FEPM0220D

基本情報技術者

第2回 模擬テスト

午後

解説・解答

株式会社インフォテック・サーブ

・本解説・解答は，株式会社インフォテック・サーブが著作権を保有し，著作権法により保護されています。解説・解答の全部または一部につき，無断で複写，複製あるいは転載されると著作権の侵害となりますので，ご注意ください。

・本解説・解答に記載されている会社名または製品名は，それぞれ各社の商標または登録商標です。なお，解説・解答では，® 及び ™ を明記していません。

問１　【解答】設問１　ａ－イ，　ｂ－オ，　ｃ－エ
設問２　ｄ－イ，　ｅ－エ
設問３　ｆ－ウ，　ｇ－ア

設問１

ａ：NPCからの認証要求を受けて，ユーザ認証を行う機器である。Ｆ社では，IEEE 802.1x方式による認証VLANを利用するため，この認証はIEEE 802.1xに準拠して行われる。IEEE 802.1xは有線LAN及び無線LANで用いられる認証方式の規格である。IEEE 802.1xでは，認証を受けるPCにインストールされた認証用ソフトウェア（サプリカント）が認証サーバに対して認証要求を送信し，認証が成功するとLANを利用できるようになる。このとき，認証サーバとしては，ネットワークにおける認証を一元管理する「RADIUSサーバ」が利用される。

ｂ：RADIUSサーバからの問合せに対して，ユーザ情報を返すサーバである。このユーザ情報とは，RADIUSサーバによる認証処理で使用されるユーザIDやパスワードのことなので，それらの情報を管理している「ディレクトリサーバ」が該当する。

ｃ：新社内ネットワーク（検疫用のVLAN）に接続されたNPCに対して，セキュリティ対策検査を行うのは「検疫サーバ」である。検疫サーバとは，検疫ネットワークに接続されたPCに対して，セキュリティパッチが適用されているかどうか，ワクチンソフトのパターンファイルが最新かどうか，スクリーンセーバなどの各種設定状況が適切かどうか，などのセキュリティ対策検査やウイルス対策を実施し，検疫に合格するまでPCを隔離するサーバである。

設問２

ｄ：RADIUSサーバによる認証が成功したNPCは，セキュリティ対策検査やウイルス対策を実施するために検疫サーバと同じネットワーク（VLAN）に接続する。したがって，“VLAN ID設定①”においてVLAN IDとして，検疫サーバが所属する「VLAN 10」（空欄d1）を設定する。一方，検疫に合格したNPCについては，〔新社内ネットワークの概要〕(2)に“…認証が成功して検疫に合格すると，外交員サーバと同じVLANに所属するNPCとして利用できる”とあるので，外交員サーバと同じネットワーク（VLAN）に接続する。したがって，“VLAN ID設定②”においてVLAN IDとして，外交員サーバが所属する「VLAN 30」（空欄d2）を設定する。

ｅ：図２から，新社内ネットワークに接続されたNPCに対しては，検疫サーバによるセキュリティ対策検査の他に，ウイルス対策サーバによるウイルスチェックも行われていることが分かる。つまり，ウイルスに感染しているNPCは，ウイルスチェックの結果を元にしたウイルスの駆除が行われ，安全性が確保されるまで隔離される。その結果，検疫用のVLAN（VLAN 10）以外のネットワーク（VLAN）にウイルスが拡散するのを防止することができる。したがって，検疫処理が終了するまでNPCを隔離するのは，「接続されたNPCがウイルスに感染している」ような場合でも被害を限定するためである。

設問３

ｆ：今回，NPCがウイルスに感染した原因については，“ウイルスに感染したNPCでは，ワクチンソフトのパターンファイルが最新版に更新されていなかった”という記述がある。したがって，パターンファイルを最新版にするために，「ワクチンソフトのパターンファイルを自動更新する機能を有効にする」べきである。

ｇ：ランサムウェアは，データを暗号化などで使えないように“人質”にし，身代金（ransom）を要求するマルウェアである。したがって，ランサムウェアに感染する場合に備える対策としては，「NPCに保存されているデータのバックアップを定期的に取得する」べきである。

問２　【解答】設問１　ａ－カ，　ｂ－イ
　　設問２　ｃ－ウ，　ｄ－カ，　ｅ－ウ

設問１

ａ：命令“20601000h”を解釈すると，次のようになる。なお，ｒ，ｘ，ｍの値を求めるために，１語目$(2060)_{16}$の下位２桁$(60)_{16}$を２進数に変換し，命令の後に併記する。

命令$(2060\ \ 1000)_{16}$：$(60)_{16}=(0110\ \ 0000)_2$

⇒　op＝20h，ｒ＝$(01)_2$＝1，ｘ＝$(10)_2$＝2，ｍ＝$(00)_2$，adr＝1000h

⇒　ｘが２（１～３に該当する値）なので，実効アドレスは，レジスタ番号２のレジスタの内容（0005h）を，adrに加算して求める。

実効アドレス＝adr＋[レジスタ番号２のレジスタ]＝1000h＋0005h＝「1005h」

⇒　命令コードが20hなので，実効アドレス(1005h)に格納されている内容を，r(1)で指定されたレジスタ番号１のレジスタに格納（上書き）する。

ｂ：実効アドレス（1005h）に格納されている内容が０（0000h）であれば，命令“20601000h”を実行するとレジスタ番号１のレジスタに０が格納（上書き）される。つまり，レジスタに格納された値の符号は負（最上位ビットが１）以外なのでSFが０，値は零（全てのビットが０）なのでZFが１になる。したがって，FRにはSF，ZFの順に，２進数「01」が設定される。

設問２

設問１と同様に，2010h番地から始まるプログラムの各命令を解釈してトレースする。

①　2010h～2011h番地の命令$(2170\ \ 201A)_{16}$：$(70)_{16}=(0111\ \ 0000)_2$

⇒　op＝21h，ｒ＝$(01)_2$＝1，ｘ＝$(11)_2$＝3，ｍ＝$(00)_2$，adr＝201Ah

⇒　実効アドレス＝adr＋[レジスタ番号３のレジスタ]＝201Ah＋0001h＝201Bh

⇒　実効アドレス(201Bh)に格納されている内容(0002h)を，r(1)で指定されたレジスタ番号１のレジスタの内容(0000h)に加算する。･･･ 0000h＋0002h＝0002h：FR$(00)_2$

②　2012h～2013h番地の命令$(21C0\ \ 201A)_{16}$：$(C0)_{16}=(1100\ \ 0000)_2$

⇒　op＝21h，ｒ＝$(11)_2$＝3，ｘ＝$(00)_2$＝0，ｍ＝$(00)_2$，adr＝201Ah

⇒　ｘ＝0なので，実効アドレス＝adr＝201Ah

⇒　実効アドレス(201Ah)に格納されている内容(0001h)を，r(3)で指定されたレジスタ番号３のレジスタの内容(0001h)に加算する。･･･ 0001h＋0001h＝0002h：FR$(00)_2$

