

平成 26 年度春期試験対応 基本情報技術者模擬試験

【午後 解答解説】

目 次

●解答	1
●解説	3
問 1	3
問 2	9
問 3	19
問 4	24
問 5	39
問 6	48
問 7	52
問 8	55
問 9	66
問 10	69
問 11	71
問 12	73
問 13	79

2014 05 FE mogi A AM

●解答

問1 12点

設問1 a ウ b ウ c ア d イ (各1点)

設問2 e ア f イ g ウ h エ (各2点)

(g, hは順不同)

問2 12点

設問 a:ク b:オ c:カ d:エ (各3点)

問3 12点

設問 a:ウ b:エ c:エ d:オ e:ア (a~c各2点 d, e各3点)

問4 12点

設問1 a:キ b:ウ (各3点)

設問2 エ (3点)

設問3 ア (3点)

問5 12点

設問1 イ, ウ, オ (完答3点)

設問2 a:エ b:カ c:イ (各3点)

問6 12点

設問 a:オ b:イ c:キ (各4点)

問7 12点

設問1 a:ウ b:ア (各3点)

設問2 エ (各3点)

設問3 ア (各3点)

問8 20点

設問 a:ウ b:オ c:イ d:オ e:エ (各4点)

問 9 **20 点**

設問 1 ア (**5 点**)

設問 2 イ (**5 点**)

設問 3 ア (**5 点**)

設問 4 イ (**5 点**)

問 10 **20 点**

設問 1 a カ b オ c エ d エ e オ (**各 3 点**)

設問 2 ア (**5 点**)

問 11 **20 点**

設問 1 a カ b ウ c オ d ア (**各 4 点**)

設問 2 ウ (**2 点**)

設問 3 ウ (**2 点**)

問 12 **20 点**

設問 1 a ウ b エ c オ (**各 4 点**)

設問 2 ウ (**4 点**)

設問 3 イ (**4 点**)

問 13 **20 点**

設問 1 a ウ b エ c ウ d イ e ア (**各 1 点**)

設問 2 f オ g カ h カ i ア j カ (**各 1 点**)

設問 3 k ウ l ウ m イ n ウ o エ (**各 2 点**)

●解説

問 1

解答：

設問 1

a ウ b ウ c ア d イ

設問 2

e ア f イ g ウ h エ

(g, h は順不同)

解説：

インターネット上でのデータ通信に関連する技術（IP アドレス、ファイアウォールのフィルタリングなど）が主題となっています。

まず、IP アドレスについての基礎知識について説明し、その後に解答を示します。

【IP アドレスのネットワーク部のビット数】

IP アドレスは、インターネット上のコンピュータやネットワーク接続機器などを全世界中で一意に識別するために付与されるもので、電話網の「電話番号」などに相当します。

IP アドレスは 32 ビット（4 バイト）で表現され、32 ビット中の上位ビットを「ネットワーク部（ネットワークアドレス）」、残りの下位ビットを「ホスト部（ホストアドレス）」に区分しています。

IP アドレスのうち、ネットワーク部は組織・団体などを区別する部分で、電話番号の「市外局番」および「市内局番」に相当します。ホスト部は、組織・団体内のコンピュータなどの機器（以下「ホスト」とする）を一意に識別するための部分で、電話番号の「加入者番号」にあたります。

例えば“03-xxxx-……”という電話番号なら、“03”という市外局番より「東京 23 区内のいずれか」の番号であると判断ができるのと同様に、IP アドレスのネットワーク部によって、その IP アドレスをもつホストが、どの組織等に所属しているかの判別が可能となります。

IP アドレスは、以下に示す各種の「クラス」に分類されます。「クラス」とは IP アドレスの範囲を示す単位です。

クラス A … 大規模組織のネットワークに対して与えられる。ネットワーク部に 1 バイト（8 ビット）、ホスト部に残り 3 バイト（24 ビット）を割り当てる形式。クラス A の IP アドレスを割り当てられた組織は、最大で約 1600 万（ $2^{24} = 16777216$ ）のホスト部のアドレスを所有できる。

- クラス B … 中規模な組織などのネットワークに対して与えられる。ネットワーク部に 2 バイト (16 ビット)、ホスト部に残り 2 バイト (16 ビット) を割り当てる形式。クラス B の IP アドレスを割り当てられた組織は、最大で約 6 万 5 千 ($2^{16}=65536$) のホスト部のアドレスを所有できる。
- クラス C … 小規模な組織などのネットワークに対して与えられる。ネットワーク部に 3 バイト (24 ビット)、ホスト部に残り 1 バイト (8 ビット) を割り当てる形式。クラス C の IP アドレスを割り当てられた組織は、最大で約 250 ($2^8=256$) のホスト部のアドレスを所有できる。

【サブネット】

規模が大きい LAN では、LAN 全体を一つのネットワークとするより、LAN を細分化して小規模なネットワーク (サブネットワーク、又はサブネットと呼ぶ) を複数作成し、サブネット内の PC は同じサブネット内の別の PC とだけ通信可能にするという形態にした方が、業務上便利な場合があります。

ネットワーク内にこのサブネットを構築するには、IP アドレスのホスト部のうちの一部を、サブネットを区別するための別のアドレスとして扱うことになります。

【サブネット分割しないネットワーク】

IP アドレス：

ネットワーク部	ホスト部
---------	------

↓

【サブネット分割したネットワーク】

IP アドレス：

ネットワーク部	サブネット部	ホスト部
---------	--------	------

IP アドレスのうち、ホスト部の上位の数ビットをサブネット識別のためのアドレス「サブネット部」とします。サブネットに分割された各ネットワークは、ネットワーク部とサブネット部をひとまとめでしたアドレスによって区別されます。

このサブネット部のビット数により、ネットワークを細分化できる個数 (サブネットの個数) が決定します。例えばサブネット部を 3 ビット確保した場合、 $2^3=8$ 個の異なるビット列を得られます。しかし、規定によりビットがすべて 0 またはすべて 1 のパターンはサブネットに割り当てられるアドレスとして使用できないため、最大 $8-2=6$ 個のサブネットにネットワークを分割できます。

IP アドレスの上位何ビットまでが、サブネットを区別するための「ネットワーク部とサブネット部」であるかを表現するには、サブネットマスクを用います。サブネットマスクは、「ネットワーク部とサブネット部」が IP アドレスの左から何ビット目までを占めているかを表し、「ネットワーク部とサブネット部」にあたる部分を“1”のビットで、ホスト部に当たる部分を“0”のビットで記載します。

例えば、クラス C（ネットワーク部＝24 ビット、ホスト部＝8 ビット）の IP アドレスをもつネットワークを 5 個のサブネットに分割する場合、サブネット部のビット数は 3 ビットあれば十分です。よって、サブネットマスクの値は以下のようになります。

		↓サブネット部	
ネットワーク部			ホスト部
1111 1111 . 1111 1111. ...	1111 . 111	00000	
↓ (10 進数に)			
255 . 255 . 255	. 224	と表す	

IP アドレスから、その IP アドレスをもつホストが所属するネットワークのネットワーク部およびサブネット部のみを取り出すには、その IP アドレスと、サブネットマスクとの AND（論理積）をとればよいことになります。例えば、“192. 168. 10. 138”という IP アドレスのサブネットマスクが上記の値（255. 255. 255. 224）である場合、以下のようになります。

$$\begin{array}{rcl}
 192. 168. 10. 138 & = & 11000000. 10101000. 00001010. 10001010 \\
 255. 255. 255. 224 & = & 11111111. 11111111. 11111111. 11100000 \quad (\text{AND}) \\
 \hline
 & & 11000000. 10101000. 00001010. 10000000 \\
 & = & 192. 168. 10. 128
 \end{array}$$

すなわち、この IP アドレスの所属するネットワークのネットワーク部およびサブネット部のアドレスの値は“192. 168. 10. 128”となります。

・設問 1

・空欄 a, b

本問の社内ネットワークではサブネットマスクに 255. 255. 255. 0 を使用しています。例として、社内ネットワークの PC1 の IP アドレス（192. 168. 100. 101）と、このサブネットマスクとの AND をとると、以下のようになります。

$$\begin{aligned}
 192.168.100.101 &= \boxed{11000000.10101000.01100100.01100101} \\
 255.255.255.0 &= \boxed{11111111.11111111.11111111.00000000} \quad (\text{AND}) \\
 &= 11000000.10101000.01100100.00000000 \\
 &= 192.168.100.0
 \end{aligned}$$

社内ネットワーク中の他の PC またはファイルサーバなどの IP アドレスを用いても、すべて同じネットワークアドレスとなります。

すなわち、社内ネットワークのネットワークアドレスは“192.168.100.0（空欄 a=ウ）”となり、またサブネットマスクの 1 のビットの数から、ネットワークアドレス部の長さは 24 ビット（空欄 b=ウ）となります。

・空欄 c, d

PC1 もファイルサーバも、ネットワークアドレスの値が同一となる（空欄 c=ア）ため、同じネットワークに所属することになります。よって、このような場合はルータなどを経由して外部にパケットを送る必要はなく、送信元のホストからあて先のホストに直接パケットが送られます。

なお、PC1 からインターネット上の外部の Web サーバにアクセスする場合、外部の Web サーバは異なるネットワークに所属するため、ネットワークアドレスの値は異なる（空欄 d=イ）ことになります。よって、このような場合はルータなどを経由して外部にパケットを送る必要があるため、各 PC にあらかじめ登録しておいたデフォルトゲートウェイ（本問では、ルータが相当する）にパケットを送付し、ルータ経由で外部にパケットを送付します。

・設問 2

【ファイアウォール】

ファイアウォールはセキュリティを向上させるためのネットワーク機器の一種で、以下のような機能を持ちます。

- ・外部から社内に到達するパケット、および社内から外部に出て行くパケットの内容を検査し、あらかじめ許可した IP アドレスや利用プロトコルなどの条件を用いて、パケットを評価し、許可されたパケットのみを通過させる。
- ・許可されていない不審なパケットは、通過を許可せず遮断する。これにより、外部からの不正アクセスなどを防止する。

ファイアウォールにて行われる、代表的なフィルタリング方式を以下にまとめます。

・パケットフィルタリング方式

ネットワーク層からトランスポート層において、パケット制御を行います。IP パケットの送信元・あて先 IP アドレスやポート番号の内容により、フィルタリングを行います。な

お、この方式ではパケットのヘッダ部分のみを参照し、パケットに搭載されたデータ自体の内容は評価しないため、Web コンテンツやメール本文などの内容によってフィルタリングすることはできません。

- ・アプリケーションゲートウェイ方式

アプリケーション層レベルでの制御を行います。利用者のアプリケーションごとに、パケットの中継制御を行います。送受信される内容自体をフィルタリングの対象とすることが可能です。

A 社のルータはファイアウォール機能をもっています。この機能により、ルータを通過するすべてのパケットが問題文表 1 のアクセス制御リストの条件に一致するかが検査され、番号 1～12 までの条件に一致するパケットのみを通過させていることがわかります。例えば、番号 1 は、発信元の IP アドレスが PC1～10 であり、あて先の IP アドレスが ISP の SMTP サーバで、利用プロトコルが SMTP となっています。この条件は、「PC1 などから ISP の SMTP サーバに向けて送られた、SMTP プロトコルでの問合せの通信用パケット（メール送信など）」となります。

TCP/IP での一般的な通信プロトコルでは、クライアント（PC など）からサーバに問合せがあると、サーバはその応答パケットをクライアントに返送し、処理の正否などをクライアントに伝える必要があります。よって、PC1 などから ISP の SMTP サーバに向けて SMTP プロトコルでの問合せが到達すると、ISP の SMTP サーバはその応答パケットを PC1 などに返送します。

もし、この応答パケットがルータによって遮断されてしまうと、サーバからの応答を正常に受け取れないため、PC 上のメーラーなどは正常なメールの送受信処理を実行できません。よって、この応答パケットも通過を許可するために、問題文表 1 の番号 2 の条件（発信元の IP アドレスが ISP の SMTP サーバであり、あて先の IP アドレスが PC1～10 で、利用プロトコルが SMTP）が必要です。すなわち、番号 1 と 2 は、PC1～10 と ISP の SMTP サーバとの SMTP による通信を可能にするためのペアの条件となります。

問題文表 1 の番号 3 と 4、5 と 6 などの条件も、各プロトコル（POP3、DNS など）について、PC1～10 と ISP のサーバもしくは外部の Web サーバとの通信を可能にするためのペアの条件です。

【パーソナルファイアウォール】

パーソナルファイアウォールは、物理的な機器ではなく、PC 上で稼働するセキュリティ対策用プログラムで、PC に出入りするパケットの内容を検査し、あらかじめ許可した IP アドレスや利用プロトコルなどの条件に合致するパケットのみを許可する機能を持ちます。このパーソナルファイアウォールでは、通常のファイアウォールよりもさらに細かい条件（利用するアプリケーションなど）でパケットのフィルタリングが可能なため、より細か

なアクセス制限を課すことができます。

- ・ 空欄 e, f

問題文表 2 を参照すると, PC10 の Web ブラウザ B が発信元またはあて先となる HTTP プロトコルの通信は番号 7, 8 によって許可されていますが, PC10 の電子メールソフト A が発信元またはあて先となる HTTP プロトコルの通信は, 1~12 のどの番号でも許可されていないため, 通過禁止となります。よって, 同じ HTTP 通信であっても, PC10 の Web ブラウザ B からの通信 (空欄 e=ア) は許可されますが, PC10 の電子メールソフト A からの通信 (空欄 f=イ) は禁止されます。

- ・ 空欄 g, h

問題文表 1 と同表 2 を比較すれば, 容易に解答可能です。各解答群について, 同表 1 および同表 2 の条件にて許可されるかを検証します。

- ・ ア (PC10 の Web ブラウザ B からの HTTP 通信)

この通信は, 表 1 (番号 7, 8), 表 2 (番号 7, 8) とともに許可されるため, 問題文の条件 (ルータ上では許可されるがパーソナルファイアウォールでは禁止) に一致しません。

- ・ イ (PC10 の Web ブラウザ B からの SSH 通信)

SSH (Secure Shell) とは, ネットワーク経由で遠隔地のコンピュータにアクセスするためのプロトコルです。この通信は, 表 1, 表 2 とともに許可されていないため, 問題文の条件に一致しません。

- ・ ウ (PC10 の業務ソフト C からの SMTP 通信)

この通信は, 表 1 では番号 1, 2 にて許可されますが, 表 2 では許可されないため, 問題文の条件に一致します。

- ・ エ (PC10 の電子メールソフト A からの HTTPS 通信)

この通信は, 表 1 では番号 9, 10 にて許可されますが, 表 2 では許可されないため, 問題文の条件に一致します。

- ・ オ (PC10 のファイル転送ソフト D からの FTP 通信)

