は高く，よく理解されていた。 設問3の正答率は高く，よく理解されていた。Webアプリケーションソフトのセキュアな実装に加え，システムにセキュリティ対策機能を組み込むことによって，より多層的な防御が可能となる。 Webサイトを構築する場合，Webアプリケーションソフトの脆弱性への適切な対策を行う能力は欠くことができないものなので，身につけておいてほしい。

問2

出題趣旨
ソフトウェアに関する技術として，OSの機能の一つである，プロセスをCPUに割り当てる順序を決定する方式（プロセスのスケジューリング）を理解しておくことは重要である。 本問は，プロセスの状態遷移，代表的なプロセスのスケジューリングであるラウンドロビン方式及び優先度順方式についての理解を主題としている。 本問では，入出力処理を含んだプロセスの状態遷移や，優先度順方式における優先度の推移やプロセスの実行順序を理解する能力を評価する。

採点講評
問2では，プロセスの状態遷移と，代表的なプロセスのスケジューリングであるラウンドロビン方式及び優先度順方式について出題した。 設問1の正答率は平均的で，おおむね理解されていた。 設問2では，aの正答率は高く，よく理解されていた。bの正答率は平均的で，おおむね理解されていた。cの正答率は低く，あまり理解されていなかった。cでは，オと誤って解答した受験者が見受けられた。プロセスAは，最初のタイムクウォンタムの経過によって実行を中断した後，キュー［優先度2］に登録されることが分かれば，正答できた。 OSにおけるプロセスのスケジューリング方式は，システムの処理能力に影響を与えるので，よく理解しておいてほしい。

問3

出題趣旨
Web上の販売サイトでBtoCサービスを提供する場合，サイトの構築に当たっては関係データベースを用いることが多く，関係データベースの設計，運用を理解することは重要である。 本問は，コンサートチケット販売サイトを題材に，関係データベースからの必要な情報の抽出と，新たなサービスの導入に伴って修正した表の利用を主題としている。 本問では，表の制約，表の結合を用いた情報の抽出，及び表の更新処理を問うことによって，関係データベースの設計，運用に関する能力を評価する。

採点講評
問3では，コンサートチケット販売サイトを題材に，関係データベースからの必要な情報の抽出と，新たなサービスの導入に伴って修正した表の利用について出題した。 設問1の正答率は低く，あまり理解されていなかった。ウやエと誤って解答した受験者が見受けられた。支払手続が行われると決済表にレコードが追加され，決済期限日までに支払手続が行われないと決済表に決済額が−1のレコードが追加されるので，決済表の決済額は−1以上の値をとる項目となる。このことが分かれば，正答できた。制約条件は表を設計する上で特に必要とされることであり，是非理解してもらいたい。 設問2の正答率は低く，あまり理解されていなかった。イと誤って解答した受験者が見受けられた。販売できない席かどうかを判定する条件，すなわち購入申込みされた席のうち決済期限日前で支払手続が行われていない，又は支払手続が行われた販売IDであるという条件文がa2に入る。支払手続が行われていない場合は，外部結合した決済表のレコードがないので決済額にNULLが設定されることが分かれば，イを選ぶことはなかった。 設問3の正答率は平均的で，おおむね理解されていた。 設問4の正答率は低く，あまり理解されていなかった。エと誤って解答した受験者が多く見受けられた。SQLでは，分岐処理をCASE文で記述することが分かれば，エを選ぶことはなかった。 関係データベースでは，外部結合やCASE文を適切に利用することによって，複雑な条件でも一つのSQL文で処理を行うことができる場合が多いので，しっかり身につけておいてほしい。

問4

出題趣旨
ネットワークの運用業務では，障害が発生したときの原因の特定や，障害発生の予防のための対策を施すことは重要である。 本問は，社内ネットワークの運用を題材に，障害の原因の切り分けと，障害発生の予防策の検討を主題としている。 本問では，障害の内容と，その後に実施した障害箇所特定のための切り分け作業の結果から，障害の原因として考えられる箇所を絞り込む能力や，障害発生の予防を目的とした機器の増設において，機器構成による効果の違いを検証する能力などを評価する。