③　2014h～2015h番地の命令$(2280\ \ 201A)_{16}$：$(80)_{16}=(1000\ \ 0000)_2$

⇒　op＝22h，ｒ＝$(10)_2$＝2，ｘ＝$(00)_2$＝0，ｍ＝$(00)_2$，adr＝201Ah

⇒　ｘ＝0なので，実効アドレス＝adr＝201Ah

⇒　実効アドレス(201Ah)に格納されている内容(0001h)を，r(2)で指定されたレジスタ番号２のレジスタの内容(0003h)から減算する。･･･ 0003h－0001h＝0002h：FR$(00)_2$

④　2016h～2017h番地の命令$(4004\ \ 2010)_{16}$：$(04)_{16}=(0000\ \ 0100)_2$

⇒　op＝40h，ｒ＝$(00)_2$＝0，ｘ＝$(00)_2$＝0，ｍ＝$(01)_2$，adr＝2010h

⇒　ｘ＝0なので，実効アドレス＝adr＝2010h

⇒　ｍ$(01)_2$とFR$(00)_2$の排他的論理和を求め，その結果が$(01)_2$で$(00)_2$ではないので，実効アドレス(2010h)に分岐する。

⑤　2010h～2011h番地の命令$(2170\ \ 201A)_{16}$：$(70)_{16}=(0111\ \ 0000)_2$

⇒　op＝21h，ｒ＝$(01)_2$＝1，ｘ＝$(11)_2$＝3，ｍ＝$(00)_2$，adr＝201Ah

⇒　実効アドレス＝adr＋[レジスタ番号３のレジスタ]＝201Ah＋0002h＝201Ch

⇒　実効アドレス(201Ch)に格納されている内容(0003h)を，r(1)で指定されたレジスタ番号１のレジスタの内容(0002h)に加算する。･･･ 0002h＋0003h＝0005h：FR$(00)_2$

⑥ 2012h～2013h番地の命令(21C0　201A)$_{16}$：(C0)$_{16}$＝(1100 0000)$_2$

⇒ op＝21h，r＝(11)$_2$＝3，x＝(00)$_2$＝0，m＝(00)$_2$，adr＝201Ah

⇒ x＝0なので，実効アドレス＝adr＝201Ah

⇒ 実効アドレス(201Ah)に格納されている内容(0001h)を，r(3)で指定されたレジスタ番号３のレジスタの内容(0002h)に加算する。･･･ 0002h＋0001h＝0003h：FR(00)$_2$

⑦ 2014h～2015h番地の命令(2280　201A)$_{16}$：(80)$_{16}$＝(1000 0000)$_2$

⇒ op＝22h，r＝(10)$_2$＝2，x＝(00)$_2$＝0，m＝(00)$_2$，adr＝201Ah

⇒ x＝0なので，実効アドレス＝adr＝201Ah

⇒ 実効アドレス(201Ah)に格納されている内容(0001h)を，r(2)で指定されたレジスタ番号２のレジスタの内容(0002h)から減算する。･･･ 0002h－0001h＝0001h：FR(00)$_2$

⑧ 2016h～2017h番地の命令(4004　2010)$_{16}$：(04)$_{16}$＝(0000 0100)$_2$

⇒ op＝40h，r＝(00)$_2$＝0，x＝(00)$_2$＝0，m＝(01)$_2$，adr＝2010h

⇒ x＝0なので，実効アドレス＝adr＝2010h

⇒ m(01)$_2$とFR(00)$_2$の排他的論理和を求め，その結果が(01)$_2$で(00)$_2$ではないので，実効アドレス(2010h)に分岐する。

⑨ 2010h～2011h番地の命令(2170　201A)$_{16}$：(70)$_{16}$＝(0111 0000)$_2$

⇒ op＝21h，r＝(01)$_2$＝1，x＝(11)$_2$＝3，m＝(00)$_2$，adr＝201Ah

⇒ 実効アドレス＝adr＋[レジスタ番号３のレジスタ]＝201Ah＋0003h＝201Dh

⇒ 実効アドレス(201Dh)に格納されている内容(0004h)を，r(1)で指定されたレジスタ番号１のレジスタの内容(0005h)に加算する。･･･ 0005h＋0004h＝0009h：FR(00)$_2$

⑩ 2012h～2013h番地の命令(21C0　201A)$_{16}$：(C0)$_{16}$＝(1100 0000)$_2$

⇒ op＝21h，r＝(11)$_2$＝3，x＝(00)$_2$＝0，m＝(00)$_2$，adr＝201Ah

⇒ x＝0なので，実効アドレス＝adr＝201Ah

⇒ 実効アドレス(201Ah)に格納されている内容(0001h)を，r(3)で指定されたレジスタ番号３のレジスタの内容(0003h)に加算する。･･･ 0003h＋0001h＝0004h：FR(00)$_2$

⑪ 2014h～2015h番地の命令(2280　201A)$_{16}$：(80)$_{16}$＝(1000 0000)$_2$

⇒ op＝22h，r＝(10)$_2$＝2，x＝(00)$_2$＝0，m＝(00)$_2$，adr＝201Ah

⇒ x＝0なので，実効アドレス＝adr＝201Ah

⇒ 実効アドレス(201Ah)に格納されている内容(0001h)を，r(2)で指定されたレジスタ番号２のレジスタの内容(0001h)から減算する。･･･ 0001h－0001h＝0000h：FR(01)$_2$

⑫ 2016h～2017h番地の命令(4004　2010)$_{16}$：(04)$_{16}$＝(0000 0100)$_2$

⇒ op＝40h，r＝(00)$_2$＝0，x＝(00)$_2$＝0，m＝(01)$_2$，adr＝2010h

⇒ x＝0なので，実効アドレス＝adr＝2010h

⇒ m(01)$_2$とFR(01)$_2$の排他的論理和を求め，その結果が(00)$_2$なので，何もしない。

⑬ 2018h～2019h番地の命令(FF00　0000)$_{16}$：(00)$_{16}$＝(0000 0000)$_2$

⇒ op＝FFh，r＝(00)$_2$＝0，x＝(00)$_2$＝0，m＝(00)$_2$，adr＝0000h

⇒ プログラムの実行を終了する。

ｃ：太線で囲まれた命令“2014h～2015h番地の命令(2280　201A)$_{16}$”は，③，⑦，⑪の「３」回実行される。

ｄ：プログラムの実行終了後のレジスタ番号１のレジスタの内容は，⑨で命令コード(21h)の動作によって設定された「0009h」になる。

ｅ：プログラムの実行終了後のレジスタ番号３のレジスタの内容は，⑩で命令コード(21h)の動作によって設定された「0004h」になる。

問３　【解答】設問１　ａ－イ，　ｂ－イ
　　設問２　ｃ－イ，　ｄ－ウ
　　設問３　ｅ－エ

設問１

ａ：扶養手当を求める部分である。扶養手当の内訳は，“扶養者数が２人までは１人につき１万円が，３人目以降は１人につき５千円が支給される”とあるので，扶養者数によって分けて求める必要がある。したがって，この場合の扶養手当は検索CASE式を使用し，扶養者数が２人以下なら“扶養者数×10,000”で，それ以外（３人以上）なら“３人目以降の人数（扶養者数－2）×5,000＋２人分（20,000)”で求めるようにする。

```
「CASE
    WHEN X.扶養者数 <= 2 THEN X.扶養者数 * 10000
    ELSE (X.扶養者数 - 2) * 5000 + 20000
  END」
```