この通信は, 表 1, 表 2 とともに許可されていないため, 問題文の条件に一致しません。

以上より, 解答群ウ, エのみが問題文の条件に一致します。

問 2

解答：

設問

a：ク b：オ c：カ d：エ

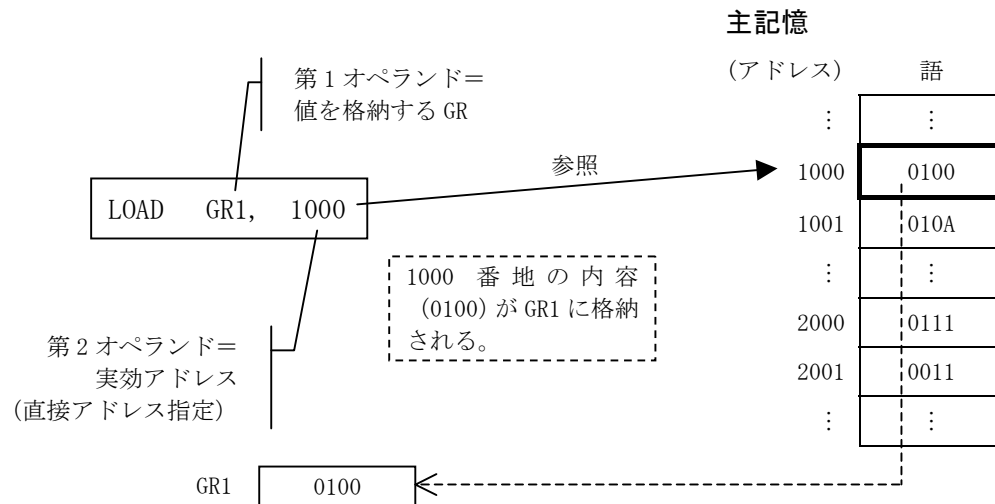
解説：

指定された命令語の形式にしたがって処理を行う架空の計算機を用いて、主記憶に格納されたプログラム中の複数の命令を実行した結果をトレースすることを主題としています。問題文中に示された命令語の形式を正確に把握し、主記憶中の各命令の内容を正しくとらえれば、容易に解答できます。

なお、本解説では「汎用レジスタ」を、「GR」と略記します。例えば、「汎用レジスタ 1」は、「GR1」とします。

【直接アドレス指定】

直接アドレス指定は、命令が参照する語のアドレスを示す値である実効アドレスを指定する形式の一つで、命令中のオペランドに示された値をそのまま実効アドレスとして用いる方法をとります。



(オペランド、アドレスや語の数値はすべて 16 進数である。以下同じ)

図 1

図 1 の命令 (LOAD) は、直接アドレス指定によって求めた実効アドレス (第 2 オペランド) が指す語の内容を、指定した GR (第 1 オペランド) に格納する処理を行う架空の命令です。“LOAD GR1, 1000” という形式でこの命令を実行すると、第 2 オペランドの値がそのまま実効アドレスとなるため、実効アドレス=1000 となり、1000 番地の語の内容 (0100)

が GR1 に格納されます。

直接アドレス指定では、命令に示された値がそのまま実効アドレスとなるため、アドレスの参照を簡単に行うことができます。本問の命令を解釈する際に間接アドレス指定を行わない場合は、直接アドレス指定によって実効アドレスの値が決定されます。

【間接アドレス指定】

直接アドレス指定は、実効アドレスを指定する形式の一つで、命令中のオペランドに示されたアドレスが指す語に格納された値を取り出し、その値を実効アドレスとして用いる方法をとります。

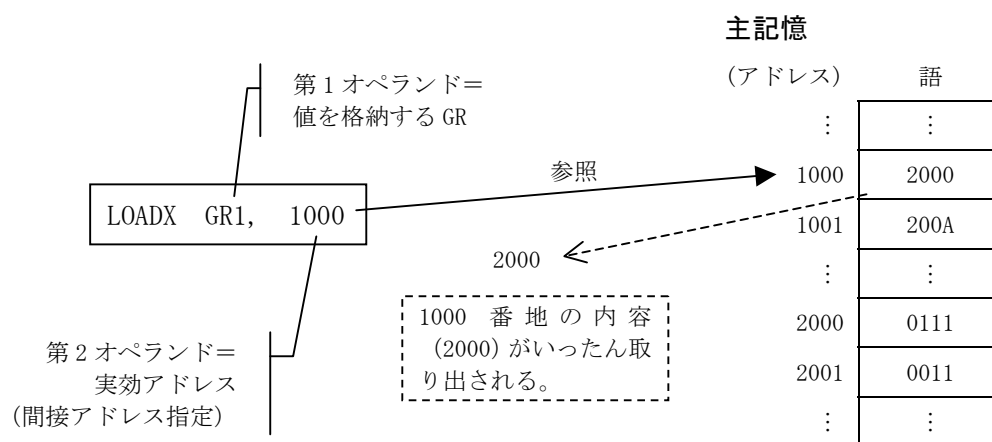


図 2

図 2 の命令 (LOADX) は、間接アドレス指定によって、第 2 オペランドの値から求めた実効アドレスが指す語の内容を、指定した GR (第 1 オペランド) に格納する処理を行う架空の命令です。“LOADX GR1, 1000” という形式でこの命令を実行すると、第 2 オペランドの値 (1000) をそのまま実効アドレスとするのではなく、1000 番地の語の内容 (2000) がいったん取り出され、取り出された 2000 という値が実効アドレスとなります。そして、2000 番地 (実効アドレス) が指す語、すなわち 2000 番地の語に格納されている 0111 という値が、GR1 に格納されます (図 3)。したがって、この命令の実行直後の GR1 の値は、1000 番地に格納されている値 (2000) ではなく、2000 番地に格納されている値 (0111) となります。

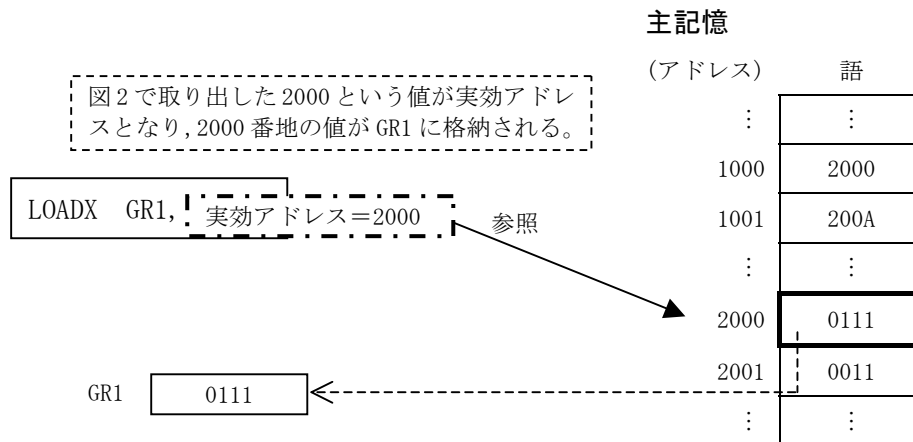


図 3

間接アドレス指定では、命令に示された値がそのまま実効アドレスとならないため、オペランドに示された値のアドレスの語をいったん参照し、その内容を実効アドレスとして、実効アドレスが指す語を再度参照することになります。よって、直接アドレス指定と比較して、アドレスの参照を行う処理が複雑になります。

【設問】

本問のプログラムの最初の命令が解釈されるときの実行過程をトレースします。

GR0 : 0003 GR1 : 0000 GR2 : 0000 GR3 : 0000
PR : 0100

主記憶								
アドレス	+0	+1	+2	+3	+4	+5	+6	+7
00F8	0000	0000	0000	0000	0000	0000	0000	0000
0100	2100	011B	20C0	0003	2170	0111	21B8	011A
0108	FF00	0000	0000	0000	0000	0000	0000	0000
0110	0000	0001	0002	0003	0004	0005	0006	0007
0118	0110	0111	0112	0113	0114	0115	0116	0117
0120	0118	0119	011A	011B	011C	011D	011E	011F

図 4

図 4 は、プログラム開始時点での GR0～GR3、PR（プログラムレジスタ）及び主記憶の値を示しています。

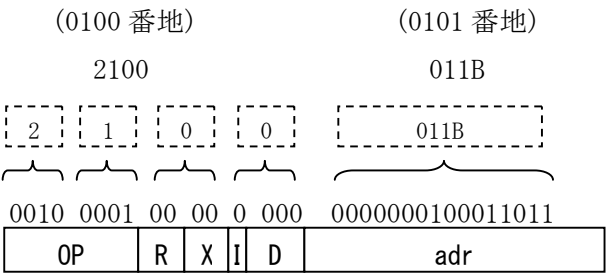
PR は、参照すべき命令が格納されている語のアドレスを格納するレジスタです。PR に 0100_{16} という値が設定されているため、プログラムを実行すると、主記憶の 0100 番地から 0101 番地の 2 語分の内容（図 5 の太線囲み部分）が、命令として最初に取り出され、解釈されます。

GR0 : 0003 GR1 : 0000 GR2 : 0000 GR3 : 0000
PR : 0100

主記憶								
アドレス	+0	+1	+2	+3	+4	+5	+6	+7
00F8	0000	0000	0000	0000	0000	0000	0000	0000
0100	2100	011B	20C0	0003	2170	0111	21B8	011A
0108	FF00	0000	0000	0000	0000	0000	0000	0000
0110	0000	0001	0002	0003	0004	0005	0006	0007
0118	0110	0111	0112	0113	0114	0115	0116	0117
0120	0118	0119	011A	011B	011C	011D	011E	011F

図 5

上記の命令のビット列を、本問に示された OP, R, X, I, D 及び adr として取り出して解釈した結果を示します。



OP : 0010 0001 = 21_{16}
R : 00 = 0 (GR0)
X : 00 = 0 (指標レジスタによる修飾は行わない)
I : 0 (直接アドレス指定)
D : 常に 0
adr : 0000000100011011 = 011B (実効アドレス)

命令コード = 21_{16} となるため、実効アドレスが示す語の内容を、R で指定された汎用レジスタ、すなわち GR0 に格納することになります。直接アドレス指定が用いられるため、adr の値がそのまま実効アドレスとなります。また、指標レジスタによる修飾は行われません。

以上から、実効アドレス (011B) が示す語の内容、すなわち図 6 の 011B 番地の内容が GR0 に格納されるため、GR0 の内容は 0113_{16} となります。

主記憶								
アドレス	+0	+1	+2	+3	+4	+5	+6	+7
00F8	0000	0000	0000	0000	0000	0000	0000	0000
0100	2100	011B	20C0	0003	2170	0111	21B8	011A
0108	FF00	0000	0000	0000	0000	0000	0000	0000
0110	0000	0001	0002	0003	0004	0005	0006	0007
0118	0110	0111	0112	0113	0114	0115	0116	0117
0120	0118	0119	011A	011B	011C	011D	011E	011F

0118 番地からアドレスが+3 された番地=0118+3=011B となる。

この値がGR0に格納される。

図 6

以下、同様にしてプログラムの残りの命令を解釈していきます。

・ 2 番目の命令

本問の計算機の命令語=2 語のため、先ほど解釈した命令の次に実行される命令は、現在 PR が指しているアドレスの 2 語分後ろに存在することになります。よって、次の命令の先頭アドレスを正しく指すため、PR の値は 2 増加します (図 7)。

GR0 : 0113 GR1 : 0000 GR2 : 0000 GR3 : 0000
PR : 0102

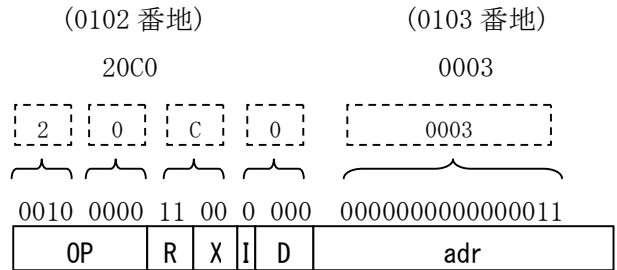
主記憶								
アドレス	+0	+1	+2	+3	+4	+5	+6	+7
00F8	0000	0000	0000	0000	0000	0000	0000	0000
0100	2100	011B	20C0	0003	2170	0111	21B8	011A
0108	FF00	0000	0000	0000	0000	0000	0000	0000
0110	0000	0001	0002	0003	0004	0005	0006	0007
0118	0110	0111	0112	0113	0114	0115	0116	0117
0120	0118	0119	011A	011B	011C	011D	011E	011F

図 7

よって、0102 番地から 0103 番地の 2 語分 (図 7 の太線囲み部分) が、命令として解釈さ

れ，実行されます。

この命令のビット列を，OP，R，X，I，D 及び adr として取り出して解釈した結果を示します。



OP : 0010 0000 = 20₁₆

R : 11 = 3 (GR3)

X : 00 = 0 (指標レジスタによる修飾は行わない)

I : 0 (直接アドレス指定)

D : 常に 0

adr : 0000000000000011 = 0003 (実効アドレス)

命令コード = 20₁₆ となるため，実効アドレスそのものの値を，R で指定された汎用レジスタ，すなわち GR3 に格納することになります。直接アドレス指定が用いられるため，adr の値がそのまま実効アドレスとなります。また，指標レジスタによる修飾は行われません。

以上から，実効アドレスの値 (0003) が GR3 に格納されるため，GR3 の内容は 0003₁₆ となります。

・ 3 番目の命令

次の命令を取り出すため，PR の値が 2 増加します (図 8)。

GR0 : 0113 GR1 : 0000 GR2 : 0000 GR3 : 0003

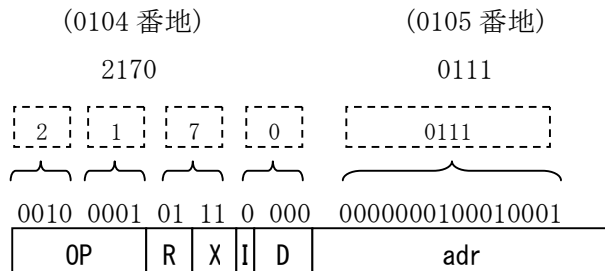
PR : 0104

主記憶								
アドレス	+0	+1	+2	+3	+4	+5	+6	+7
00F8	0000	0000	0000	0000	0000	0000	0000	0000
0100	2100	011B	20C0	0003	2170	0111	21B8	011A
0108	FF00	0000	0000	0000	0000	0000	0000	0000
0110	0000	0001	0002	0003	0004	0005	0006	0007
0118	0110	0111	0112	0113	0114	0115	0116	0117
0120	0118	0119	011A	011B	011C	011D	011E	011F

図 8

よって、0104 番地から 0105 番地の 2 語分（図 8 の太線囲み部分）が、命令として解釈され、実行されます。

この命令のビット列を、OP, R, X, I, D 及び adr として取り出して解釈した結果を示します。



OP : 0010 0001 = 21_{16}

R : 01 = 1 (GR1)

X : 11 = 3 (GR3 を指標レジスタとする)

I : 0 (直接アドレス指定)

D : 常に 0

adr : 0000000100010001 = 0111 (実効アドレス)

命令コード = 21_{16} となるため、実効アドレスが示す語の内容を、R で指定された汎用レジスタ、すなわち GR1 に格納することになります。直接アドレス指定が用いられ、かつ、指標レジスタ GR3 による修飾が行われるため、adr の値 (0111) に GR3 の値 (0003) を加算した値である 0114 が実効アドレスとなります。すなわち、図 9 の 0114 番地の内容が GR1 に格納されるため、GR1 の内容は 0004_{16} となります。

主記憶								
アドレス	+0	+1	+2	+3	+4	+5	+6	+7
00F8	0000	0000	0000	0000	0000	0000	0000	0000
0100	2100	011B	20C0	0003	2170	0111	21B8	011A
0108	FF00	0000	0000	0000	0000	0000	0000	0000
0110	0000	0001	0002	0003	0004	0005	0006	0007
0118	0110	0111	0112	0113	0114	0115	0116	0117
0120	0118	0119	011A	011B	011C	011D	011E	011F