採点講評
問4では，社内ネットワークの運用を題材に，障害の原因の切分けと，障害発生の予防策について出題した。 a～cの正答率は平均的で，おおむね理解されていた。aとbでは，エやカと誤って解答した受験者が見受けられた。障害箇所の特定のための切分け作業では，その作業の結果，何が確認できたかを正確に把握することが重要である。例えば，“PCB-1からSSHを用いてログインした開発セグメント1内のPC1-2”からスイッチB及びルータBを経由した通信ができたので，スイッチB及びルータBのLANポートには障害がないことを理解できれば，エやカを選ぶことはなかった。 dとeの正答率は高く，よく理解されていた。 ネットワークの障害時における障害箇所の切分けの考え方は重要なので，よく理解してほしい。

問5

出題趣旨
ソフトウェアの設計に際しては，処理における判定条件とそれに伴う動作を明確に示しておくことが求められる。 本問は，購買管理システムにおける購買ファイル更新可否チェック処理を題材に，チェック処理の設計とテストケースの設計を主題としている。 本問では，求められる処理の要件に基づいた流れ図を作成する能力と，出力するファイルに着目して決定表を作成する能力を評価する。

採点講評
問5では，購買管理システムにおける購買ファイル更新可否チェック処理を題材に，チェック処理の設計とテストケースの設計について出題した。 設問1の正答率は高く，よく理解されていた。 設問2の正答率は高く，よく理解されていた。cでは，エと誤って解答した受験者が見受けられた。購買ステータスが“納品済”のときは，更新対象外依頼ファイルに出力することが理解できれば，正答できた。 設問3では，dの正答率は平均的で，おおむね理解されていた。エと誤って解答した受験者が見受けられた。図2の破線で囲んだ処理が実行されるのは，購買ステータスが“購買受付”又は“見積り中”のときであることが理解できれば，正答できた。eの正答率は高く，よく理解されていた。 求められる処理の要件に基づいた設計と適切なテストケースの設計の能力は，ソフトウェアの設計をする上で前提となる能力なので，しっかり身につけておいてほしい。

問6

出題趣旨
プロジェクトを円滑に進めるためには，スケジュール作成が重要である。 本問は，りん議書電子化プロジェクトを題材に，アローダイアグラムを用いたスケジュールの作成を主題としている。 本問では，アローダイアグラムを用いて，作業全体及び作業間の依存関係を把握し，整合性のあるスケジュールを作成する能力を評価する。

採点講評
問6では，りん議書電子化プロジェクトを題材に，アローダイアグラムを用いたスケジュールの作成について出題した。 設問1では，a，b及びdの正答率は平均的で，おおむね理解されていた。cの正答率は低く，あまり理解されていなかった。アと誤って解答した受験者が多く見受けられた。作業Jを始めるためには結合点6までの全ての作業が終了している必要がある。作業Jの最早開始日は，クリティカルパス上の作業B，C及びGの所要日数の合計が7日になることが分かれば，アを選ぶことはなかった。 設問2では，eとfの正答率は平均的で，おおむね理解されていた。gの正答率は低く，あまり理解されていなかった。エと誤って解答した受験者が見受けられた。これは結合点4から結合点5のダミー作業を含んでいない総所要日数を計算した結果と思われる。 作業の依存関係を把握し，整合性のあるスケジュールを作成する能力及びクリティカルパスを認識する能力は，プロジェクトを円滑に進めるために重要なので，しっかり身につけておいてほしい。

問7

出題趣旨
企業にとって，業務の問題点を追及し，改善を図ることは重要であり，そのために業務の現状を把握し，継続的に改善していくことが求められる。 本問は，広告制作会社の制作部門における作業の流れを題材に，広告原稿や作成済広告などの受渡し及び保管時における現状の問題点の分析，さらに電子化による改善効果の算出に関する理解を主題としている。 本問では，作業の流れを正確に把握し，問題点を摘出する能力や，改善効果の算出を的確に行える能力などを評価する。