ｂ：FROM句の指定で，社員表と職位表の直積をとる（公差結合をしている）ので，二つの表の職位コードが等しいレコードを抽出する結合条件が必要になる。また，管理職の社員を抽出するため，職位名が“一般”と等しくないレコードを抽出する条件も必要となる。したがって，二つの条件をANDで結んだ「**X.職位コード = Y.職位コード AND Y.職位名 <> '一般'**」となる。

設問２

ｃ：残業手当を求める部分である。残業手当は，“残業賃金（基本給の150分の１に当たる時間賃金)×該当月の残業時間”で求めるので，この算術式を次のように展開する。

残業手当＝残業賃金×残業時間
＝(基本給÷150)×残業時間
＝基本給×残業時間÷150

したがって，「**X.基本給 * Y.残業時間 / 150**」となる。

ｄ：FROM句の指定で，社員表と残業表の直積をとる（公差結合をしている）ので，二つの表の社員番号が等しいレコードを抽出する結合条件が必要になる。また，残業手当が支給されるのは一般職の社員だけなので，職位コードが“S4”と等しいレコードを抽出する条件と，2020年９月分の残業手当（残業時間）を求めるため，残業年月が“2020-09”と等しいレコードを残業表から抽出する条件も必要となる。ただし，残業表に記録されているのは職位名“一般”の社員のレコードだけなので，条件“職位コード = 'S4'”は必要ない。したがって，残りの二つの条件をANDで結んだ「**X.社員番号 = Y.社員番号 AND Y.残業年月 = '2020-09'**」となる。

設問３

ｅ：職位名が“部長”以外の社員を副照会文を用いて抽出するには，二つの方法が考えられる。

①：職位名“部長”の職位コードと等しくない職位コードのレコードを抽出
②：職位名“部長”以外の職位コードのいずれかと等しい職位コードのレコードを抽出

ただし，②の方法では複数の職位コード（S2～S4）が求められるため，ウのように“=”を使用して判定するとエラーになる（IN述語を使用しなければいけない)。したがって，①の方法を使用して「**職位コード <> (SELECT 職位コード FROM 職位表 WHERE 職位名 = '部長')**」と判定する。なお，解答群のアやイのようにEXISTS述語を使用する場合は，相関副照会の形にしなければ正しく判定できない。

問４　【解答】設問１　ａ－エ，　ｂ－ウ
設問２　ｃ－イ
設問３　ｄ－ウ，　ｅ－ア，　ｆ－ア

設問１

ａ：整列処理１の整列の目的は，販売金額計算で商品番号をキーとして，ファイル１と商品マスタを突き合わせることである。商品マスタは商品番号の昇順に並んでいる順ファイルなので，整列処理１では，整列キーを商品マスタと同じ「(商品番号：昇順)」とする。

ｂ：整列処理２の整列の目的は二つある。一つ目の目的は，リスト作成でカード番号をキーにして顧客マスタから顧客名を読み込むことである。顧客マスタはカード番号の昇順に並んでいる順ファイルなので，整列の第１キーはカード番号の昇順になる。二つ目の目的は，請求金額リストを順序どおりに印字することである。請求金額リストは顧客ごとに印字されるので，整列の第１キーはカード番号となる。また，同じ顧客（カード番号）の場合，請求明細の印字順序は販売日の昇順とし，販売日が同じ場合は商品番号の昇順とする。したがって，整列処理２の整列キーは「(カード番号：昇順)，(販売日：昇順)，(商品番号：昇順)」となる。

設問２

ｃ：販売金額計算では，ファイル１と商品マスタを突き合わせ，商品マスタから必要な情報を読み込むと同時に販売金額（＝販売数量×商品単価）を計算する。販売金額は請求金額リストに印字されるので，ファイル２に保存しておかなければならない。また，販売金額を求めるための商品単価も請求金額リストに印字されるので，ファイル２に保存しておかなければならない。したがって，ファイル２のレコードには，「商品単価，販売金額」が必要となる。なお，商品名は請求金額リストに印字されないので必要ない。また，顧客名はリスト作成で顧客マスタから読み込むので，必要ない（商品マスタからは取得できない)。

設問３

ｄ：空欄ｄの処理後に，“残金 － 金額(I)”で残金を求めているので，空欄ｄは毎月の請求金額（金額(I)）を求める処理である。毎月の請求金額は，“(月初残金 ＋ 利息) ÷ 残りの決済回数”で求める。設問文には，“１か月後（翌月）の請求金額を金額(0)に，２か月後の請求金額を金額(1)に，…，６か月後の請求金額を金額(5)に，それぞれ求める”とあるので，処理時点での残りの決済回数は変数Ｉを用いて，“決済回数 － I”で求められる。したがって，毎月の請求金額は「金額(I) ← 残金 ÷ (決済回数 － I)」で求める。

ｅ：この判定が行われるのは，直前の繰返し処理を“I ＝ 決済回数 － 1”となって抜けた後である。つまり，決済最終月の直前の月までの請求金額の計算が終わった後の処理なので，決済最終月の請求金額を求めればよい。繰返し処理は，月初残金（利息も含む）から毎月の請求金額（金額(I)）を差し引いた値を残金に代入して抜けているので，この時点の残金には次月の月初残金（利息を含まない）が求められている。そこで，空欄ｅの条件が成立した場合の処理を見てみると，金額(I)に残金を代入している。このように，利息を含まない月初残金を決済最終月の請求金額とするのは，利息が発生しない場合（決済回数が１回の場合）である。この場合，繰返し処理は一度も実行されず，変数Ｉの値が初期値（０）のまま繰返しを抜けることになるので，「I ＝ 0」と判定する。

ｆ：決済回数が２回以上のときに，決済最終月の請求金額を求める処理である。この場合，繰返しを抜けた時点の“残金（利息を含まない月初残金）＋ 利息”を請求金額（金額(I)）とすればよいので，「金額(I) ← 残金 × 1.03」とする。