0110 番地からアドレスが +4 された番地 = $0110 + 4 = 0114$ となる。

この値が GR1 に格納される。

図 9

・ 4 番目の命令

次の命令を取り出すため，PR の値が 2 増加します（図 10）。

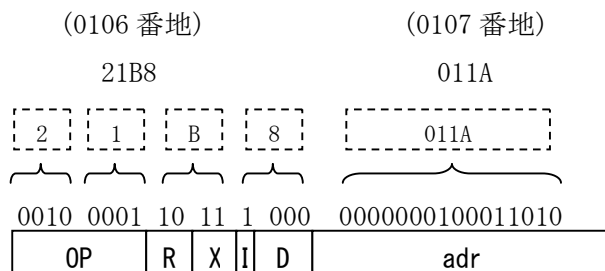
GR0 : 0113 GR1 : 0004 GR2 : 0000 GR3 : 0003
PR : 0106

主記憶								
アドレス	+0	+1	+2	+3	+4	+5	+6	+7
00F8	0000	0000	0000	0000	0000	0000	0000	0000
0100	2100	011B	20C0	0003	2170	0111	21B8	011A
0108	FF00	0000	0000	0000	0000	0000	0000	0000
0110	0000	0001	0002	0003	0004	0005	0006	0007
0118	0110	0111	0112	0113	0114	0115	0116	0117
0120	0118	0119	011A	011B	011C	011D	011E	011F

図 10

よって，0106 番地から 0107 番地の 2 語分（図 10 の太線囲み部分）が，命令として解釈され，実行されます。

この命令のビット列を，OP，R，X，I，D 及び adr として取り出して解釈した結果を示します。



OP : 0010 0001 = 21₁₆

R : 10 = 2 (GR2)

X : 11 = 3 (GR3 を指標レジスタとする)

I : 1 (間接アドレス指定)

D : 常に 0

adr : 0000000100011010 = 011A (実効アドレス)

命令コード = 21₁₆ となるため，実効アドレスが示す語の内容を，R で指定された汎用レジスタ，すなわち GR2 に格納することになります。

この命令では間接アドレス指定が用いられるため、注意が必要です。指標レジスタ GR3 による修飾が行われるため、adr の値に GR3 の値 (0003) が加算された値のアドレス (すなわち、011A+0003=011D) に格納された値が、まず取り出されます (図 11)。

主記憶								
アドレス	+0	+1	+2	+3	+4	+5	+6	+7
00F8	0000	0000	0000	0000	0000	0000	0000	0000
0100	2100	011B	20C0	0003	2170	0111	21B8	011A
0108	FF00	0000	0000	0000	0000	0000	0000	0000
0110	0000	0001	0002	0003	0004	0005	0006	0007
0118	0110	0111	0112	0113	0114	0115	0116	0117
0120	0118	0119	011A	011B	011C	011D	011E	011F

0118 番地からアドレスが+5 された番地=0118+5=011D となる。

この値がいったん取り出される。

図 11

そして、取り出された値 (0115) が、実効アドレスとして解釈されます。したがって、この命令の実効アドレスは、011D ではなく 0115 となります。

以上から、実効アドレス (0115) の語の内容、すなわち図 12 の 0115 番地の内容が GR2 に格納されるため、GR2 の内容は 0005₁₆ となります。

主記憶								
アドレス	+0	+1	+2	+3	+4	+5	+6	+7
00F8	0000	0000	0000	0000	0000	0000	0000	0000
0100	2100	011B	20C0	0003	2170	0111	21B8	011A
0108	FF00	0000	0000	0000	0000	0000	0000	0000
0110	0000	0001	0002	0003	0004	0005	0006	0007
0118	0110	0111	0112	0113	0114	0115	0116	0117
0120	0118	0119	011A	011B	011C	011D	011E	011F

0110 番地からアドレスが+5 された番地=0110+5=0115 となる。

この値が GR2 に格納される。

図 12

・ 5 番目の命令

次の命令を取り出すため、PR の値が 2 増加します（図 13）。

GR0 : 0113 GR1 : 0004 GR2 : 0005 GR3 : 0003

PR : 0108

主記憶								
アドレス	+0	+1	+2	+3	+4	+5	+6	+7
00F8	0000	0000	0000	0000	0000	0000	0000	0000
0100	2100	011B	20C0	0003	2170	0111	21B8	011A
0108	FF00	0000	0000	0000	0000	0000	0000	0000
0110	0000	0001	0002	0003	0004	0005	0006	0007
0118	0110	0111	0112	0113	0114	0115	0116	0117
0120	0118	0119	011A	011B	011C	011D	011E	011F

図 13

よって、0108 番地から 0109 番地の 2 語分（図 13 の太線囲み部分）が、命令として解釈され、実行されます。この命令は OP の部分が“FF₁₆”となっているため、この命令が解釈されると、プログラムの実行を終了することになります。

・空欄 a～d

図 13 の内容から、プログラムの実行終了時には、汎用レジスタ 0（GR0）には“0113₁₆”，汎用レジスタ 1（GR1）には“0004₁₆”，汎用レジスタ 2（GR2）には“0005₁₆”，汎用レジスタ 3（GR3）には“0003₁₆”が格納されることになります。よって、空欄a=“0113₁₆ (ク)”，空欄b=“0004₁₆ (オ)”，空欄c=“0005₁₆ (カ)”，空欄d=“0003₁₆ (エ)”となります。

問 3

解答：

設問

a：ウ b：エ c：エ d：オ e：ア

解説：

LAN の伝送方式の一つである CSMA/CD に関する問題です。伝送路を経由するフレームの到達時間などの計算や、CSMA/CD の処理手順の読解が主題となっています。問題文の内容から解答を容易に求められるため、難易度は低くなっています。

本解説では、接続機器のことを「機器」と略することがあります。

【CSMA/CD】

CSMA/CD では、LAN 上の各機器は伝送路にキャリア（搬送波）を送出することで、他の機器にデータを送信しています。他の機器から送出されたデータのキャリアが、すでに伝送路に存在している場合（伝送路が使用中の場合）は、データの送信を行うことはできません。

なお、伝送路に送出したキャリアが他の機器から送出された別のキャリアと衝突した場合、送出を一旦停止してランダムに決めた時間だけ待機し、データの送出をやり直します。これを再送といいます。衝突の頻度が増加すると再送が頻繁に起こるため、データ伝送のスループットは減少していきます。

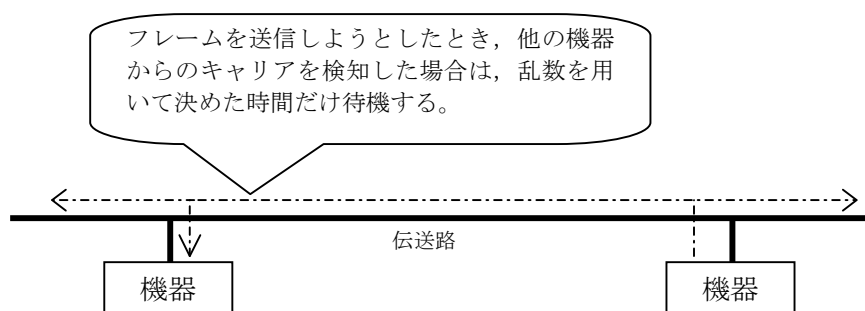
CSMA/CD では、以下のような手順でデータ伝送を行います。

・フレーム送出手順の説明

ステップ 0：機器は、送出するフレームが発生するまで待機する。フレームが発生したら、ステップ 1 に進む。

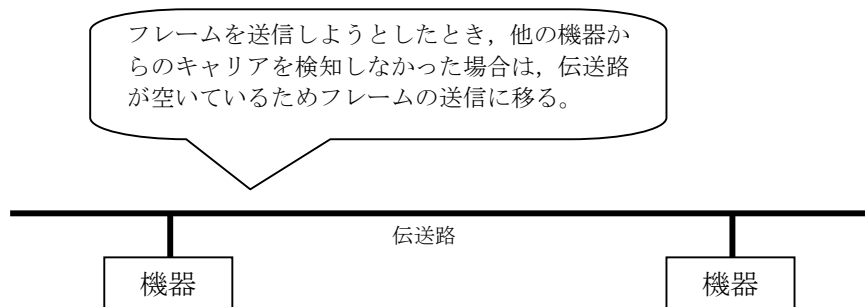
ステップ 1：機器は、伝送路上に他の機器からのフレームが流れていないことを確認してから、フレームの伝送を開始する。伝送路に他の機器からのフレームのキャリアが検知された場合は、ステップ 2 に進む。

<ステップ 1>



ステップ 1 の続き：伝送路に他の機器からのキャリアが検知されない場合は、ステップ 3 に進む。

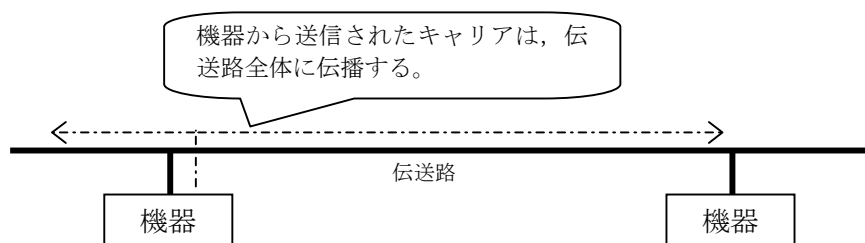
<ステップ 1>



ステップ 2：乱数を用いて決めた時間が経過したら、ステップ 1 に戻る。

ステップ 3：フレーム送出を開始し、ステップ 4 へ移る。

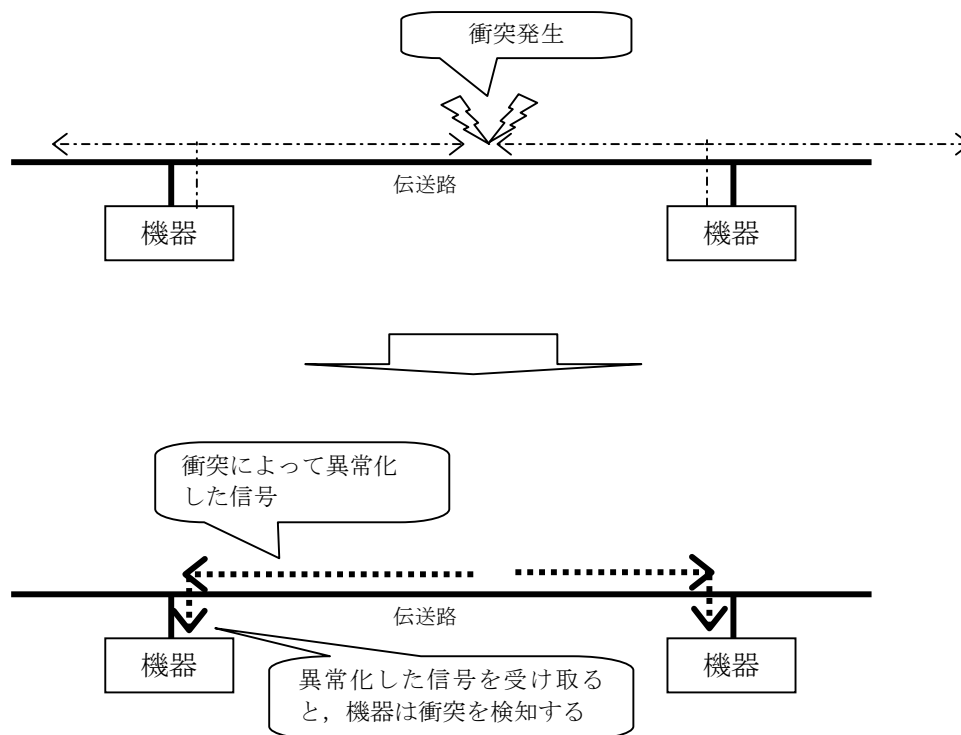
<ステップ 3>



ステップ 4：フレーム送出中に衝突がなければ、送信成功としてステップ 0 に戻る。同時に複数の機器がデータ伝送を開始すると、伝送路上で衝突が発生する。この衝突を

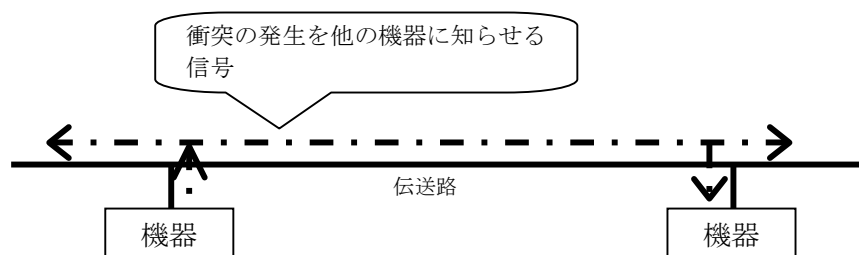
検知した場合、ステップ5に移る。

<ステップ4>



ステップ5: 衝突が発生すると、送出していたフレームが壊れてしまい、データの内容が正常でなくなる。よって、送出中のフレームをこれ以上送信する意味はなくなる。機器は、送出中のフレームを、衝突の発生を他の機器に知らせる信号に切り替えて、一定時間送出する。その後、ステップ2に戻る。この信号によって、他の機器は衝突の発生を知ることができる。機器はステップ2に戻り、その後、フレームの送信をやり直すことになる。

<ステップ5>



【設問】

- ・ 空欄 a, b

接続機器 X と接続機器 Y の間の伝送路の距離（以下、距離という）は 230m、伝送路の信号伝播速度は 230m/マイクロ秒となります。よって、 $230(\text{m}) \div 230(\text{m}/\text{マイクロ秒}) = 1$ マイクロ秒となり、接続機器 X から接続機器 Y に対してフレームを送出すると、1 マイクロ秒で接続機器 Y に到着することになります。

したがって、接続機器 X がフレームの送出を開始してから、1 マイクロ秒が経過する前に接続機器 Y から送出するフレームが発生すると、接続機器 Y がフレームを送出しようとしたときには、他の接続機器からのキャリアが検出されない（接続機器 Y に、接続機器 X からのキャリアが届いていない）ことになります。この場合は、接続機器 Y はステップ 1 からステップ 3 に遷移します。

以上から、空欄a = “1 (ウ)”，空欄b = “ステップ 3 (エ)” となります。

- ・ 空欄 c

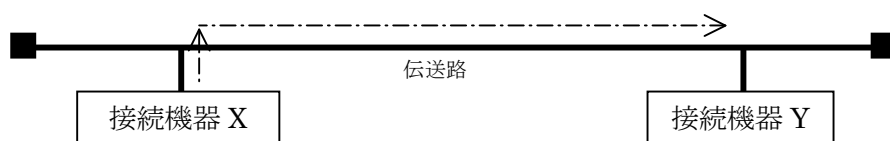
接続機器 X が衝突を検出するまでにかかる時間が最も長くなる場合について考えます。接続機器 X が送出したキャリアが、その途中で他の機器（接続機器 Y）が送出したキャリアと衝突すると、衝突によって破損したキャリア（以下、破損キャリアという）が接続機器 X に入り込みます。このキャリアを確認することで、接続機器 X は衝突を検知します。

接続機器 X に近い位置で衝突が発生した場合は、破損キャリアが接続機器 X に早期に入り込むため、衝突が比較的早く検知されます。接続機器 X に遠い位置で衝突が発生するほど、同機器に破損キャリアが入り込むのに時間がかかり、衝突の検知が遅れると推測できます。

衝突が発生する可能性がある位置のうち、接続機器 X から最も離れているのは、接続機器 Y の位置となります。衝突が発生しないまま接続機器 Y よりも遠くにキャリアが到達した場合は、当該キャリアはすでに接続機器 Y に到達しているため、衝突が発生しないままキャリアを伝送できたことになります。