採点講評
問7では，広告制作会社の制作部門における作業の流れを題材に，広告原稿や作成済広告などの受渡し時及び保管時における現状の問題点の分析，さらに電子化による改善効果の算出について出題した。 設問1の正答率は平均的で，おおむね理解されていた。ウやエと誤って解答した受験者が見受けられた。〔制作部門での作業の流れ〕(1)④の作業完了時では，点検の結果，不備がないことが理解できれば，正答できた。 設問2の正答率は高く，よく理解されていた。 設問3の正答率は低く，あまり理解されていなかった。広告原稿を電子化する時点が異なることで，どの受渡しの作業がなくなるかが理解できれば，正答できた。b では，ウと誤って解答した受験者が見受けられた。制作部門内での広告原稿の受渡しだけがなくなることに気がつけば，正答できた。 業務の改善においては，作業の流れや作業内容を正確に把握した上で問題点を抽出し，その対策を基に継続的に改善していくことが求められる。更に，改善効果の算出を的確に行える能力が重要なので，身につけておいてほしい。

問8

出題趣旨
データ構造のうち，配列は最も基本的なものであり利用範囲が広い。 本問は，整数式を受け取って，その値を返すプログラムを題材に，配列を用いた基本的な文字処理及び数値処理を主題としている。 本問では，プログラムの処理内容の理解に加えて，プログラム中の定数のより一般的な表現への拡張，四則演算の基本的な理解と演算の実行順序が計算結果に及ぼす影響，符号付き整定数の扱いなどに関して，プログラムの読解力と論理的な思考力を評価する。

採点講評
問8では，整数式を受け取って，その値を返すプログラムを題材に，配列を用いた基本的な文字処理及び数値処理について出題した。 設問1では，aの正答率は平均的で，おおむね理解されていた。bの正答率は低く，あまり理解されていなかった。括弧内の加減算の優先順位が，括弧外の乗除算の優先順位よりも高ければよいことに着目すれば，正答できた。 設問2の正答率は低く，あまり理解されていなかった。cでは，ウと誤って解答した受験者が見受けられた。ウの場合，行⑤でipに設定する値が0のままでは，正しい処理ができない。dでは，ウと誤って解答した受験者が多く見受けられた。括弧内の減算の結果が変わることを見落としたものと思われる。 設問3では，gの正答率は平均的で，おおむね理解されていた。eとfの正答率は低く，あまり理解されていなかった。eではイと，fではウと誤って解答した受験者が多く見受けられた。図3の式について，“プログラム（解析処理の部分）”の動作を追跡すれば，正答できた。 プログラム中の定数をより一般的な表現に拡張できる能力，及びプログラムの動作を実際に追跡できる能力は重要なので，身につけておいてほしい。

問９

出題趣旨
構造体は，Ｃプログラム内で，複数のデータを関連性の観点でまとめる場合によく用いられる。 本問は，鉄道模型における列車の運行シミュレーションを題材に，構造体を用いた基本的な処理の理解を主題としている。 本問では，鉄道模型の路線を構成する区間の情報及び区間内にいる列車の情報を，構造体で表現する。これらの構造体を扱うプログラムを通して，鉄道模型とこれらのデータ構造を正しく結び付ける能力や，構造体を操作するプログラムの作成能力を評価する。また，プログラムを実行したときの状態を問うことによって，プログラムの動作を追跡する能力を評価する。

採点講評
問９では，鉄道模型における列車の運行シミュレーションを題材に，構造体を用いた基本的な処理について出題した。 設問１の正答率は低く，あまり理解されていなかった。ア，ウやエと誤って解答した受験者が見受けられたこれらの選択肢はいずれも変数blockが示す区間の次区間に関するものであり，変数 block が示す区間の信号機の表示を決定する処理の条件式とはなり得ない。列車を進行させるルールをプログラムと正しく対応付けることで，正答できた。 設問２の正答率は平均的で，おおむね理解されていた。 設問３の正答率は低く，あまり理解されていなかった。e ではウやエと，f ではオと誤って解答した受験者が見受けられた。各列車はどの区間にいるか，また各区間の信号機が赤と緑のいずれを表示しているかを，繰り返すごとに丁寧に追うことで，正答できた。 構造体を用いたデータ構造は実務においてよく用いられるので，その扱いに習熟しておいてほしい。また，データ構造や処理を仕様と正しく結び付ける能力は，プログラミングにおいて重要なので，しっかり身につけておいてほしい。