問５　【解答】設問１　ａ－イ，　ｂ－エ，　ｃ－イ
　　設問２　ｄ－イ，　ｅ－ク，　ｆ－オ

設問１

ａ：総資本経常利益率は，経常利益と総資本を用いて求める。

経常利益＝売上高－売上原価－販管費＋営業外収益－営業外費用
＝3,600百万円－2,160百万円－1,278百万円＋24百万円－6百万円
＝180百万円

総資本＝負債合計（他人資本)＋純資産合計（自己資本）
＝1,040百万円＋160百万円
＝1,200百万円

総資本経常利益率＝経常利益÷総資本×100
＝180百万円÷1,200百万円×100
＝「15.0」（％）

ｂ：固定資産回転率は，売上高と固定資産を用いて求める。

固定資産回転率＝売上高÷固定資産
＝3,600百万円÷800百万円
＝「4.5」（回）

ｃ：固定比率は，固定資産と純資産（自己資本）を用いて求める。

固定比率＝固定資産÷純資産×100
＝800百万円÷160百万円×100
＝「500.0」（％）

設問２

ｄ：総資本が20％増加した場合，売上高が同じなら“売上高÷総資本”で求める総資本回転率は低下する。しかし，売上高は，前年度2,800百万円から今年度3,600百万円と大きく増加（＋800百万円）している。したがって，総資本回転率が前年度の2.8回から3.0回と高くなったのは，「売上高の増加」が貢献した結果と考えられる。

ｅ：売上高総利益率（＝売上総利益÷売上高）と売上高経常利益率（＝経常利益÷売上高）の算出式の分母は，どちらも売上高である。つまり，売上高総利益率が低下して売上高経常利益率が向上しているのは，売上総利益（＝売上高－売上原価）と経常利益（＝売上総利益－販管費＋営業外収益－営業外費用）の差によるものと考えられる。また，経常利益の式中にある販管費は売上を得るための営業費用や会社全体を運営管理するための人件費などの費用であり，通常は売上高が増加すると同じように増加する。ところが，今年度は売上高の増加率（約29％増）と比べ，販管費の増加率（約11％増）は非常に少ない。つまり，「販管費の増加」を抑制できたことで，売上高経常利益率が前年度よりも高くなったと考えられる。

ｆ：流動比率（＝流動資産÷流動負債）は，短期間で返済しなければならない債務（流動負債）に対して，短期間に現金化できる資産（流動資産）がどれだけあるかという短期支払能力を示す指標である。本来，流動比率が低い場合，資金繰りに困る状態に陥りやすい。しかし，Ｋ社の場合は，流動資産に占める当座資産（現金及び預金，短期間で現金化できる売掛金）の割合が90％（＝360百万円÷400百万円×100）と高く，当座比率が流動比率に占める割合も90％（＝45.0％÷50.0％×100）と高い。つまり，「相対的に当座比率が高い」ので，資金不足も起こりにくく，短期支払能力も指標値が示すほど低くはないといえる。ただし，一般的な当座比率の目安は100％以上が望ましいとされるので，流動比率と同様，改善の余地がある。

問６　【解答】設問１　ａ－エ，　ｂ－ウ，　ｃ－オ
設問２　ｄ－ウ，　ｅ－オ
設問３　ｆ－ウ，　ｇ－イ

設問１

ａ：〔プログラム１の説明〕(3)③の“登録した英単語と，その親の英単語について，辞書順の大小関係を比較し，親の英単語の方が大きければ交換する”処理である。選択処理前に，変数Cidxには，初期値としてヒープの要素数（HeapIdx[0]）を代入しているが，これは登録した英単語が記録されたWordTable[]の要素番号を格納した，HeapIdx[]の最後尾の要素の要素番号である。つまり，HeapIdx[Cidx]に格納された要素番号が示す英単語の，親に当たる英単語の要素番号が格納された，HeapIdx[]の要素番号を変数Pidxに求める。ヒープの親子関係については，〔英単語の管理方法の説明〕(3)③に“HeapIdx[I]に格納された要素番号が示す英単語を親とする，左の子の英単語が記録されたWordTable[]の要素番号をHeapIdx[2×I]に格納し，右の子の英単語が記録されたWordTable[]の要素番号をHeapIdx[2×I＋1]に格納する”と記されている。つまり，HeapIdx[Cidx]に格納された要素番号が示す英単語を，左の子又は右の子とする親の場合，親の英単語が記録されたWordTable[]の要素番号はHeapIdx[Cidx÷2]に格納されている。したがって，Pidxは“Pidx ← 「Cidx ÷ 2」”で求める。なお，整数同士の除算では整数の商を結果として返すため，右の子の場合でも(2×I＋1)÷2＝Iとなるので，解答群のイのように１を減算する必要はない。

ｂ：〔プログラム１の説明〕(5)に“整数型関数CompareWordは，引数としてWordTable[]の要素番号X，Yを受け取り”とあるので，渡すのはHeapIdx[Cidx]とHeapIdx[Pidx]である。また，整数型関数CompareWordの返却値が０よりも大きい場合に，HeapIdx[Cidx]とHeapIdx[Pidx]を交換している。この交換は，子の英単語よりも親の英単語の方が大きい場合の処理である。言い換えれば，子の英単語（HeapIdx[Cidx]に格納された要素番号が示す英単語）よりも親の英単語（HeapIdx[Pidx]に格納された要素番号が示す英単語）が大きいときに，０よりも大きい値が返ってくるように引数を渡す。整数型関数CompareWordは，英単語１（WordTable[X]）が英単語２（WordTable[Y]）よりも辞書順で大きければ１を，そうでなければ(－1)を返す。したがって，第１引数（X）には親の英単語が記録されたWordTable[]の要素番号「HeapIdx[Pidx]」（空欄b1）を指定し，第２引数（Y）には子の英単語が記録されたWordTable[]の要素番号「HeapIdx[Cidx]」（空欄b2）を指定する。

ｃ：〔プログラム１の説明〕(4)④の“ヒープの根に移動した英単語と，その子（子が二つある場合は辞書順で小さい方の子）の英単語について，辞書順の大小関係を比較し，子の英単語の方が小さければ交換する”処理である。繰返し処理前に，変数Pidxに初期値として代入している１は，ヒープの根に移動した英単語の，WordTable[]の要素番号を格納したHeapIdx[]の要素番号である。つまり，HeapIdx[Pidx]に格納された要素番号が示す英単語の，子に当たる英単語の添字が格納された，HeapIdx[]の要素番号を変数Cidxに求める。ここで，空欄ｃの後，条件式“Cidx ＜ HeapIdx[0] and CompareWord(HeapIdx[Cidx], HeapIdx[Cidx＋1]) ＞ 0”が成立するとき，変数Cidxに１を加算する。これは，左の子の英単語が記録されたWordTable[]の要素番号を格納したHeapIdx[]の要素番号（2×I）を，右の子の英単語が記録されたWordTable[]の要素番号を格納したHeapIdx[]の添字（2×I＋1）に変更する処理である。したがって，変数Cidxには，左の子の英単語が記録されたWordTable[]の要素番号を格納したHeapIdx[]の添字（2×I）を，“Cidx ← 「Pidx × 2」”で求める。

設問２

d：テストプログラムを行番号7まで実行すると，HeapIdx[]の内容は次のようになる。

① ・SetHeap("ccu")　：英単語“ccu”をヒープに登録する。

ヒープ

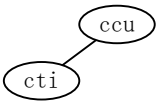
	0	1	2	3	4
WordTable[]	"ccu"				