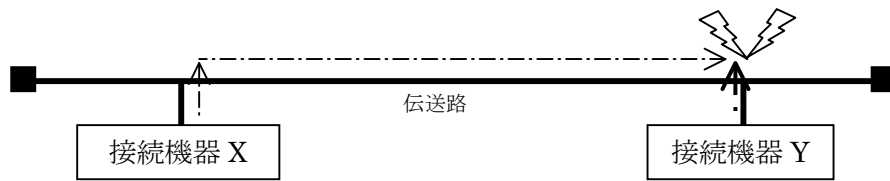
以下のような場合に、接続機器 X が衝突を検出するまでにかかる時間が最も長くなります。なお、以下の解説では、「接続機器 X が衝突を検出するまでにかかる最も長い時間」を、「最長時間」と呼びます。

- (1) 接続機器 X が送出したキャリアが、接続機器 Y の位置に到着した。

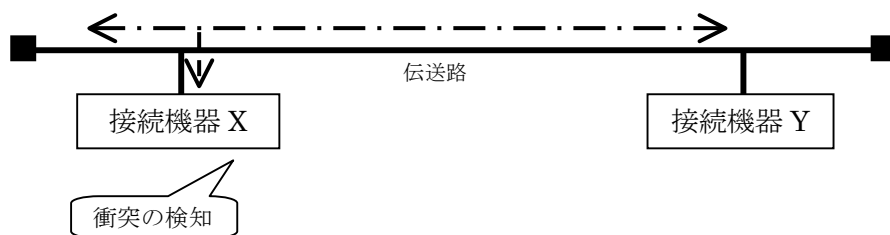


- (2) (1) とほぼ同じタイミングで、接続機器 Y がキャリアを送出し始めた。その結果、接

続機器 Y の位置において衝突が発生した。



- (3) 接続機器 Y の位置において発生した衝突によって生じた破損キャリアが、接続機器 Y の位置から接続機器 X の位置にまで移動し、接続機器 X に入り込んだ。その結果、接続機器 X は衝突を検知した。



以上から、最長時間は、接続機器 X が送出したキャリアが接続機器 Y に到着するまでの時間と、接続機器 Y の位置から接続機器 X の位置まで破損キャリアが到達するまでの時間の和となります。空欄 a の解説から、接続機器 X から接続機器 Y に対してキャリア（フレーム）が到達するのに要する時間は 1 マイクロ秒とわかります。よって、接続機器 X から接続機器 Y に対してキャリアが到達し、接続機器 Y の位置から接続機器 X の位置まで破損キャリアが到達するまでの時間は、 $1+1=2$ マイクロ秒となります。

したがって、空欄 c = “2 (エ)” となります。

・空欄 d, e

最長時間が経過するより前に、接続機器 X からのキャリアの送出を終えてしまうと、破損キャリアが入り込む前にキャリアの送出が終わるため、接続機器 X は衝突を検知できないままキャリア（フレーム）の送出を終えてしまうことになります。したがって、接続機器 X は「フレームを送出しつつ衝突の有無を確認する」ステップであるステップ 4 から、「衝突を検知しないままフレームの送出を終了する」ステップであるステップ 0 に遷移してしまいます。

以上から、空欄 d = “ステップ 4 (オ)”，空欄 e = “ステップ 0 (ア)” となります。

問 4

解答：

設問 1

a：キ b：ウ

設問 2

エ

設問 3

ア

解説：

関係データベースの表を検索する SQL 文（SELECT 文）の評価に関する問題です。副問合せにおいて用いられる EXISTS 句について理解していないと解答しにくい設問があり、難易度が高くなっています。

本解説では、SELECT 文や副問合せについての説明を行い、その後に各設問の解説を行います。

【SELECT 文】

・表の結合

SQL の SELECT 文によって、複数の表から特定の項目（列）を取り出して表示する際には、表の結合を行うのが一般的です。

<例>

売上表

売上番号	年月日	顧客コード
110000	'07-04-21	101
110001	'07-04-21	103

顧客表

顧客コード	氏名
101	山田 一郎
102	上田 次郎
103	川田 三郎

この二つの表から、売上番号 110001 の売上について、その年月日と顧客の氏名のみを取り出して表示することを考えます。このとき、下のような SELECT 文を用いると、処理を正

しく実行できません。

```
SELECT 売上表.年月日, 顧客表.氏名
FROM 売上表, 顧客表
WHERE 売上表.売上番号 = '110001'
```

この SELECT 文では、“FROM 売上表, 顧客表”という指定によって、売上表と顧客表を直積演算によって結びつけた結果の表が作成され、その結果の表から売上表の売上番号の値が'110001'である行のみが、“WHERE 売上表.売上番号 = '110001'”という条件式によって取り出されることになります。

直積演算で二つの表（表 A, 表 B とする）を結合すると、表 A のすべての行と、表 B のすべての行同士をそれぞれ結びつけた結果が出力されます。

売上表と顧客表を直積演算で結合した結果の表

売上番号	年月日	顧客コード	顧客コード	氏名
110000	'07-04-21	101	101	山田 一郎
110000	'07-04-21	101	102	上田 次郎
110000	'07-04-21	101	103	川田 三郎
110001	'07-04-21	103	101	山田 一郎
110001	'07-04-21	103	102	上田 次郎
110001	'07-04-21	103	103	川田 三郎

この結果は、売上表の顧客コードと顧客表の顧客コードが異なる行同士も結び付けられているため、各売上と関係のない顧客も売上に結びついていることになります。よって、この結果から売上表の売上番号の値が'110001'である行のみを取り出すと、顧客コードが 101, 102 の顧客の行も取り出されてしまいます。売上番号の値が'110001'である売上に関係しているのは顧客コードが 103 の顧客のみのため、本来は取り出してはならない顧客の行も、取り出してしまうことになります。

よって、前述の二つの表から、売上番号 110001 の売上の年月日と顧客の氏名のみを正しく取り出して表示するためには、次のような SELECT 文を用いるべきです。

```
SELECT 売上表.年月日, 顧客表.氏名
FROM 売上表, 顧客表
WHERE 売上表.売上番号 = '110001'
      AND 売上表.顧客コード = 顧客表.顧客コード
```

この SELECT 文では、“FROM 売上表, 顧客表”という指定によって、売上表と顧客表を直積演算によって結びつけた結果の表が作成された後に、“売上表.顧客コード = 顧客表.顧客コード”という条件式によって、売上表の顧客コードと顧客表の顧客コードが等しい行のみが最初に取り出されることになります（図 1）。

売上番号	年月日	顧客コード	顧客コード	氏名
110000	'07-04-21	101	101	山田 一郎
110001	'07-04-21	103	103	川田 三郎

図 1

なお、二つの表を結合して図 1 のような表を作成する演算を、等結合演算といいます。以下の解説では、“売上表.顧客コード = 顧客表.顧客コード”のような条件式によって二つの表を等結合演算で結合することを、単に「結合する」と呼ぶことがあります。

そして、図 1 の表から売上表の売上番号の値が'110001'である行のみが、“売上表.売上番号 = '110001'”という条件式によって取り出されることになります。この措置により、顧客コードが 103 の顧客の氏名のみが正しく出力されます。

以上から、複数の表（表 A, 表 B, 表 C, ……とする）を結合する際には、“表 A. 列 X = 表 B. 列 X”, “表 B. 列 Y = 表 C. 列 Y”, ……のように、結合する各表に共通する列の値が等しい行だけを結果として残すための条件式を SELECT 文に加えることが一般的な措置となります。

・副問合せ

副問合せとは、本問の設問 1, 2 にて示されているような、SELECT 文中に別の SELECT 文が()囲みで含まれている形式の SQL 文のことです。このような SELECT 文を実行する際には、()内の SELECT 文（副問合せ）を先に実行して求めた結果を、()の外側の SELECT 文（主問合せ）に与えるという処理が行われます。

売上表

売上番号	年月日	顧客コード
110000	'07-04-21	101
110001	'07-04-21	103
110002	'07-04-23	102

顧客表

顧客コード	氏名
101	山田 一郎
102	上田 次郎
103	川田 三郎

図 2

図 2 の例について、年月日が”07-04-21”である売上と関係する顧客の氏名だけを取り出す処理を、前述のような条件式を用いた表の結合処理を用いず、副問合せを用いて記述することができます。その SELECT 文の例を次に示します。なお、次の SELECT 文中の IN 演算子は、左辺の列の値が右辺の値のいずれかと等しい場合に「真」とする演算子です。

主問合せ
SELECT 顧客表.氏名
FROM 顧客表
WHERE 顧客表.顧客コード IN (副問合せ
SELECT 売上表.顧客コード FROM 売上表
WHERE 売上表.年月日 = "07-04-21")

(IN 演算子は、左の列の値が右の値のいずれかと等しい場合に、真とする)

この SELECT 文では、最初に副問合せ（破線囲み部分）が先に実行され、売上表の顧客コード（101, 103）が取り出されます。その後、主問合せの IN 演算子によって、副問合せによって取り出されたこれらの値のうちのいずれかと顧客コードの値が等しい行のみが、顧客表から取り出されます。その結果、年月日が”07-04-21”である売上と関係する顧客の氏名（山田 一郎、川田 三郎の 2 名）だけが正しく取り出されます。

・副問合せと比較演算子

副問合せを用いるときは、主問合せの列と副問合せの結果とを比較することが多くなり

ます。その際には、“=”などの比較演算子を用いるとエラーになることがあるため、注意が必要です。

主問合せ
SELECT 顧客表.氏名
FROM 顧客表
WHERE 顧客表.顧客コード = (SELECT 売上表.顧客コード FROM 売上表
WHERE 売上表.年月日 = "07-04-21")

(この記述が含まれている SELECT 文は、実行時にエラーとなる可能性がある)

上の SELECT 文は、前述の文の IN 演算子を比較演算子の“=”で置き換えた例です。このようにすると、この SELECT 文の実行時にエラーとなる可能性があります。年月日列の値が”07-04-21”である行が売上表に複数存在する場合、この SELECT 文の副問合せは、複数の顧客コードの値を結果として返してくる可能性があります。SQL 文の規則により、比較演算子の“=”は、右辺及び左辺に単独の行（値）しかもつことができず、右辺または左辺が複数の行（値）になる場合はエラーとなるためです。

なお、以下のような SELECT 文は、比較演算子の“=”を用いても問題なく実行できます。

主問合せ
SELECT 顧客表.氏名
FROM 顧客表
WHERE 顧客表.顧客コード = (SELECT MAX(売上表.顧客コード)
FROM 売上表
WHERE 売上表.年月日 = "07-04-21")

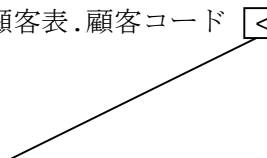
この文の副問合せは、“MAX(売上表.顧客コード)”という集計関数の値を結果として返します。集計関数は、表の列の合計値や平均値または最大値などの単独の値のみを返すため、この文の副問合せは必ず単独の値を返すことが保証されています。よって、実行時にエラーが発生することなく、問題なく実行できます。

また、“<”や“>”のような、等号以外の比較演算子を用いて副問合せの値を評価することも可能です。例えば、以下ようになります。以下の SELECT 文では、副問合せの結果の顧客コードの最大値より小さい値の顧客コードをもつ行のみを、顧客表から抽出する処理を示しています。

主問合せ

```
SELECT 顧客表.氏名
FROM 顧客表
WHERE 顧客表.顧客コード < (
  SELECT MAX(売上表.顧客コード)
  FROM 売上表
  WHERE 売上表.年月日 = "07-04-21" )
```

副問合せ



(副問合せの結果として返ってきた顧客コードの最大値よりも、顧客コードの値が小さい行のみを、主問合せの結果として表示する)

・ EXISTS 句

EXISTS 句（本問では、EXISTS 述語と呼ばれている）は、「EXISTS 句以降の副問合せから 1 行以上の結果が返ってきた場合は真、結果が 1 行も返ってこない場合は偽」という判定を行います。なお、EXISTS 句の直前に NOT 演算子が付き、NOT EXISTS となっている場合は、「EXISTS 句以降の副問合せから 1 行以上の結果が返ってきた場合は偽、結果が 1 行も返ってこない場合は真」という判定が行われます。

売上表

売上番号	年月日	顧客コード
110000	'07-04-21	101
110001	'07-04-21	103
110002	'07-04-23	102

顧客表

顧客コード	氏名
101	山田 一郎
102	上田 次郎
103	川田 三郎

上に示した表について、以下の SELECT 文を実行することを考えます。

```
SELECT 売上番号 FROM 売上表
WHERE EXISTS (SELECT * FROM 顧客表
              WHERE 顧客コード = '101')
```

この SELECT 文を実行すると、副問合せ (SELECT * FROM 顧客表 WHERE 顧客コード = '101') の結果として 1 つの行が返ってくるため、副問合せの結果が存在することになります。よって、EXISTS 句の評価を行った結果は必ず真となります。よって、主問合せのどの行についても WHERE 以下の条件 (EXISTS 句による評価) が真となるため、売上表のすべての行が検索条件に一致するとみなされ、売上表のすべての売上番号が出力されます。

EXISTS 句は、次に説明する相関副問合せと一緒に使用されることが多く、普通の副問合せでは余り使用されません。

- ・相関副問合せ

相関副問合せとは、副問合せの FROM 句では指定されていない表を副問合せ中の条件式において参照しているものを指します。相関副問合せでは、主問合せの表に含まれる行の列の値を利用して条件判断を行っています。

設問 1 の SQL 文などでは、相関副問合せが用いられています。相関副問合せの例を示します。

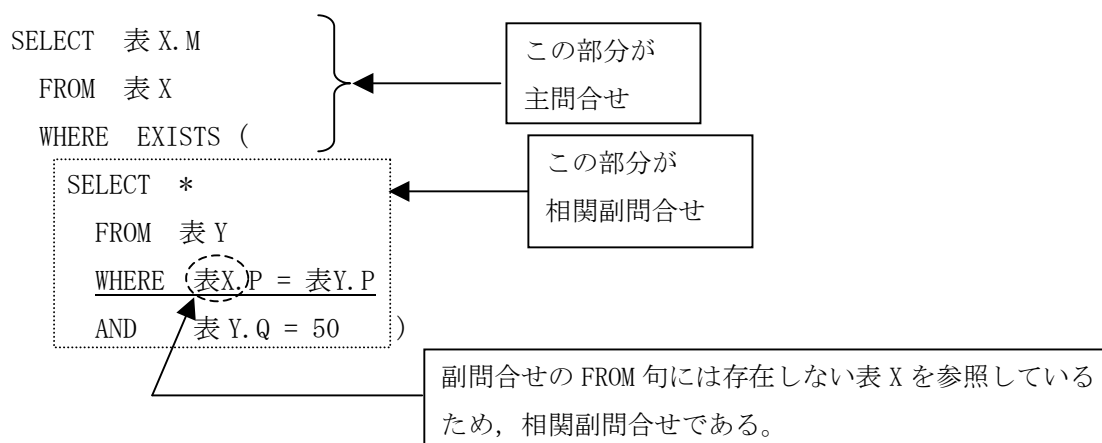


図 3

相関副問合せでは、主問合せで指定されている表 (図 3 では、表 X) の先頭から 1 行ずつ行を受け取り、その行の特定の列の値と、副問合せ中で指定されている表 (図 3 では、表 Y) の各行の列の値とを比較していきます。

例えば、表 X と表 Y が図 4 のようになっているとき、まず表 X の (1) の行 (先頭の行) が取り出されます。

表 X						
M	N	O	P			
101	1	2	10	(1)	表 Y	
102	2	1	20	(2)	P	Q
103	2	3	25	(3)	10	400
					20	40
					10	50
					20	80