問10

出題趣旨
順ファイルに格納したトランザクションデータを，索引ファイルに格納してあるマスタデータに反映する処理は，COBOLで記述された業務プログラムで多用される。 本問は，従業員が取得した社内資格の管理を題材に，新たに取得した資格をマスタデータに反映する処理を主題としている。 本問では，ファイル入出力の基本操作に加えて，整列，添字付けなどの実装を問うことによって，COBOLプログラムの作成能力を評価する。また，追加した処理の内容を把握する能力を評価する。

採点講評
問10では，従業員が取得した社内資格の管理を題材に，新たに取得した資格をマスタデータに反映する処理について出題した。 設問1では，aとbの正答率は低く，あまり理解されていなかった。aでは，アと誤って解答した受験者が見受けられた。整列ファイルでAT ENDを検出した時点では，直前に処理した情報の書き出しが完了していないので，WRI-PROCを呼び出して処理する必要があることが理解できていれば，正答できた。bでは，イと誤って解答した受験者が見受けられた。期間中に複数の資格に合格した従業員の場合，保有状況を蓄積し，従業員番号が変化したタイミングで保有資格ファイルに書き込むことが理解できていれば，正答できた。cの正答率は平均的で，おおむね理解されていた。 設問2では，dの正答率は低く，あまり理解されていなかった。ウと誤って解答した受験者が見受けられた。従業員番号の切替えと，保有資格レコードへの反映処理のタイミングが理解できていれば，正答できた。eの正答率は平均的で，おおむね理解されていた。 設問3の正答率は平均的で，おおむね理解されていた。 複数のトランザクションデータを一つのマスタデータに反映する場合，マスタファイルへの入出力を減らすために，複数のトランザクションデータを整列して一度に処理する手法が用いられる。データ処理の流れに着目してプログラムを読み解く能力を身につけておいてほしい。

問 11

出題趣旨
文書の書式を表すひな形を用意しておき，これに任意の情報を埋め込むことによって，電子メールで送信する文書や HTML 文書などを完成させるプログラムは，テンプレートエンジンとして広く利用されている。 本問は，ひな形に置換表を適用して出力文書を得るプログラムを題材に，ひな形を解析するプログラムと，ひな形に置換表を適用するプログラムの完成を主題としている。 本問では，Java プログラム作成においてのインスタンスの生成や同じインタフェースを実装した複数のクラスの利用についての理解，メソッドに渡す引数を適切に選択する能力などを評価する。

採点講評
問 11 では，ひな形に置換表を適用して出力文書を得るプログラムを題材に，ひな形を解析するプログラムと，ひな形に置換表を適用するプログラムの完成について出題した。 設問 1 では，a，d 及び e の正答率は平均的で，おおむね理解されていた。b と c の正答率は低く，あまり理解されていなかった。b では，オやカと誤って解答した受験者が見受けられた。“<” は置換指示の開始を表す文字であることから，ここで置換指示を表すクラスである Replacer のインスタンスを生成すると勘違いしたと思われる。しかし，置換指示のキー名称は，“>” を検知するまで特定できないので，まだ Replacer のインスタンスは生成できないことに気がつけば，オやカを選ぶことはなかった。“<” を検知する直前までに buf に追加された文字をどう扱うべきかに着目できていれば，正答できた。c では，ウと誤って解答した受験者が見受けられた。“>” を検知したときに buf に存在する文字列がキー名称であることに気がつけば，正答できた。 設問 2 の正答率は低く，あまり理解されていなかった。ウと誤って解答した受験者が見受けられた。“\” に続く 1 文字を，置換指示以外の部分やキー名称の一部として扱う部分である。しかし，f の位置では変数 c の値がまだ “\” なので，変数 buf に追加するのは変数 c ではなく，次に読む 1 文字であることに気がつけば，正答できた。 Java のプログラム作成では，何の値を使ってインスタンスを生成するのかや，変数の値がどのような状態にあるかをしっかり理解することは重要なので，ぜひ身につけておいてほしい。