Wnum	1

	0	1	2	3	4
HeapIdx[]	1	0			

② ・SetHeap("cti")　：英単語“cti”をヒープに登録する。

ヒープ



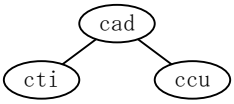
	0	1	2	3	4
WordTable[]	"ccu"	"cti"			

Wnum	2

	0	1	2	3	4
HeapIdx[]	2	0	1		

③ ・SetHeap("cad")　：英単語“cad”をヒープに登録する。

ヒープ



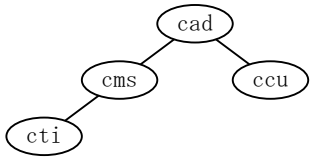
	0	1	2	3	4
WordTable[]	"ccu"	"cti"	"cad"		

Wnum	3

	0	1	2	3	4
HeapIdx[]	3	2	1	0	

④ ・SetHeap("cms")　：英単語“cms”をヒープに登録する。

ヒープ



	0	1	2	3	4
WordTable[]	"ccu"	"cti"	"cad"	"cms"	

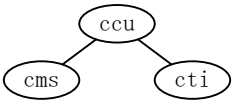
Wnum	4

	0	1	2	3	4
HeapIdx[]	4	2	3	0	1

e：行番号8以降を続けて実行すると，ヒープは次のように再編成される。

⑤ ・dummy ← GetHeap()　：ヒープの根の英単語“cad”を取り出す。

ヒープ

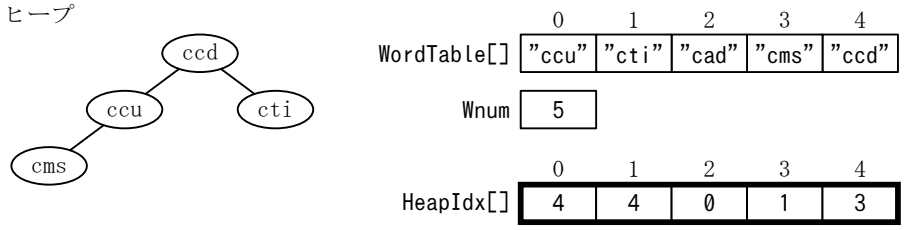


	0	1	2	3	4
WordTable[]	"ccu"	"cti"	"cad"	"cms"	

Wnum	4

	0	1	2	3	4
HeapIdx[]	3	0	3	1	1

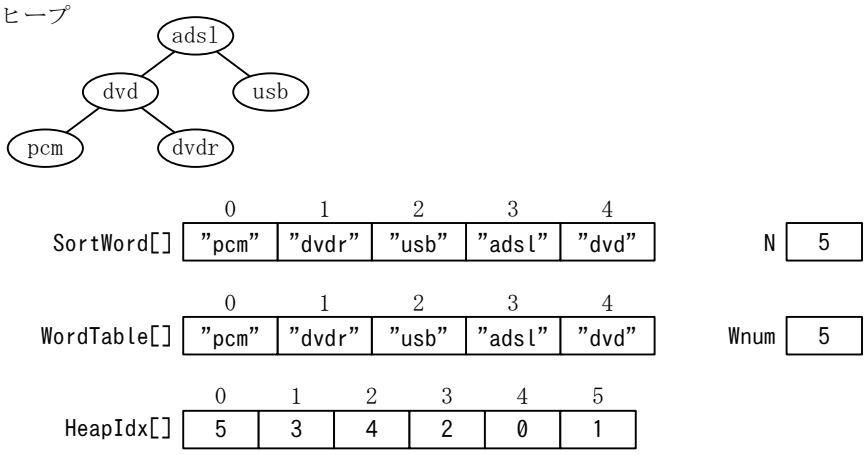
⑥ ・SetHeap(”ccd”) ：英単語“ccd”をヒープに登録する。



設問３

ｆ：ヒープを利用した整列方法をヒープソートという。ヒープソートは，ヒープの根の値が最大値（又は最小値）であることを利用した整列である。具体的には，整列対象の値を全てヒープに登録した後，ヒープの根から取り出した値を順番に並べて整列する。したがって，前半の繰返し処理では整列対象の英単語（SortWord[0]，SortWord[1]，…，SortWord[N－1]）を順番にヒープに登録していくために，「SetHeap(SortWord[Idx1])」と英単語を順番に渡しながら関数SetHeapを実行する。なお，解答群のアやエのように，HeapIdx[]やWordTable[]に値を直接代入してもヒープは構築されないため，整列することはできない。

ｇ：関数SetHeapで図６の全ての英単語をヒープに登録した結果は，次のようになる。

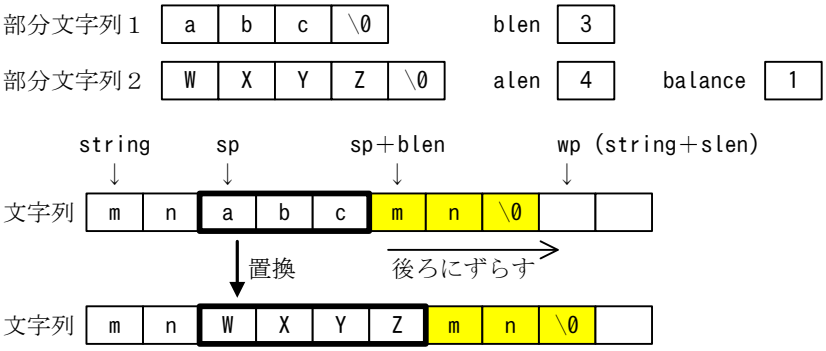


後半の繰返し処理では，ヒープの根から取り出した英単語をSortWord[0]，SortWord[1]，…，SortWord[N－1]の順番に格納する。このとき，ヒープの根から英単語を取り出すのに，整数型関数GetHeapを使用している。ただし，整数型関数GetHeapが返すのは，ヒープの根の英単語が記録されたWordTable[]の要素番号である（上図の場合，ヒープの根の英単語“adsl”が記録されたWordTable[]の要素番号３が返される）。したがって，直前の処理で整数型関数GetHeapの返却値を代入した変数Idx2 を使用して，WordTable[]に記録されたヒープの根の英単語を「SortWord[Idx1] ← WordTable[Idx2]」と代入する。なお，前半の繰返し処理が終わった時点（上図）でのWordTable[]とSortWord[]の状態は同じであるが，ヒープの根から取り出した英単語をSortWord[0]，SortWord[1]，…と順番に上書きしていくことになるため，解答群アの“SortWord[Idx1] ← SortWord[Idx2]”では正しい結果が得られない。