図 4

そして、副問合せ中の下線の条件判定によって、表 X の(1)の行の列 P の値（10）と同じ値の列 P をもつ行のみが、表 Y から選別されます。その結果、図 5 の網掛け部分の行のみが表 Y から選別されることになります。

表 Y		
P	Q	
10	400	(1)
20	40	(2)
10	50	(3)
20	80	(4)

図 5

図 5 の網掛け部分の行の中から、WHERE 句の残りの条件（AND 表 Y.Q = 50）によって、列 Q の値が 50 のみのものだけが残されます。よって、図 5 の(1)の行は当該条件を満たさないため残されず、同(3)の行のみが残ることになります。すなわち、図 3 の副問合せでは、結果の行が 1 行返ってくることになります。

EXISTS 句は、EXISTS 句以降の副問合せから 1 行以上の結果が返ってきた場合は真、結果が 1 行も返ってこない場合は偽を返します。よって、図 3 の主問合せでは EXISTS 句での判定が「真」となるため、現在取り出している表 X の(1)の行は WHERE 以降の条件が「真」と判断され、主問合せの結果として抽出されます。

続いて、表 X の(2)の行が取り出されます。そして、副問合せ中の下線の条件判定によって、表 X の(2)の行の列 P の値（20）と同じ値の列 P をもつ行のみが、表 Y から選別されます。その結果、図 6 の網掛け部分の行のみが表 Y から選別されることになります。

表 Y

P	Q	
10	400	(1)
20	40	(2)
10	50	(3)
20	80	(4)

図 6

図 6 の網掛け部分の行の中から，WHERE 句の残りの条件（AND 表 Y.Q = 50）によって，列 Q の値が 50 のみのものだけが残されます。しかし，図 5 の(2)及び(4)の行は当該条件を満たさないため残されません。すなわち，図 3 の副問合せでは，結果の行が 1 行も返ってこないことになります。

よって，図 3 の主問合せでは EXISTS 句での判定が「偽」となるため，現在取り出している表 X の(2)の行は WHERE 以降の条件が「偽」と判断され，主問合せの結果として抽出されなくなります。

さらに続いて，表 X の(3)の行が取り出されます。そして，副問合せ中の下線の条件判定によって，表 X の(3)の行の列 P の値（25）と同じ値の列 P をもつ行のみが，表 Y から選別されます。しかし，表 Y には列 P の値が 25 の行は存在しないため，この条件判定の時点で表 Y からは行は選別されなくなります。よって，WHERE 句の残りの条件（AND 表 Y.Q = 50）に当てはまる行は存在しないことになり，副問合せでは結果の行が 1 行も返ってこないことになります。

したがって，図 3 の主問合せでは EXISTS 句での判定が「偽」となるため，現在取り出している表 X の(3)の行は WHERE 以降の条件が「偽」と判断され，主問合せの結果として抽出されなくなります。

以上から，図 3 の SQL 文では，図 4 の表 X の(1)の行のみが抽出され，出力されます。したがって，この SQL 文の主問合せにより抽出される行は，下記の条件を満たすもののみとなります。

- (1) 表 X の行の列 P の値と，同じ値の列 P をもつ行が表 Y にも存在している
- (2) かつ，その表 Y の行の列 Q の値が 50 である

設問 1, 2 の SQL 文は，上記の EXISTS 句の働きを理解していれば，読解が容易になります。

【設問 1】

- ・ 空欄 a

本間の社員表、社員スキル表の内容を図7に示します。

社員表

社員番号	社員名	所属
0001	鈴木	A1
0002	田中	A2
0003	佐藤	B1
0004	橋本	D3
0005	今井	A1
0006	木村	B1

社員スキル表

社員番号	スキルコード	登録日
0001	FE	19991201
0001	DB	20010701
0002	SAD	19980701
0002	FE	19990701
0002	SW	20000701
0005	SAD	19991201

図 7

```
SELECT 社員番号 FROM 社員スキル表
WHERE EXISTS (SELECT * FROM 社員スキル表
               WHERE スキルコード = 'FE')
```

この SELECT 文を実行すると、図7の内容から社員スキル表にはスキルコードが“FE”である行が2行存在しているとわかるため、副問合せの結果として2つの行が返ってきます。すなわち、副問合せの結果が存在することになります。よって、EXISTS 句の評価を行った結果は必ず真となります。よって、主問合せのどの行についても WHERE 以下の条件が真となるため、社員スキル表のすべての行が検索条件に一致するとみなされ、社員スキル表のすべての行の社員番号が出力されます。

以上から、社員スキル表のすべての行の社員番号が出力されるため、空欄a= “6 (キ)” となります。

・空欄 b

```
SELECT 社員名 FROM 社員表 A
WHERE EXISTS (SELECT * FROM 社員スキル表 B
               WHERE スキルコード = 'FE'
               AND A.社員番号 = B.社員番号)
```

この SELECT 文では、副問合せ中の FROM 句の後に存在していない表（社員表 A）が副問合せの条件式にて参照されているため、相関副問合せが用いられていると理解できます。

主問合せにて参照される社員表の各行について、相関副問合せによって評価した結果を

以下に示します。

(1) 社員番号=0001 の行

この行と同じ値の社員番号をもつ行は、社員スキル表の 1 行目と 2 行目に存在します。これらの行のうちの 1 行目のスキルコードの値が“FE”のため、副問合せにおいて 1 つの行が社員スキル表から取り出されます。よって、社員表の社員番号が 0001 の行について EXISTS 句の評価を行った結果は真となります。したがって、この行は検索条件に合致していることになり、抽出されます。

(2) 社員番号=0002 の行

この行と同じ値の社員番号をもつ行は、社員スキル表の 3 行目から 5 行目までに存在します。これらの行のうちの 4 行目のスキルコードの値が“FE”のため、副問合せにおいて 1 つの行が社員スキル表から取り出されます。よって、社員表の社員番号が 0002 の行について EXISTS 句の評価を行った結果は真となります。したがって、この行は検索条件に合致していることになり、抽出されます。

(3) 社員番号=0003 の行

この行と同じ値の社員番号をもつ行は、社員スキル表には存在しません。したがって、副問合せにおいて行が社員スキル表から取り出されないため、社員表の社員番号が 0003 の行について EXISTS 句の評価を行った結果は偽となります。よって、この行は検索条件に合致していないため、抽出されません。

(4) 社員番号=0004 の行

この行と同じ値の社員番号をもつ行は、(3)と同様に社員スキル表には存在しません。よって、この行は検索条件に合致していないため、抽出されません。

(5) 社員番号=0005 の行

この行と同じ値の社員番号をもつ行は、社員スキル表の 6 行目に存在します。この行のスキルコードの値は“FE”ではないため、副問合せにおいて行が社員スキル表から取り出されないこととなります。よって、社員表の社員番号が 0005 の行について EXISTS 句の評価を行った結果は偽となります。したがって、この行は検索条件に合致していないため、抽出されません。

(6) 社員番号=0006 の行

この行と同じ値の社員番号をもつ行は、(3)と同様に社員スキル表には存在しません。よって、この行は検索条件に合致していないため、抽出されません。

以上から、主問合せで抽出されるのは、社員番号が 0001 と 0002 の 2 行のみとなります。よって、選択される社員名は“2 (ウ)”となります。

【設問 2】

設問 2 の SQL 文を示します。

```

SELECT DISTINCT 社員番号 FROM 社員スキル表 B1
WHERE EXISTS ( SELECT * FROM 社員スキル表 B2
                WHERE B1.社員番号 = B2.社員番号
                AND B1.スキルコード <> B2.スキルコード )

```

この SQL 文では、主問合せにて社員スキル表を B1 として扱い、副問合せでも社員スキル表を B2 として扱っています。このように、同じ表に別名をつけて扱っているときは、主問合せの社員スキル表と副問合せの社員スキル表は別のものとみなされます。よって、ここでも相関副問合せが行われることになります。

社員スキル表 B1

社員番号	スキルコード	登録日
0001	FE	19991201
0001	DB	20010701
0002	SAD	19980701
0002	FE	19990701
0002	SW	20000701
0005	SAD	19991201

社員スキル表 B2

社員番号	スキルコード	登録日
0001	FE	19991201
0001	DB	20010701
0002	SAD	19980701
0002	FE	19990701
0002	SW	20000701
0005	SAD	19991201

図 8

図 8 の表 B1 と表 B2 は、前述の SELECT 文では「構成や内容は同じであるが、異なる表である」ものとして扱われます。

主問合せにて参照される社員スキル表 B1 の各行について、相関副問合せによって評価した結果を以下に示します。

(1) 社員番号=0001, スキルコード=“FE” の行

この行と同じ値の社員番号をもつ行は、社員スキル表 B2 の 1 行目と 2 行目に存在します。これらのうちの 1 行目のスキルコードの値は“FE”のため、副問合せの“B1.スキルコード <> B2.スキルコード”という条件には合致しません。2 行目のスキルコードの値は“DB”のため、“B1.スキルコード <> B2.スキルコード”という条件に合致します。よって、社員スキル表の 2 行目が副問合せの条件をすべて満たすため、副問合せでは 1 つの行が社員スキル表 B2 から取り出されます。よって、社員スキル表 B1 の社員番号が 0001, スキルコードが“FE”の行について EXISTS 句の評価を行った結果は真となります。したがって、この行は検索条件に合致していることになり、抽出されます。

(2) 社員番号=0001, スキルコード=“DB” の行

この行と同じ値の社員番号をもつ行は、社員スキル表 B2 の 1 行目と 2 行目に存在します。これらのうちの 1 行目のスキルコードの値は“FE”のため、副問合せの“B1. スキルコード <> B2. スキルコード”という条件に合致します。2 行目のスキルコードの値は“DB”のため、“B1. スキルコード <> B2. スキルコード”という条件に合致しません。よって、社員スキル表の 1 行目が副問合せの条件をすべて満たすため、副問合せでは 1 つの行が社員スキル表 B2 から取り出されます。よって、社員スキル表 B1 の社員番号が 0001、スキルコードが“DB”の行について EXISTS 句の評価を行った結果は真となります。したがって、この行は検索条件に合致していることになり、抽出されます。

(3) 社員番号=0002, スキルコード=“SAD”の行

この行と同じ値の社員番号をもつ行は、社員スキル表 B2 の 3 行目から 5 行目に存在します。これらのうち、社員スキル表 B1 の行のスキルコード=“SAD”と異なるスキルコードをもつ行は 4 行目と 5 行目になります。よって、社員スキル表の 4, 5 行目が副問合せの条件をすべて満たすため、副問合せでは 2 つの行が社員スキル表 B2 から取り出されます。よって、社員番号=0002, スキルコード=“SAD”の行について EXISTS 句の評価を行った結果は真となります。したがって、この行は検索条件に合致していることになり、抽出されます。

(4) 社員番号=0002, スキルコード=“FE”の行

この行と同じ値の社員番号をもつ行は、社員スキル表 B2 の 3 行目から 5 行目に存在します。これらのうち、社員スキル表 B1 の行のスキルコード=“FE”と異なるスキルコードをもつ行は 3 行目と 5 行目になります。よって、社員スキル表の 3, 5 行目が副問合せの条件をすべて満たすため、副問合せでは 2 つの行が社員スキル表 B2 から取り出されます。よって、社員番号=0002, スキルコード=“FE”の行について EXISTS 句の評価を行った結果は真となります。したがって、この行は検索条件に合致していることになり、抽出されます。

(5) 社員番号=0002, スキルコード=“SW”の行

この行と同じ値の社員番号をもつ行は、社員スキル表 B2 の 3 行目から 5 行目に存在します。これらのうち、社員スキル表 B1 の行のスキルコード=“SW”と異なるスキルコードをもつ行は 3 行目と 4 行目になります。よって、社員スキル表の 3, 4 行目が副問合せの条件をすべて満たすため、副問合せでは 2 つの行が社員スキル表 B2 から取り出されます。よって、社員番号=0002, スキルコード=“SW”の行について EXISTS 句の評価を行った結果は真となります。したがって、この行は検索条件に合致していることになり、抽出されます。

(6) 社員番号=0005, スキルコード=“SAD”の行

この行と同じ値の社員番号をもつ行は、社員スキル表 B2 の 6 行目のみに存在します。社員スキル表 B2 の 6 行目は、社員スキル表 B1 の行のスキルコード=“SAD”と異なるスキルコードをもたないため、副問合せの条件をすべて満たす行は社員スキル表 B2 には存在しないとわかります。よって、社員番号=0005, スキルコード=“SAD”の行について EXISTS 句の評価を行った結果は偽となり、抽出されません。

以上から、社員スキル表 B1 の 1 行目から 5 行目までが主問合せによって抽出されるため、{0001, 0001, 0002, 0002, 0002} という社員番号が出力対象となります。しかし、前述の SELECT 文では DISTINCT 句が用いられているため、重複する値は除かれて出力されます。よって、{0001, 0002} という 2 行分の値が出力されることになります。

社員スキル表を参照すると、前述の SELECT 文で出力された社員番号 (0001 及び 0002) の社員は、複数のスキルをもっているとわかります。社員番号が 0005 の社員は一つしかスキルをもっていないため、出力されなかったと推測できます。

以上から、“スキルを複数もつ。(エ)” が正解となります。

ア … スキルを少なくとも一つはもっている社員が要求されている場合、社員番号が 0005 の社員も出力されるべきであるため、誤りです。

イ … スキルを一つだけもっている社員が要求されている場合、社員番号が 0005 の社員だけが出力されるべきであるため、誤りです。

ウ … スキルを一つももたない社員が要求されている場合、社員番号が 0001, 0002 の社員が出力されてはならないことになります。よって誤りです。

【設問 3】

設問 2 の SELECT 文では、社員スキル表のみを用いて処理を行っているため、社員スキル表に存在している列 (社員番号やスキルコードなど) だけを取り出すことができます。社員スキル表に存在していない列 (氏名など) は取り出せないため、出力できないことになります。

社員名を出力できるようにするためには、社員表と社員スキル表を結合して、設問 2 の 関連副問合せにて得られた社員番号 (スキルを複数もっている社員の社員番号) と同じ社員番号の行のみを社員表から取り出し、その行の社員名を出力すればよいとわかります。よって、主問合せの条件に “A. 社員番号 = B1. 社員番号” を加え、社員表 A と社員スキル表 B1 を結合することになります。

以上から、“A. 社員番号 = B1. 社員番号 (ア)” が適切です。

イ … “BETWEEN A. 社員番号 AND B1. 社員番号” のような条件式を用いると、例えば社員スキル表 B1 の社員番号 0006 の行が副問合せの条件に合致した場合、社員表 A の社員番号 0001 から、社員スキル表 B1 の社員番号 0006 までの社員番号がすべて検索条件に一致してしまい、すべての社員が抽出されてしまう可能性があるため、誤りです。

ウ … LIKE 句は、“社員表. 社員名 LIKE %木%” のように、指定した字句を含む文字列を検索する際に用いられるものです。“LIKE A. 社員番号 = B1. 社員番号” のように LIKE 句を用いることは、SQL の文法上の誤りとなります。

エ … “OR A. 社員番号 = B1. 社員番号” という OR 演算子及び条件式を用いると、社員表 A の社員番号と社員スキル表 B1 の社員番号が同じ行は、すべて検索条件に一

致してしまい、スキルを複数もっていない社員も抽出されてしまうため、誤りです。

問 5

解答：

設問 1

イ，ウ，オ

設問 2

a：エ b：カ c：イ

解説：

輸入販売企業の輸入割当てシステム（以下、システムという）の設計に関する問題です。
〔システムの概要説明〕の(5)に記述されている，輸入拠点の割当ての優先順位について適切に理解できているかが，本問を解答するためのポイントとなります。