問12

出題趣旨

多くのOSや言語処理系では，1970年1月1日を基準日として日付や時刻の計算を行っている。

本問は，基準日から指定された日付までの日数を求めるプログラムを完成させることを主題としている。

本問では，うるう年を判定する方法，日数のデータを保持するテーブルを参照する方法に関わるアセンブラプログラムの作成能力，さらに，あふれを処理する能力などを評価する。

採点講評

問12では，基準日から指定された日付までの日数を求めるプログラムについて出題した。

設問1では，aの正答率は低く，あまり理解されていなかった。オと誤って解答した受験者が見受けられた。LAD命令とLD命令の違いを理解していれば，正答できた。bとcの正答率は平均的で，おおむね理解されていた。dの正答率は低く，あまり理解されていなかった。イと誤って解答した受験者が見受けられた。年を100で割り切れるかどうかを判定するために副プログラムDIVISIBLを呼び出し，その戻り値が0であれば，100で割り切れきれないことが分かり，うるう年であることが確定する。副プログラムLEAPYEARは，うるう年の場合，その戻り値が1でなければならないので，DIVISIBLの戻り値0を反転させ，1を戻り値として戻る。イでは，DIVISIBLの戻り値が変わらないので，正しい判定ができないことに注意が必要である。

設問2では，eの正答率は低く，あまり理解されていなかった。GR0でその年の日数を求めるので，計算過程で1年の日数を超えない値であることが理解できれば，正答できた。fの正答率は高く，よく理解されていた。

LAD命令とLD命令との違い，XOR命令によるビットの反転など，命令の機能を正しく理解して使用できるようにしてほしい。

問 13

出題趣旨
現状の分析に必要なデータを得ることや，条件を変えたときの変化をシミュレーションして業務改善に生かすなどの実用的な用途に，表計算ソフトが利用できる。 本問は，サービス窓口の待ち行列を題材に，来店状況を示すワークシート，来店状況を分析するためのワークシート及び待ち時間の短縮を検討するためのマクロの作成を主題としている。 本問では，表計算ソフトの関数を利用して，平均値や度数分布などを適切に求める能力，条件に応じてデータを加工する能力，検索によって必要なデータを求める能力などを評価する。また，処理手順の記述や配列の利用に関する基礎的な知識の理解と，設定された条件を理解してマクロを作成する能力を評価する。

採点講評
問 13 では，サービス窓口の待ち行列を題材に，来店状況を示すワークシート，来店状況を分析するためのワークシート及び待ち時間の短縮を検討するためのマクロの作成について出題した。 設問 2 では，c の正答率は平均的で，おおむね理解されていた。b と d の正答率は低く，あまり理解されていなかった。b では，イと誤って解答した受験者が見受けられた。セルの値が時間帯の境界に当たるときに，そのセルを数えるかどうかを正しく判断できれば，正答できた。d では，ウやオと誤って解答した受験者が見受けられた。“順位”という語に惑わされず，求める値の意味を正しく理解すれば，正答できた。 設問 3 の正答率は低く，あまり理解されていなかった。出題されたマクロのアルゴリズムの概要は〔マクロの説明〕として本文に記述されているので，これとプログラムを対応させて読み取ることができれば，正答できた。 複数の関数を組み合わせて必要な値を求めるような式を正しく記述するためには，日頃から式に実際の値を当てはめ，結果を確かめるなどして思考力を高めることが大切である。また，マクロの作成では，単に問題の空欄を埋める学習だけでなく，アルゴリズムをきちんと理解し，プログラム言語で記述する基本的な力を身につけてほしい。