問７　【解答】設問１　ａ－オ，　ｂ－ア，　ｃ－ア
　　設問２　ｄ－ア，　ｅ－ウ

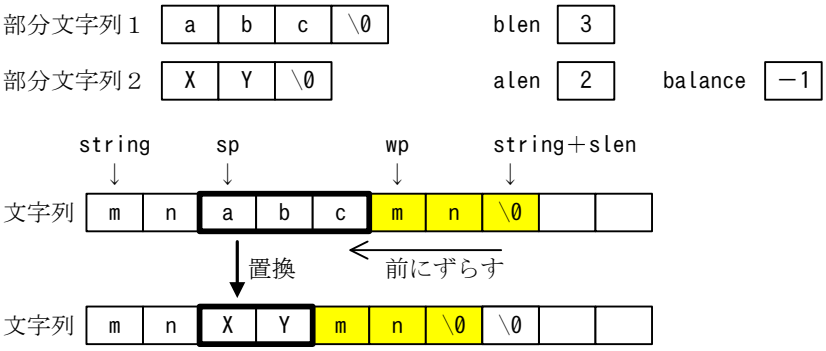
設問１

ａ：探索中の文字列に部分文字列１が見つかり，部分文字列１の文字数が部分文字列２の文字数以下の場合に，後続の文字列を後ろにずらす処理である。



したがって，*wpにbalance文字（この例でいえば１文字）前の文字を転記して文字を後ろにずらす処理を，sp＋blenの文字まで繰り返せばよいので「*wp = *(wp - balance)」となる。

ｂ：探索中の文字列に部分文字列１が見つかり，部分文字列１の文字数が部分文字列２の文字数よりも大きい場合に，後続の文字列を前にずらす処理である。



したがって，*wpの文字を１文字前に転記して文字を前にずらす処理を，string＋slenの文字まで繰り返せばよい。ただし，balanceには－1が求められているので，１文字前に転記する命令は「*(wp + balance) = *wp」となる。

ｃ：ここまで説明してきたように，部分文字列２は*spから始まる領域に複写すればよい。また，全ての文字（alen文字）を複写するので，「strncpy(sp, after, alen)」となる。

設問２

ｄ：現在のプログラムでは，置換した部分文字列２を飛ばして次の文字以降を探索する。spの位置を“ ”で示しながらトレースすると，“abccabcc”（置換）→“cabcabcc”→“cabcabcc”（置換）→“cabccabc”→「cabccabc」となる。

ｅ：**命令α**を削除すると，置換した部分文字列２を飛ばさずに（置換した文字を含めて），spの位置を１文字ずつ後ろにずらして探索する。空欄ｄと同様にspの位置を“ ”で示しながらトレースすると，“abccabcc”（置換）→“cabcabcc”（置換）→“ccababcc”→“ccababcc”→“ccababcc”（置換）→“ccabcabc”（置換）→“ccabccab”→“ccabccab”→「ccabccab」となる。

問８　【解答】設問１　ａ－イ，　ｂ－ク，　ｃ－ウ
　　設問２　ｄ－エ，　ｅ－イ，　ｆ－イ，　ｇ－イ

設問１

ａ：図１の実行結果から，一般入試，推薦入試の受験者の“合　格”又は“不合格”を表示する処理であることが分かる。この場合，boolean型のexamResultがtrueのときに合格・不合格のどちらなのか判断する必要がある。examResultにはメソッドgetResultで取得した，一般入試の受験者の試験結果（scoreResult）又は推薦入試の受験者の面接結果（interviewResult）が，メソッドsetExamResultによって設定される。この一般入試の試験結果及び推薦入試の面接結果は，メソッドtoResultの中でtrueのときに文字列“(合)”に置き換えられている。つまり，trueが合格を表しているので，入試結果（examResult）がtrueのときに“合　格”，falseのときに“不合格”を表示する「(examResult ? ”合　格” : ”不合格”)」となる。

ｂ：メソッドsetResultを用いて，試験の得点による合否結果を設定する処理である。一般入試では得点が60点以上を合格とするので，試験の得点（score）が60点以上のときにtrue（合格）を設定するように「score >= 60」と判定する。なお，“|| score <= 100”を付けると，全ての受験者が100点以下の条件を満たして合格となるので誤りである。

ｃ：メソッドsetResultを用いて，面接の評価による合否結果を設定する処理である。推薦入試では評価がＢ以上（Ａ又はＢ）を合格とするので，面接の評価（interview）がＡ又はＢのときにtrue（合格）を設定するように「interview == 'A' || interview == 'B'」と判定する。なお，評価Ｂ以上を，“interview >= 'B'”と判定すると，文字コードでは'A'は'B'よりも小さい値となるので誤りである。

設問２

ｄ：クラスSpecialExamineeは特待入試の受験者のクラスである。特待入試の受験者は試験と面接を受けるが，クラスSpecialExamineeは属性として試験の得点（score）と得点による合否結果（scoreResult）しかもっていない。したがって，面接の評価（interview）と面接の評価による合否結果（interviewResult）を，スーパクラス「RecommendExaminee」から継承する。

ｅ：クラスSpecialExamineeのコンストラクタでは，スーパクラスRecommendExamineeのコンストラクタを用いて，受験番号と面接の評価をもつインスタンスを生成している。このとき，スーパクラスRecommendExamineeのコンストラクタで，面接の評価による合否結果も設定されるが，この合否判定では推薦入試の基準（評価がＢ以上）が使用されている。そのため，特待入試の基準（評価がＡ）で再判定し，面接の評価による合否結果（interviewResult）を更新しなければいけない。したがって，面接結果を設定するメソッドsetResultを用いて，面接の評価がＡであるときにtrue（合格）を設定するように「setResult(interview == 'A')」とする。

ｆ：特待入試の受験結果を返す処理である。特待入試の場合は，“試験の得点が90点以上 かつ 面接の評価がＡ”の受験者を合格とする。つまり，試験の得点による合否結果（scoreResult）と面接の評価による合否結果（interviewResult）の両方にtrueが設定されているときに，特待入試の受験結果としてtrueを返せばよい。ただし，スーパクラスRecommendExamineeでは面接の評価による合否結果（interviewResult）をprivateで指定しているため，スーパクラスのメソッドgetResultを用いて，「scoreResult && super.getResult()」とする。

ｇ：特待入試の受験結果の文字列表現を返す処理である。図２の実行結果から，特待入試の受験結果の文字列表現は，“試験結果／面接結果”としていることが分かる。したがって，スーパクラスRecommendExamineeのメソッドtoResultで面接結果の文字列表現を取得し，“／”の後に「str.append(”／” + super.toResult())」と追加する。

問９　【解答】設問１　ａ－ウ，　ｂ－ア，　ｃ－ウ，　ｄ－ウ
　　設問２　ｅ－イ，　ｆ－ウ，　ｇ－オ

設問１

ａ：文字列realStrから取り出した文字cを，数値に変換する処理である。これは，数字'0'～'9'を順番に並べた文字列cNumから文字cを検索し，最初の出現位置を求めればよい。したがって，メソッドfindを使用して「cNum.find(c)」で求める。なお，メソッドindexでも最初の出現位置を求めることはできるが，文字列cNumに含まれない小数点'.'を検索したときにエラーとなるため，正しく処理することができない（メソッドfindの場合は(－1)を返すので，次のif文の条件“val >= 0”で小数点の処理を分けることができる)。