【システムの概要など（設問 1）】

本問のシステムの概要及び発注拠点を割り当てる手順について，例を用いて説明します。
また，その過程において設問 1 の解答についても説明します。

・システムの概要について

(1) ここでは，商品コードが“10001”である商品を，希望納期を 2003 年 4 月 1 日として 200 個受注したことを想定します。すなわち，本問のシステムは，販売部門から以下の受注情報を受け取ることになります。

受注情報

商品コード	受注数	希望納期		
		年	月	日
10001	200	2003	4	1

現在の日時を 2003 年 3 月 22 日とすると，配送日数の余裕は 10 日間となります。

(2) システムは，受注情報の商品コードを基に，その商品を扱っている輸入拠点に在庫数を問い合わせます。

(3) 輸入拠点は，問合せを受けた商品の在庫情報を返信します。海外発注部門は，返信された在庫情報をいったん在庫情報ファイルへ書き出します。

ここでは，商品コードが“10001”である商品の在庫情報を五つの拠点が返信し，以下に示す在庫情報ファイルに当該在庫情報が記録されたと想定します。

在庫情報ファイル

拠点コード	商品コード	在庫数
01	10001	100
03	10001	50
04	10001	0
14	10001	200
18	10001	0

(4) 輸入拠点から販売部門への商品配送手段には、船便と航空便の 2 種類があります。輸入拠点によって配送日数が異なり、同じ輸入拠点でも船便と航空便とで配送日数が異なります。

ここでは、上の在庫情報ファイル中のレコードの各輸入拠点について、拠点マスタファイルに記録されている各拠点の航空便での配送日数と船便での配送日数の例を付記します。本問の図の手順では、拠点マスタファイルの航空便での配送日数などの情報を用いて、希望納期に間に合うかどうかを調べています。

(在庫情報ファイル)

(拠点マスタファイル)

拠点コード	商品コード	在庫数	航空便での 配送日数	船便での 配送日数
01	10001	100	6	13
03	10001	50	5	9
04	10001	0	5	10
14	10001	200	12	21
18	10001	0	7	11

図 1

(5) 本問の図に示された手順を用いて、在庫情報と、航空便での配送日数及び船便での配送日数を基に、輸入の拒否を判断します。そして、各輸入拠点からの輸入数量を決定します。

(6) 輸入拠点と発注数及び配送手段が決定したら、該当輸入拠点へ発注情報を送信します。発注情報の様式は、設問 1 の解答を得る上で重要となります。

- ・発注拠点を割り当てる手順について

図 1 の在庫情報ファイルが与えられたと仮定して、本問の図の手順をトレースします。

- (1) 商品コードなどの受注情報を記憶

この処理では、受注情報中の商品コードなどの情報がシステムに記憶されます。ここから、受注した商品の商品コードや受注数などの受注情報に関する値はシステムに記憶されているため、一時ファイル A などにあらかじめ記録する必要はないと推測できます。

(2) 在庫情報ファイルからレコードを 1 件読み込む

図 1 の先頭のレコード（在庫数=100，航空便での配送日数=6，船便での配送日数=13）が読み込まれます。このレコードが、在庫情報ファイル読み込みの繰返し処理の最初の処理対象となります。

(3) 在庫情報ファイル読み込みの繰返し処理

① 図 1 の先頭のレコード

「船便及び航空便での納期の計算」処理では、図 1 の先頭のレコードの航空便での配送日数や船便での配送日数を参照し、納期に間に合うかどうか計算されます。配送日数の余裕は 10 日間のため、船便では間に合いませんが、航空便では間に合うとわかります。また、在庫数が 0 ではないため、在庫があると判明します。

以上から、図 1 の先頭のレコードの輸入拠点は希望納期に間に合います。そのため、「情報を書き出す」の処理によって、このレコードの情報が一時ファイル A に書き出されます。

ここで、一時ファイル A に最小限必要なデータ項目について考えます。本問の図から、一時ファイル A は「整列」処理によって整列され、一時ファイル B になっています。整列を行うだけならば、レコードの様式は変化しないと考えられるため、一時ファイル B の様式は一時ファイル A の様式と同じと推測できます。

本問の図の「一時ファイル B 読み込み」の繰返し処理中にて、受注数と在庫数のうちの最小値を発注数に代入する処理を実行しています。この点から、一時ファイル B すなわち一時ファイル A には、各輸入拠点の在庫数が必要となると推定できます。

また、〔システムの概要説明〕の(6)の発注情報の様式を参照すると、拠点コード、商品コード、発注数、配送手段の四つの項目があるとわかります。これらのうち、商品コードはすでにシステムに記憶されているため、一時ファイル A または一時ファイル B にあらかじめ記録する必要はないと判明しています。また、発注数は「一時ファイル B 読み込み」の繰返し処理において計算するものです。そのため、一時ファイル B のレコード中に前もって発注数をもたせておく必要はありません。よって、一時ファイル A に発注数をもたせる必要はありません。

発注情報の様式の拠点コードは、受注情報には存在しておらず、在庫情報ファイルにのみ存在しているものです。在庫情報ファイルから拠点コードを取り出して一時ファイル A に出力し、一時ファイル A を整列して作成した一時ファイル B から拠点コードを取り出すことで、発注情報に拠点コードを出力することができます。よって、拠点コードが一時ファイル A に必要となります。

また、〔システムの概要説明〕の(5)から、航空便よりも船便で間に合う輸入拠点を優先して、輸入拠点の割当てを行うと理解できます。よって、一時ファイル A に配送手段（船便

か航空便かという区分)の項目を加えておき、「整列」処理においては、配送手段が「船便」であるレコードが、「航空便」であるレコードよりも前に位置するように一時ファイルAを並べ替えることで、配送手段が「船便」であるレコードを一時ファイルBから先に取り出し、そのレコードの輸入拠点に優先して発注を割り当てることができます。したがって、配送手段が一時ファイルAに必要となります。

以上から、設問 1 の解答は“拠点コード (イ), 在庫数 (ウ), 配送手段 (オ)”となります。よって、図 1 の先頭のレコードから、拠点コード (01), 在庫数 (100) という値が取り出されます。また、船便では間に合わず、航空便では間に合うため、配送手段の値は「航空便」となります。したがって、一時ファイルAに以下のレコードが追加されます。

一時ファイル A

拠点コード	在庫数	配送手段
01	100	航空便

② 図 1 の 2 番目のレコード

図 1 の 2 番目のレコードの拠点について、在庫があるかどうか、希望納期に間に合うかを調べます。配送日数の余裕は 10 日間のため、船便で間に合います。また、在庫数が 0 ではないため、在庫があると判明します。

よって、図 1 の 2 番目のレコードから、拠点コード (03), 在庫数 (50) という値が取り出されます。また、船便で間に合うため、配送手段の値は「船便」となります。したがって、一時ファイル A に以下のレコードが追加されます。

一時ファイル A

拠点コード	在庫数	配送手段
01	100	航空便
03	50	船便

③ 図 1 の 3 番目のレコード

図 1 の 3 番目のレコードの拠点について、在庫があるかどうか、希望納期に間に合うかを調べます。配送日数の余裕は 10 日間のため、船便で間に合います。しかし、在庫数が 0 のため、在庫がないと判明します。よって、このレコードの情報は一時ファイル A に書き出されません。

④ 図 1 の 4 番目のレコード

図 1 の 4 番目のレコードの拠点について、在庫があるかどうか、希望納期に間に合うかを調べます。配送日数の余裕は 10 日間のため、船便、航空便ともに間に合いません。よって、このレコードの情報は一時ファイル A に書き出されません。

⑤ 図1の5番目のレコード

図1の5番目のレコードの拠点について、在庫があるかどうか、希望納期に間に合うかを調べます。配送日数の余裕は10日間のため、船便では間に合わず、航空便では間に合います。しかし、在庫数が0のため、在庫がないと判明します。よって、このレコードの情報は一時ファイルAに書き出されません。

図1の5番目のレコードの処理を終えた時点で、在庫情報ファイルのすべてのレコードを読み込み、ファイルの終わりに到達します。したがって、在庫情報ファイル読み込みの繰り返し処理が終了します。

(4) 整列

一時ファイルAを整列します。〔システムの概要説明〕の(5)の①～③の記述から、以下の優先順位で輸入拠点の割当てを行うと推定できます。

- ① 航空便よりも船便で間に合う輸入拠点を優先する。
- ② 該当商品の在庫数が多い輸入拠点を優先する。
- ③ 拠点コードが小さい輸入拠点を優先する。

したがって、「整列」処理では、整列の第1キーとして配送手段が選択されます。配送手段が「船便」であるレコードが「航空便」であるレコードよりも前に位置するように、レコードが並べ替えられます。

整列済みの一時ファイルA

拠点コード	在庫数	配送手段
03	50	船便
01	100	航空便

なお、配送手段が同じ値のレコードが複数ある場合は、それらのレコードは②に示した優先順位（在庫数が多い輸入拠点を優先する）で整列されるため、整列の第2キーは在庫数（降順）となります。さらに、配送手段が同じ値であり在庫数も同じ値であるレコードが複数ある場合は、それらのレコードは③で示した優先順位（拠点コードが小さい値の輸入拠点を優先する）で整列されるため、整列の第3キーは拠点コード（昇順）となります。

以上の整列キーで一時ファイルAの整列が行われ、その結果が一時ファイルBに出力されます。

一時ファイルB

拠点コード	在庫数	配送手段
03	50	船便
01	100	航空便

図2

(5) 一時ファイル B からレコードを 1 件読み込む

図 2 の先頭のレコード（拠点コード=03，在庫数=50，配送手段=船便）が読み込まれます。このレコードが，一時ファイル B 読み込みの繰返し処理の最初の処理対象となります。

(6) 一時ファイル B 読み込みの繰返し処理

本解説の例では，商品コードが“10001”である商品を，希望納期を 2003 年 4 月 1 日として 200 個受注しているため，受注数は 200 となります。受注数（200）と，図 2 の先頭のレコードの在庫数（50）のうち，最小のものは図 2 の先頭のレコードの在庫数となります。よって，発注数には 50 が代入されます。拠点コードが 03 の輸入拠点は，商品コードが“10001”である商品の在庫を 50 個しか保持していないため，50 個分だけの発注を行います。

「発注情報に追加」の処理では，拠点コード=03，商品コード=10001，発注数=50，配送手段=船便という情報が発注情報に追加されます。その後，受注数（200）から発注数（50）を減算した値（150）が，受注数に代入されます。200 個の受注のうちの 50 個分を，拠点コードが 03 の輸入拠点の在庫から賄えたため，残りの受注数は 150 個となります。

続いて，図 2 の 2 番目のレコード（拠点コード=01，在庫数=100，配送手段=航空便）が読み込まれます。受注数（150）と，図 2 の 2 番目のレコードの在庫数（100）のうち，最小のものは図 2 の 2 番目のレコードの在庫数となります。よって，発注数には 100 が代入されます。

「発注情報に追加」の処理では，拠点コード=01，商品コード=10001，発注数=100，配送手段=航空便という情報が発注情報に追加されます。その後，受注数（150）から発注数（100）を減算した値（50）が，受注数に代入されます。残り 150 個の受注のうちの 100 個分を，拠点コードが 01 の輸入拠点の在庫から賄えたため，残りの受注数は 50 個となります。

この時点で，一時ファイル B のすべてのレコードが読み込まれたため，ファイルの終わりに到達します。したがって，一時ファイル B 読み込みの繰返し処理が終了します。

一時ファイル B 読み込みの繰返し処理を抜けた時点で受注数の値が 0 になっていないため，商品の在庫があり，かつ希望納期に間に合うすべての輸入拠点から在庫を割り当てても，受注数に満たなかったことになります。このときは，受注数分の割当てができなかったため，発注情報を破棄してエラーメッセージを表示します。

【設問 2】

本問の表のパターン 5～7 について，〔システムの概要説明〕の(5)の①～③に示された優先順位に従って，輸入拠点の割当てを行うことを考えます。

- ・パターン 5（空欄 a）

拠点コード	在庫数	航空便での 配送日数	船便での 配送日数
06	70	7	20
07	70	12	30
08	(空欄 a)	8	21

パターン 5 の内容は上に示す通りとなります。受注情報の受注数は 200、配送日数の余裕は 20 日間となります。本問の表のパターン 5 の予想される結果から、エラーメッセージは表示されていない（すなわち、必要な輸入数量を確保できている）と理解できます。よって、このパターンの各輸入拠点の在庫数の合計は受注数以上となっていると推測できます。したがって、拠点コード 08 の輸入拠点の在庫数を x とおくと、

$$\begin{aligned}
 &70 + 70 + x \geq 200 \\
 \rightarrow &x \geq 200 - 70 - 70 \\
 \rightarrow &x \geq 60
 \end{aligned}$$

となります。すなわち、拠点コード 08 の輸入拠点の在庫は少なくとも 60 以上となります。この時点で、空欄 a の解答として適切なものはエ～キであり、60 未満のア～ウは不適切となります。

ただし、空欄 a の値を 100（オ）またはそれ以上の値とすると誤りとなります。仮に拠点コード 08 の輸入拠点の在庫数を 100 とおくと、この輸入拠点の在庫数は、拠点コード 07 の輸入拠点よりも多くなります。〔システムの概要説明〕の(5)の②の記述から、該当商品の在庫数が多い輸入拠点を優先しているため、拠点コード 07 の輸入拠点よりも、拠点コード 08 の輸入拠点の方が優先して割り当てられます。その結果、以下のようになります。

(1) 船便で間に合う拠点コード 06 の輸入拠点が最も優先されるため、この拠点から最初に割当てが行われる。受注数 (200) > この拠点の在庫数 (70) となるため、この拠点から船便で 70 個割り当てられ、受注数は $200 - 70 = 130$ となる。

(2) 該当商品の在庫数が多い、拠点コード 08 の輸入拠点が優先されるため、この拠点が 2 番目に割り当てられる。受注数 (130) > この拠点の在庫数 (100) となるため、この拠点から航空便で 100 個割り当てられ、受注数は $130 - 100 = 30$ となる。

(3) 最後に、拠点コード 07 の輸入拠点から割り当てられる。受注数 (30) < この拠点の在庫数 (70) となるため、この拠点から航空便で 30 個割り当てられ、受注数は $30 - 30 = 0$ となる。

以上から、拠点コード 08 の輸入拠点の在庫数を 100 とおくと、拠点コード 07 の輸入拠点には航空便で 30 個だけ発注が行われることとなります。しかし、パターン 5 の予想される結果では、拠点コード 07 の輸入拠点には航空便で 70 個発注されるため、矛盾します。したがって、拠点コード 08 の輸入拠点の在庫数は、拠点コード 07 の輸入拠点の在庫数よ

りも多い値になってはならないと判明します。

以上から、空欄aに適切な数値は70（エ）のみとなります。

・パターン 6（空欄 b）

拠点コード	在庫数	航空便での 配送日数	船便での 配送日数
09	150	2	21
10	(空欄 b)	3	22
11	120	4	23

パターン 6 の内容は上に示す通りとなります。受注情報の受注数は 200、配送日数の余裕は 20 日間となります。パターン 6 の予想される結果から、エラーメッセージは表示されないと理解できます。よって、このパターンの各輸入拠点の在庫数の合計は受注数以上となっており、必要な輸入数量を確保できています。

パターン 6 のすべての輸入拠点の航空便での配送日数は 20 未満、船便での配送日数は 21 以上となっているため、どの輸入拠点も船便では間に合わず、航空便では間に合うこととなります。よって、〔システムの概要説明〕の(5)の①～③のうち、①の「航空便よりも船便で間に合う輸入拠点を優先」という優先順位は特に考慮されず、②の「該当商品の在庫数が多い輸入拠点を優先」という優先順位が考慮されます。

空欄 b の値を 50（ウ）などとすると誤りとなります。仮に拠点コード 10 の輸入拠点の在庫数を 50 とすると、拠点コード 10 の輸入拠点よりも、拠点コード 11 の輸入拠点の在庫数の方が多くなるため、拠点コード 11 の輸入拠点が優先されてしまいます。その結果、拠点コード 09 の輸入拠点から 150 個、拠点コード 11 の輸入拠点から 50 個の計 200 個の発注が行われることとなります。これはパターン 6 の予想される結果と矛盾します。よって、拠点コード 10 の輸入拠点の在庫数は、拠点コード 11 の輸入拠点の在庫数（120）以上でなければならないとわかります。

ただし、拠点コード 10 の輸入拠点の在庫数を 160（キ）とすると誤りとなります。仮に拠点コード 10 の輸入拠点の在庫数を 160 とすると、拠点コード 09 の輸入拠点よりも、拠点コード 10 の輸入拠点の在庫数の方が多くなるため、拠点コード 10 の輸入拠点が優先されてしまいます。その結果、拠点コード 10 の輸入拠点から 160 個、拠点コード 09 の輸入拠点から 40 個の計 200 個の発注が行われることとなります。これはパターン 6 の予想される結果と矛盾します。

以上から、拠点コード 10 の輸入拠点の在庫数は、拠点コード 09 の輸入拠点の在庫数(150)未満でなければならないとわかります。これらの条件を満たしているのは、カ（120）のみとなります。

・パターン 7（空欄 c）

拠点コード	在庫数	航空便での 配送日数	船便での 配送日数
12	(空欄 c)	20	40
13	160	10	20
14	30	15	25

パターン 7 の内容は上に示す通りとなります。受注情報の受注数は 200、配送日数の余裕は 20 日間となります。パターン 7 の予想される結果から、エラーメッセージは表示されないと理解できます。よって、このパターンの各輸入拠点の在庫数の合計は受注数以上となっており、必要な輸入数量を確保できています。

パターン 7 の輸入拠点のうち、拠点コード 13 の拠点のみ、船便で間に合うとわかります。よって、この拠点から最優先で割当てが行われます。その結果、受注数は $200 - 160 = 40$ となります。

パターン 7 の予想される結果から、残りの 40 個の受注数のうち、拠点コード 12 の輸入拠点に航空便で 30 個発注を行っているといわれます。〔システムの概要説明〕の(5)の②の「該当商品の在庫数が多い輸入拠点を優先」という優先順位から、空欄 c には少なくとも 30 以上の値が入ると推定できます。仮に、空欄 c の値を 20 (ア) とすると、拠点コード 14 の輸入拠点の在庫数の方が多くなり、拠点コード 14 の輸入拠点の方が優先されてしまいます。その結果、拠点コード 14 の輸入拠点に 30 個発注が行われます。しかし、パターン 7 の予想される結果から、この輸入拠点には 10 個しか発注を行わないため、矛盾します。したがって、空欄 c の値は 30 以上となります。

ただし、空欄 c に 50 (ウ) などの値をあてはめると、受注数の残りの 40 個分を拠点コード 12 の輸入拠点の在庫だけで賄ってしまうことになり、拠点コード 14 の輸入拠点に発注をかける必要がなくなります。しかし、これもパターン 7 の予想される結果に矛盾します。したがって、空欄 c には 30 (イ) のみが適切となります。

問 6

解答：

設問

a：オ b：イ c：キ

解説：

PERT による日程計画に関する問題です。

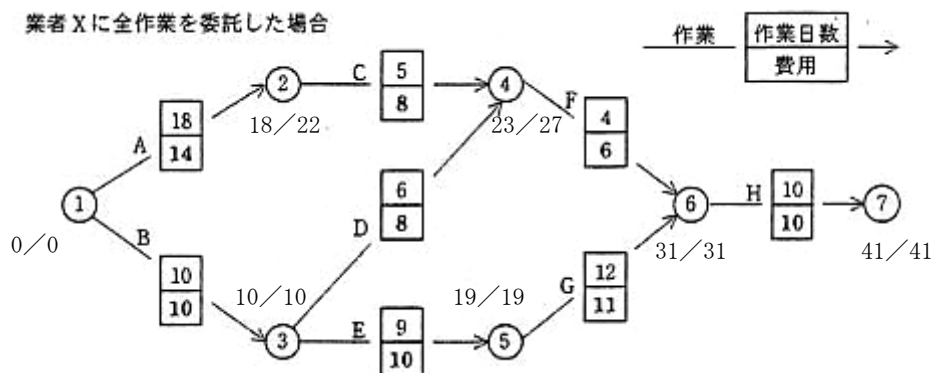
【設問】

・空欄 a

表 2 の業者 X の全ての作業の費用を合計すると、 $14+10+8+8+10+6+11+10=77$ となります。よって、プロジェクトの完成に必要な費用は 77 (オ) です。

・空欄 b

図の、業者 X に全作業を委託した場合のアローダイアグラムに、最早結合点時刻及び最遅結合点時刻を付記したものを示します。



凡例 x/y x =最早結合点時刻 y =最遅結合点時刻 以下同じ

作業 C ($18+5=23$ 日目に完了) と作業 D ($10+6=16$ 日目に完了) の両方が完了してから、④から出ている作業 F を開始できます。作業 C の方が遅く完了するので、作業 C が完了する 23 日目から作業 F を開始できます。

作業 F ($23+4=27$ 日目に完了) と作業 G ($10+9+12=31$ 日目に完了) の両方が完了してから、⑥から出ている作業 H を開始できます。作業 G の方が遅く完了するので、作業 G が完了する 31 日目から作業 H を開始できます。

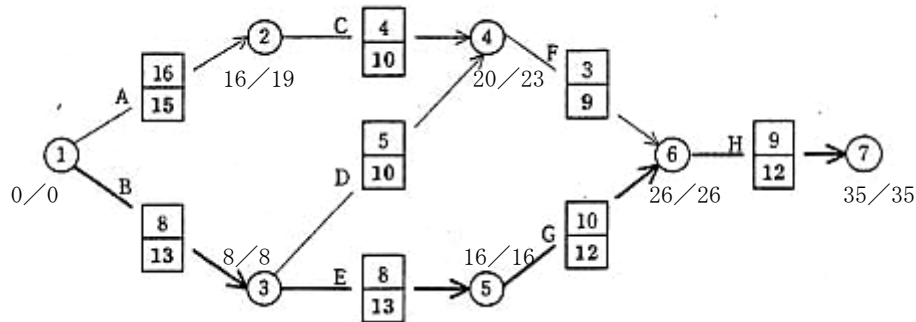
以上から、完成までの所要日数は 41 日 (イ) です。

・空欄 c

図の、業者 Y に全作業を委託した場合のアローダイアグラムに、最早結合点時刻及び最

遅結合点時刻を付記したものを示します。

業者 Y に全作業を委託した場合



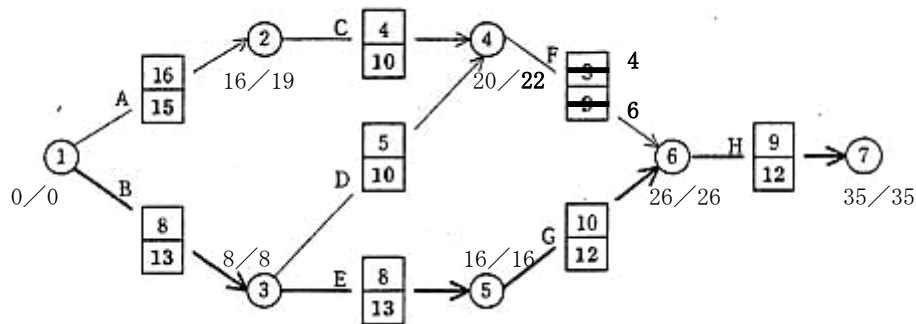
作業 C (16+4=20 日目に完了) と作業 D (8+5=13 日目に完了) の両方が完了してから、④から出ている作業 F を開始できます。作業 C の方が遅く完了するので、作業 C が完了する 20 日目から作業 F を開始できます。

作業 F (20+3=23 日目に完了) と作業 G (8+8+10=26 日目に完了) の両方が完了してから、⑥から出ている作業 H を開始できます。作業 G の方が遅く完了するので、作業 G が完了する 26 日目から作業 H を開始できます。完成までの所要日数は 35 日です。クリティカルパスは BEGH です。

クリティカルパス上のいずれかの作業の所要日数を長くすると、完成までの所要日数が長くなってしまいますので、作業 B, E, G, H のいずれかを業者 X に委託するのは不適切です。クリティカルパス上でない作業を業者 X に委託することで、所要日数は 35 日のままで費用をより少なくできます。ただし、クリティカルパス上でない作業の所要日数を長くしすぎると、クリティカルパスが変わって完成までの所要日数が長くなることがあるので、注意が必要です。

クリティカルパス上でない作業 (作業 A, C, D, F) のうち、業者 Y の費用に比べて業者 X の費用が最も少ない作業から順に、業者 X に委託していくことで、費用をより少なくできます。表 2 から、業者 Y の費用に比べて業者 X の費用が最も少ないのは作業 F です。

- ・作業 F を業者 X に委託した結果

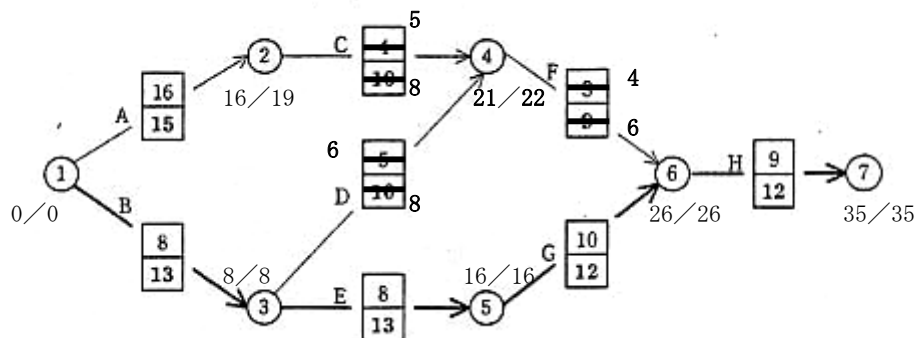


(太字は委託後の作業日数及び費用、及び委託後に値が変化した最早結合点時刻と最遅結合点時刻。以下同じ)

作業 F が完了するのは 24 日目になり、作業 G の完了 (26 日目) より早いので、作業 G が完了しないと作業 H を開始できないことには変わりはありません。クリティカルパスは BEGH のままです。

続いて、業者 Y の費用に比べて業者 X の費用が少ない作業 (作業 C, D) の二つを業者 X に委託します。

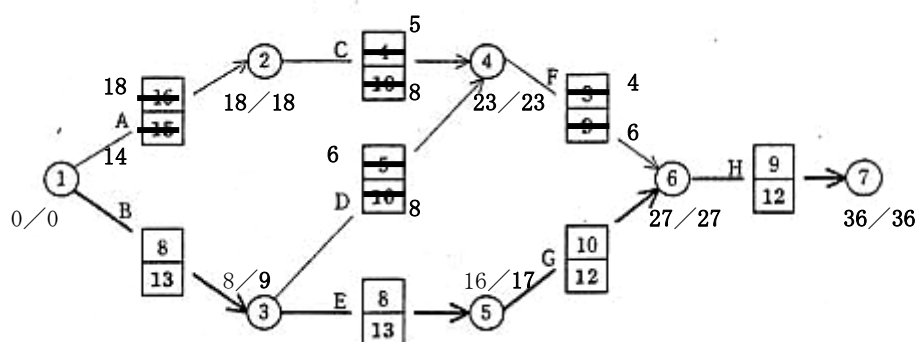
- ・さらに作業 C, D を業者 X に委託した結果



作業 C の完了が 21 日目になり、作業 F の完了が 25 日目になります。作業 F の完了は作業 G の完了より早いので、作業 G が完了しないと作業 H を開始できないことには変わりはありません。クリティカルパスは BEGH のままです。

さらに、作業 A を業者 X に委託します。

- ・さらに作業 A を業者 X に委託した結果



作業 A の完了が 18 日目、作業 C の完了が 23 日目になるため、作業 F の完了が 27 日目になります。作業 F の完了が作業 G の完了よりも遅くなるので、作業 F が完了しないと作業 H を開始できなくなり、クリティカルパスが ACFH に変化します。所要日数が 35 日よりも増えてしまうので、作業 A を業者 X に委託するのは不適切です。

以上から、作業 C, D, F を業者 X に委託するのが適切です。全作業を業者 Y に委託した場合と比較して、作業 C の費用が 2, 作業 D の費用が 2, 作業 F の費用が 3 少なくなるので、費用は $94 - 2 - 2 - 3 = 87$ に変わります。キが正解です。

問 7

解答：

設問 1

a：ウ b：ア

設問 2

エ

設問 3

ア

解説：

定量発注方式の最適な発注数量を求める数式に関する問題です。

【設問 1】

・ 空欄 a

①の式を変形した式を示します。

$$T_c = DC + \frac{Q}{2} CI + BCI + \frac{D}{Q} S$$

上の式のうち、資材の購入費用 DC は、1 回の発注数量とは関係なく一定です。

資材の消費量が年間を通じて一定でなく、急激に資材を消費する時期や資材の消費が少なくなる時期がある場合、急激に資材を消費する時期においては安全在庫数量 B を多めに
して在庫切れを防ぎ、逆に資材の消費が少なくなる時期においては安全在庫数量 B を少な
めにして在庫が過剰にならないようにする必要があります。資材の消費量が年間を通じて
一定であるとする、安全在庫数量 B を 1 回の発注数量とは関係なく一定にすることがで
きます。

したがって、1 回の発注数量の変化によって変わる費用は、次の式の網掛け以外の部分で
す。

$$T_c = \text{DC} + \frac{Q}{2} CI + \text{BCI} + \frac{D}{Q} S$$

上の網掛け以外の部分のうち、発注回数によって変わる費用は $\frac{D}{Q} S$ です。資材の年間
必要数量=D を、1 回の発注数量=Q で割った値が年間の発注回数になります。例えば、資
材の年間必要数量が 10,000 個、1 回の発注数量が 1,000 個のとき、年間の発注回数=10,000
÷1,000=10 回です。1 回の発注数量が 2,000 個に増やすと、年間の発注回数=10,000÷
2,000=5 回となり、年間の発注回数が少なくなります。

$$\frac{D}{Q}$$

Q（発注数量）が大きくなると年間の発注回数の値が小さくなり、S の値は小さくなります。

上の網掛け以外の部分のうち、在庫管理費用の変動部分は $\frac{Q}{2} CI$ です。Q（発注数量）が大きくなるとこの項の値が大きくなるためです。

以上から、空欄aにはウ、空欄bにはアが入ります。

【設問 2】

ある資材の年間必要数量=D が 4,000 個、単価=C が 5,000 円、年間の在庫管理費率=I が 20%=0.2、1 回の発注に必要な費用=S が 20,000 円であるとき、②の式にこれらの値を当てはめると、次のようになります。

$$EOQ = \sqrt{\frac{2DS}{CI}} = \sqrt{\frac{2 \times 4,000 \times 20,000}{5,000 \times 0.2}} = \sqrt{160,000} = 400$$