ｂ：変数valには，分数に変換する実数の各桁の数値が順番に取り出される（例１の“12.3456”の場合，１，２，３，４，５，６と取り出される)。そこで，valを使用して，約分する前の分子（例１の場合，123456）を求めるので，それまでに求めた分子（numerator）を10倍してvalの値を加算する「numerator * 10 + val」となる。

val	numerator
	0
1	0 × 10 ＋ 1 ＝ 1
2	1 × 10 ＋ 2 ＝ 12
3	12 × 10 ＋ 3 ＝ 123
4	123 × 10 ＋ 4 ＝ 1234
5	1234 × 10 ＋ 5 ＝ 12345
6	12345 × 10 ＋ 6 ＝ 123456

ｃ：関数realToFractionでは，分数に変換する実数の小数点以下の桁数（places）を用いて，分母を10^{places}で求めている。したがって，小数点が出現した（valが(－1)となった）位置から小数点以下の桁数を数えるため，「places = 0」とplacesを０で初期化する。

ｄ：条件が成立している間，分子（numerator）と分母（denominator）を変数factorで除算する処理を繰り返している。これは，factorが公約数（分子と分母に共通する約数）の場合の処理であるため，「numerator % factor == 0 and denominator % factor == 0」と，分子と分母がfactorで割り切れるか判断する。

設問２

ｅ：文字列αの場合は，分子（numerator）を求め終わった時点で小数点以下の桁数（places）が０にされ，分母は１となるので，“123456. ＝ 123456 ／ 1”という正しい結果が得られる。一方，文字列βの場合は，最初に小数点以下の桁数（places）が０にされ，残りの桁は全て小数点以下の桁として扱われるため，分母は1000000となるので，“.123456 ＝ 1929 ／ 15625”という正しい結果が得られる。したがって，「①の文字列αと②の文字列βの両方で正しい結果が得られた」ことになる。

ｆ：文字列に小数点が含まれない場合，小数点以下の桁数（places）が繰返し中で０にされなくなる。つまり，分子（numerator）を求め終わった時点でplacesには文字数６が求められ，全て小数点以下の桁として扱われるので，「文字列“.123456”を引数として実行したときと同じ結果が得られた」ことになる。

ｇ：文字列に小数点が二つある場合，小数点が出現するごとに小数点以下の桁数（places）は０にされる。つまり，最後に出現した小数点だけが有効となるので，「文字列“1234.56”を引数として実行したときと同じ結果が得られた」ことになる。

問10　【解答】設問１　ａ－ウ，　ｂ－ア
　　設問２　ｃ－ア，　ｄ－エ
　　設問３　ｅ－ウ，　ｆ－ク，　ｇ－エ

設問１

ａ：この処理の直後で抽出ビット数ｎ（GR2）と16を比較し，“n＝16”ならFIN段落にジャンプして処理を終了する。この場合，抽出結果のビットパターンとしてGR1の上位16ビット，すなわちGR1のビット列をそのままGR0に設定して呼出し元に戻る。したがって，「LD　GR0,GR1」とGR1のビット列をGR0に代入する。

ｂ：マスクビットとして，上位ｎビットが'1'のビット列をGR0に作成する処理である。直前の処理でGR0に#8000（２進数：1000 0000 0000 0000）を代入してあるので，算術右シフトを使用して'1'のビットをｎビットまで増やしていく。ただし，既に'1'のビットが１個あるので上位ｎビットを１にするには(n－1)ビットシフトする。したがって，抽出ビット数ｎ（GR2）を使用して，「SRA　GR0,-1,GR2」と(n－1)ビット算術右シフトする。

設問２

ｃ：この処理は，IPアドレスの上位16ビットを処理するときにだけ発生する。サブネットアドレスとして比較するビット数（GR2）が16以上の場合，上位16ビットは全てサブネットアドレスに含まれるので，比較するビット数（GR2）は16ビットとなる。したがって，比較するビット数（GR2）に，「LD　GR2,=16」で16を代入する。

ｄ：IPアドレスＡ，Ｂのビット列（上位16ビット又は下位16ビット）から，上位GR2ビットのビットパターンを副プログラムEXTRACTで抽出した後の処理である（IPアドレスＡのビット列からの抽出結果はGR5に，IPアドレスＢのビット列からの抽出結果はGR0に求められている）。二つのIPアドレスが同一ネットワークに属しているかは，抽出したビットパターン（サブネットアドレス）が一致しているかどうかで判断する。二つのビットパターン（ビット列）が同じ場合は，排他的論理和演算（XOR）の結果が０になるので，その際に返却値（GR0）を０にするために「XOR　GR0,GR5」とする。

設問３

ｅ：改良前の副プログラムIPCHECKでは，次のように副プログラムEXTRACTを呼び出す。

1≦s≦16の場合：IPアドレスＡ，IPアドレスＢの上位16ビットから，それぞれ比較するビットパターンを抽出するために２回

17≦s≦31の場合：IPアドレスＡ，IPアドレスＢの上位16ビット及び下位16ビットから，それぞれ比較するビットパターンを抽出するために４回

したがって，副プログラムEXTRACTを最大「４」回呼び出すことになる。

ｆ，ｇ：副プログラムEXTRACTによるビットパターンの抽出では，サブネットマスクとの論理積演算（AND）を使用している。また，改良前の副プログラムIPCHECKでは，副プログラムEXTRACTにより抽出したビットパターンが一致するかを，排他的論理和演算（XOR）で判定している。つまり，この関係は論理式で“(A AND S) XOR (B AND S)”と表せる。この論理式は，分配則から“(A XOR B) AND S”と変形できるので，IPアドレスＡ，Ｂのビット列の排他的論理和演算（「XOR　GR1,2,GR3」）（空欄ｆ）を先に行えば，副プログラムEXTRACT（論理積演算）の呼出し回数を１回に減らせる。また，この排他的論理和演算の結果が０の場合，二つのビット列は一致しているので，抽出ビットパターンは必ず一致する。この場合，次の語の処理に進めるので，空欄ｇは「JZE　NEXT」となる。これにより，副プログラムEXTRACTを１回も呼び出さずに（呼出し回数：０回），処理を終了することもできるようになる。

問11 【解答】設問１　ａ－エ，　ｂ－ウ，　ｃ－イ
設問２　ｄ－カ，　ｅ－エ
設問３　ｆ－ウ，　ｇ－ア

設問１

ａ：セルF3には，前月の販売利益を求める。

前月の販売利益
＝前月の販売金額－前月の製造費用
＝(前月の販売実績数×販売単価)－(前月の製造実績数×製造原価)
＝(E3×販売単価)－(D3×製造原価)

ここで，販売単価と製造原価は，ワークシート“製品情報”から該当する製品IDの販売単価と製造原価を垂直照合関数で次のように取得する。

販売単価：垂直照合(A3,製品情報!A2:D31,2,0)
製造原価：垂直照合(A3,製品情報!A2:D31,3,0)

この垂直照合関数を含む式をセルF4～F7に複写したとき，セル範囲は“製品情報!A2:D31”のままとするので，セル範囲の行番号は絶対参照で指定する。したがって，空欄ａに入れる式は「E3＊垂直照合(A3,製品情報!A$2:D$31,2,0)－D3＊垂直照合(A3,製品情報!A$2:D$31,3,0)」となる。

ｂ：セルC11には，今月の需要予測値を求める。ここで，αは指数平滑定数である。

今月の需要予測値
＝前月の販売実績数×α＋前月の需要予測値×(1－α)
＝E3×F9＋C3×(1－F9)

また，今月の需要予測値は，計算結果の一の位を四捨五入して10個単位で求める。四捨五入関数の桁位置は小数第１位の桁を０とし，右方向を正として数えたときの位置である。つまり，一の位で四捨五入する場合の桁位置は，左方向（負）に１ずらした－1となる。したがって，空欄ｂに入れる式は「四捨五入(E3＊F$9＋C3＊(1－F$9),－1)」となる。なお，セルC12～C15に複写するので，指数平滑定数α（セルF9）の行番号を絶対参照で指定している。

ｃ：セルE11には，今月の製造ロット数を求める。

今月の製造ロット数
＝↑今月の最低製造数÷製造単位↑　（↑...↑は，切上げを表す）
＝↑(必要在庫数－前月の月末在庫数)÷製造単位↑
＝↑(今月の需要予測値＋安全在庫数－前月の月末在庫数)÷製造単位↑
＝↑(C11＋D11－B11)÷製造単位↑

ここで，製造単位はワークシート“製品情報”から求めることもできるが，“安全在庫数＝製造単位”なので，セルD11を利用すると，式を次のように変形できる。

今月の製造ロット数
＝↑(C11＋D11－B11)÷製造単位↑
＝↑(C11＋D11－B11)÷D11↑
＝↑(C11－B11)÷D11＋D11÷D11↑
＝↑(C11－B11)÷D11＋1↑

したがって，この式を「切上げ((C11－B11)／D11＋1,0)」と切上げ関数の引数とする。

設問２

d：問題に示された手順で，製品ID“S01”の製品の今月の需要数を450個として，製造ロット数の変更前（製造ロット数＝12）の販売利益を求めると，次のようになる。

今月の総在庫数＝12ロット×30個／ロット＋107個＝467個

今月の販売実績数＝450個（“総在庫数≧需要数”；全て販売可能）

今月の販売利益＝450個×800円／個－360個×420円／個

－(450個－450個)×800円／個

＝「208,800」円

e：問題に示された手順で，製品ID“S01”の製品の今月の需要数を450個として，製造ロット数の変更後（製造ロット数＝11）の販売利益を求めると，次のようになる。ここで，“総在庫数＜需要数”なので，総在庫数の分しか販売できないことに注意する。

今月の総在庫数＝11ロット×30個／ロット＋107個＝437個

今月の販売実績数＝437個（“総在庫数＜需要数”；総在庫数の分だけ販売可能）

今月の販売利益＝437個×800円／個－330個×420円／個

－(450個－437個)×800円／個

＝「200,600」円

設問３

f：〔マクロ：GetProfitの説明〕(3)の記述“セルA14に，需要予測値の＋20%，＋15%，＋10%，…，－15%，－20%と５%刻みの需要数を格納”に該当する処理である。この繰返しで使用している変数cntとセルA14に格納する需要数の関係を表にまとめると，次のようになる。

cnt	セルA14に格納する需要数
0	需要予測値の＋20%　(B11＊1.20)
1	需要予測値の＋15%　(B11＊1.15)
2	需要予測値の＋10%　(B11＊1.10)
3	需要予測値の＋5%　(B11＊1.05)
4	需要予測値　(B11＊1.00)
5	需要予測値の－5%　(B11＊0.95)
6	需要予測値の－10%　(B11＊0.90)
7	需要予測値の－15%　(B11＊0.85)
8	需要予測値の－20%　(B11＊0.80)

つまり，変数cntの値が１増えるごとに，需要予測値（セルB11）に掛ける値を0.05ずつ小さくしていけばよいので「B11 ＊ (1.20 － cnt ＊ 0.05)」となる。

g：計画ロット数の最高利益（maxProfit）と平均利益（sumProfit ／ cnt）を，該当する製品IDの計画ロット数のセルに代入した後の処理である。これは，〔マクロ：GetProfitの説明〕(4)の“結果を，該当する製品IDの計画ロット数のセルに格納する”に該当する処理なので，次の(5)の記述“セルC11の値を調整ロット数に変更し，計画ロット数のときと同じ手順で最高利益と平均利益を求め，該当する製品IDの調整ロット数のセルに格納する”に該当する処理の準備をすればよい。つまり，セルC11の値を調整ロット数（計画ロット数を１ロット減らした製造ロット数）に変更すればよいので，現在の製造ロット数（計画ロット数）から１を減算する処理「C11 ← C11 － 1」となる。

配点表

必須	問1	a	b	c	d	e	f	g				a～c 2×3＝6 d，e 3×2＝6 f，g 4×2＝8 20
		イ	オ	エ	イ	エ	ウ	ア				
4問中2問を選択	問2	a	b	c	d	e						3×5＝15 15
		カ	イ	ウ	カ	ウ						
	問3	a	b	c	d	e						3×5＝15 15
		イ	イ	イ	ウ	エ						
	問4	a	b	c	d	e	f					a～c 2×3＝6 d～f 3×3＝9 15
		エ	ウ	イ	ウ	ア	ア					
	問5	a	b	c	d	e	f					a～c 2×3＝6 d～f 3×3＝9 15
		イ	エ	イ	イ	ク	オ					
必須	問6	a	b	c	d	e	f	g				a～c 3×3＝9 d～g 4×4＝16 25
		エ	ウ	オ	ウ	オ	ウ	イ				
5問中1問を選択	問7	a	b	c	d	e						5×5＝25 25
		オ	ア	ア	ア	ウ						
	問8	a	b	c	d	e	f	g				a～c 3×3＝9 d～g 4×4＝16 25
		イ	ク	ウ	エ	イ	イ	イ				
	問9	a	b	c	d	e	f	g				a～d 4×4＝16 e～g 3×3＝9 25
		ウ	ア	ウ	ウ	イ	ウ	オ				
	問10	a	b	c	d	e	f	g				a～d 4×4＝16 e～g 3×3＝9 25
		ウ	ア	ア	エ	ウ	ク	エ				
	問11	a	b	c	d	e	f	g				a～c 3×3＝9 d～g 4×4＝16 25
		エ	ウ	イ	カ	エ	ウ	ア				

（注）●答えを二つ選ぶ場合の配点
正解が「ア，エ」の場合は，解答欄にアとエの二つが記入されている場合にのみ得点とする。
●答えが順不同の場合の配点
正解が「ａ－ア，ｂ－イ(ａとｂは順不同)」の場合は，ａがイ，ｂがアの場合も得点とする。