以上から、EOQは 400 個（エ）です。

【設問 3】

発注数量が 1,000 個未満のとき、1,000～1,999 個のとき、及び 2,000 個のときにおける EOQ を計算します。

- ・発注数量が 1,000 個未満のとき
単価が 5,000 円なので、

$$EOQ = \sqrt{\frac{2DS}{CI}} = \sqrt{\frac{2 \times 4,000 \times 20,000}{5,000 \times 0.2}} = \sqrt{160,000} = 400$$

となります。EOQ が 1,000 個未満なので、この単価で発注できる数量の範囲内です。

Q を 400 個としたときの総費用は、設問 3(2)の記述から、総費用は DC 、 $QCI/2$ 、 DS/Q を求めて合計すればよいので、次のようになります。

$$DC = 4,000 \times 5,000 = 20,000,000$$

$$QCI/2 = 400 \times 5,000 \times 0.2 / 2 = 200,000$$

$$DS/Q = 4,000 \times 20,000 / 400 = 200,000$$

$$\text{合計} = 20,400,000$$

- ・発注数量が 1,000～1,999 個のとき
単価が 4,500 円なので、

$$EOQ = \sqrt{\frac{2DS}{CI}} = \sqrt{\frac{2 \times 4,000 \times 20,000}{4,500 \times 0.2}} \doteq \sqrt{177,777} \doteq 421$$

となります。EOQ が 1,000 個に満たないので、この単価で発注できる数量の範囲外です。このときは、1,000～1,999 個という範囲の最小発注数量＝1,000 個を EOQ として、総費用を計算します。

$$DC = 4,000 \times 5,000 = 20,000,000$$

$$QCI/2 = 1,000 \times 5,000 \times 0.2/2 = 500,000$$

$$DS/Q = 4,000 \times 20,000/1,000 = 80,000$$

$$\text{合計} = 20,580,000$$

・発注数量が 2,000 個以上のとき
単価が 4,000 円なので、

$$EOQ = \sqrt{\frac{2DS}{CI}} = \sqrt{\frac{2 \times 4,000 \times 20,000}{4,000 \times 0.2}} \doteq \sqrt{200,000} \doteq 447$$

となります。EOQ が 2,000 個に満たないので、この単価で発注できる数量の範囲外です。このときは、2,000 個以上という範囲の最小発注数量＝2,000 個を EOQ として、総費用を計算します。

$$DC = 4,000 \times 5,000 = 20,000,000$$

$$QCI/2 = 2,000 \times 5,000 \times 0.2/2 = 1,000,000$$

$$DS/Q = 4,000 \times 20,000/2,000 = 40,000$$

$$\text{合計} = 21,040,000$$

以上から、発注数量が 1,000 個未満のときの総費用が最も安くなるので、このときの EOQ＝400 個（ア）が正解です。

問 8

解答：

設問

a：ウ b：オ c：イ d：オ e：エ

解説：

スタックを用いて、逆ポーランド表記法で表された数式を計算するアルゴリズムの問題です。このアルゴリズムについて考える際には、逆ポーランド表記法について知っておくと便利です。

【逆ポーランド表記法】

逆ポーランド表記法では、掛け算・割り算を足し算・引き算より優先するという、通常の数式の計算順序に従って、演算子とその両側の項や式（オペランド）を後置表記していきます。ただし、式の中に括弧がある場合は、括弧の中の部分をさらに優先して処理します。

例 1（下線部分は、後置表記を施し済みの部分。以下同じ）：

$$X = A + B \times C$$

→ $X = A + \underline{BC} \times$ （掛け算部分を先に後置表記する）

→ $X = \underline{ABC \times} +$ （足し算部分は、後に後置表記する）

例 2：

$$X = (A + B) \times C$$

→ $X = \underline{AB +} \times C$ （括弧があるので、その中の部分を先に処理する）

→ $X = \underline{AB + C} \times$

なお、優先度の同じ演算子が複数存在する数式では、左側の演算子から先に後置表記することになります。

・本問の逆ポーランド表記法の注意点(1)：数値の前の空白文字

例 1, 2 に示した例では、オペランドの値を“A”のようにしています。このように、数値でないものがオペランドになっている場合、オペランド同士が並んで表記されても、区切りがわかりやすいため問題はありません。

しかし、“1”や“2”などの数値をオペランドとする数式を逆ポーランド表記法で表記すると、以下のようになります。

例 3：

中間表記法による数式… $1 + 2 - 3$

逆ポーランド表記法による表記… $12 + 3 -$

例 3 の逆ポーランド表記法による表記では、“12+…”のようになっています。“1”と“2”という二つの異なる数値が後置表記によって並んで表記されるため、“12+…”という表記

の下線部が，“1”と“2”という二つの異なる数値を並べたものか，“12”という一つの数値を表現したものなのかの区別をつけにくくなっています。数式中に出現するオペランドがすべて 1 けたの数値であるという前提などがあれば，上のような状態になっていても区別は可能となります。しかし，本問ではオペランドのけた数などに制限はないため，後置表記によって複数の数値が並んだときに，区別しにくくなります。

この問題に対処するため，本問では「先頭の整数を除き，整数の前には一つの空白文字を書く」という規則を設けています。この規則により，例 3 の数式は以下のようになります（“△”は空白文字を示す。以下同じ）。なお，中間表記法による数式も，同様の規則によって記述しています。

中間表記法による数式…1+△2-△3

逆ポーランド表記法による表記…1△2+△3-

以上から，“1”と“2”という二つの異なる数値が後置表記によって並んで表記される部分では，“1”と“2”が“△”で区切られるようになり，容易に区別できるようになっています。

・本問の逆ポーランド表記法の注意点(2)：“-”の扱い

本問の数式では，減算を表す演算子として“-”という文字が用いられていると同時に，“-100”などの負の数を表す文字としても“-”が用いられています。よって，負の数を含む数式を逆ポーランド表記法で表記すると，以下ようになります。

例 4：

中間表記法による数式…(100-200)*5

逆ポーランド表記法による表記…100200-5*

例 4 の表記では，“-5*”となっており，“-”の文字が減算を表す演算子なのか，負の数（-5）を表す文字なのかの区別がつけにくくなっています。ただし，注意点(1)のように，数値の前に空白文字を書く規則を設けると，区別しやすくなります。

中間表記法による数式…(100-△200)*△5

逆ポーランド表記法による表記…100△200-△5*

この表記では，“-△5”となっており，“-”の文字と“5”の数字は空白文字で区切られているため，“-”はその後の数値“5”に付加されているものではなく，減算を表す演算子であるとわかります。

例 5：

中間表記法による数式…100*(-200)

逆ポーランド表記法による表記…100△-200*

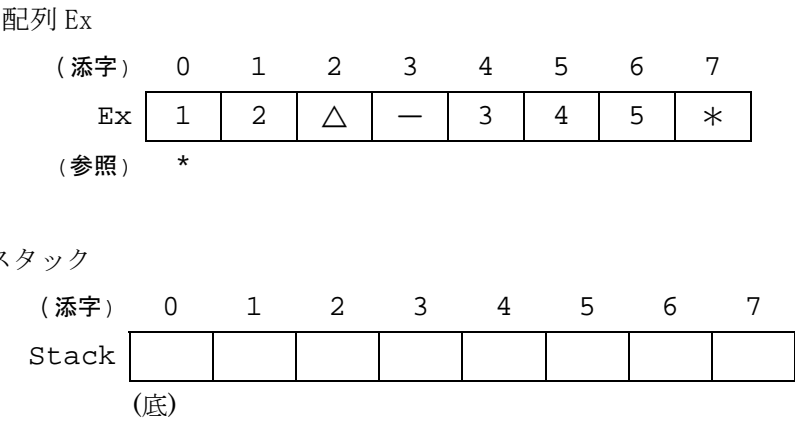
例 5 の表記では“-200”となっており，“-”の文字はその後の数値“200”に付加されている，負の数を示す値であるとわかります。

以上から，“-”の直後に数字があるか，それとも空白文字があるかによって，“-”の文字が減算を表す演算子なのか，負の数を表す文字なのかの区別をつけることができます。

“－”の直後に数字がある場合は負の数を表す文字，“－”の直後に数字がない場合は減算を表す演算子となります。

【設問】

本解説では、図 1 に示す逆ポーランド表記法の数式を例として扱います。図 1 はプログラムの開始時点の状態を示しており、スタックは空になっています。また、プログラムのすべての部分をトレースする例を用いると解説が煩雑になるため、空欄に関連する部分のみをトレースする例を用います。



((参照)は、現在参照している文字を“*”で示す欄である。)

図 1

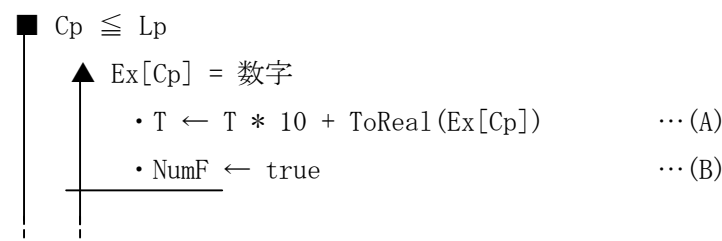


図 2

図 2 は、本問のプログラムの一部です。図 2 の(A)の式では、 $Ex[C_p]$ の数字を取り出し、その値を実数値に変換するなどの処理を行うことで、数字の並びを数値に変換する処理を行っています。

図 1 の $Ex[0] \sim Ex[1]$ の範囲に格納されている数字の並び“12”は、数値に変換すると 12.0 となります。T の初期値は 0 であり、 $C_p=0$ のときは $Ex[0]$ の数字“1”を取り出して(A)の式の処理を行い、T には $T \times 10 + 1.0 = 0 \times 10 + 1.0 = 1.0$ が格納されます。

また、図 1 の(B)の式で、NumF の値が true となります。数字を取り出した場合は、「現在

は数字の並びを扱っている状態である」ということを示すため、NumF の値を true としています。本問のプログラムでは、NumF の値が true ならば数字の並び（数値）を扱っているという状態を示し、そうでなければ “+” などの演算子を取り出して処理していることを示しています。

Cp=1 のときは Ex[1] の数字 “2” を取り出して (A) の式の処理を行い、T には $T \times 10 + 1.0 = 1.0 \times 10 + 2.0 = 12.0$ が格納されます。この措置により、Ex[0]~Ex[1] の範囲に格納されている数字の並び “12” を実数値に正しく変換した値が、T に格納されます。

この後、Cp=2 となり、Ex[2] に格納されている “△” を取り出した時点で、数字の並びが終わったこととなります。以降の処理で、現時点までに T に格納した 12.0 という値を、オペランドとしてスタックに積む処理などが行われます。

- 空欄 a, b

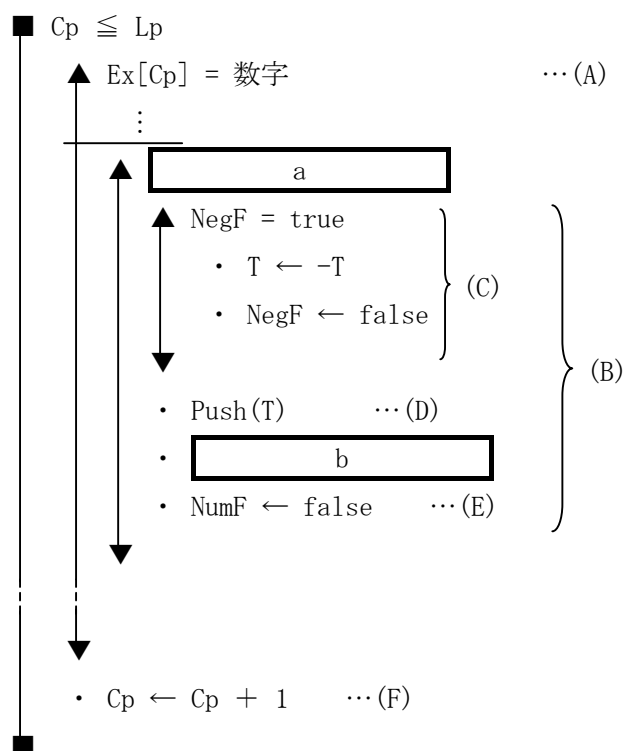


図 3

図 3 は、本問のプログラムの一部です。

図 3 の空欄 a 以降の処理に分岐するのは、図 3 の (A) の条件式が成立しなかった場合のみとなります。すなわち、配列 Ex から取り出した文字が数字でない場合 (“△”, “+” などの文字が取り出された場合) に、空欄 a 以降の処理に分岐します。空欄 a 以降の処理では、T に格納した実数値を、オペランドとしてスタックに積む処理などが行われます。

これまでの解説から、NumF の値が true ならば数字の並び（数値）を扱っているという状

態を示しており、Tに実数値が格納されているとわかります。よって、NumFの値がtrueであるとき、Tに格納されている実数値をオペランドとしてスタックに積む処理（図3の(B)の部分の処理）が行われるべきであるため、空欄a=“NumF = true(ウ)”となります。

図3の(C)の処理は、実数値に変換していた数字の並びが“−123”のような、負の値を示すものである場合に行うべき処理となります。本問のプログラムでは、“−123”のような負の値の数字の並びを処理するときには、数字の並びの先頭の“−”の文字はとりあえず置いておき、数字の並びの2番目の数字から末尾の数字までを先に取り出して正の数値に変換してTに格納します。その後、数字の並びの先頭の文字が“−”であった場合には、Tの数値を負に変換しています（図3の“ $\cdot T \leftarrow -T$ ”の式）。NegFは、数字の並びの先頭の文字が“−”である場合はtrue、そうでない場合はfalseとなる変数です。NegF=trueとなっているとき、図3の(C)の処理においてTの数値を負に変換します。

図3の(D)の式では、Tに格納されている実数値をスタックに積みます。その結果、Tに格納されていた12.0という値が、オペランドとしてスタックに格納されます（図4）。また、図3の(E)の式により、NumFの値はfalseになります。

配列 Ex

(添字)	0	1	2	3	4	5	6	7
Ex	1	2	△	−	3	4	5	*
(参照)				*				

スタック

(添字)	0	1	2	3	4	5	6	7
Stack	12.0							
(底)								

図4

この時点で、Tに格納されていた12.0という実数値はスタックに格納できたため、以降の処理では“−345”という数字の並びを実数値に変換し、その値(−345.0)をTに格納することになります。そのためには、Tをゼロクリアする必要があります。Tの値を12.0としたままで以降の処理を続けてしまうと、−345.0という値を正しく変換できないため、問題があります。

よって、空欄bでは“ $T \leftarrow 0$ (オ)”という処理が適切です。

なお、Ex[2]の文字は“△”であり、“+”などではないため、“Ex[Cp] = “+””などの条件式はすべて偽と判定されます。よって、図3の(F)の処理に遷移し、Cpの値が1加算されて3となります。

・ 空欄 c

Cp=3 のとき，“-345” という数字の並びの先頭の “-” を取り出すことになります（図 5）。このときは，以下のように処理が行われます。

配列 Ex

	0	1	2	3	4	5	6	7
(添字)								
Ex	1	2	△	—	3	4	5	*
(参照)				*				

スタック

	0	1	2	3	4	5	6	7
(添字)								
Stack	12.0							
(底)								

図 5

Ex[Cp]=Ex[3]= “-” のため，図 3 の (A) の条件式が成立せず，空欄 a 以降の処理に分岐します。このとき，NumF の値は false となっているため，図 3 の (B) の部分の処理，すなわち T に格納した実数値をオペランドとしてスタックに積む処理は行われません。

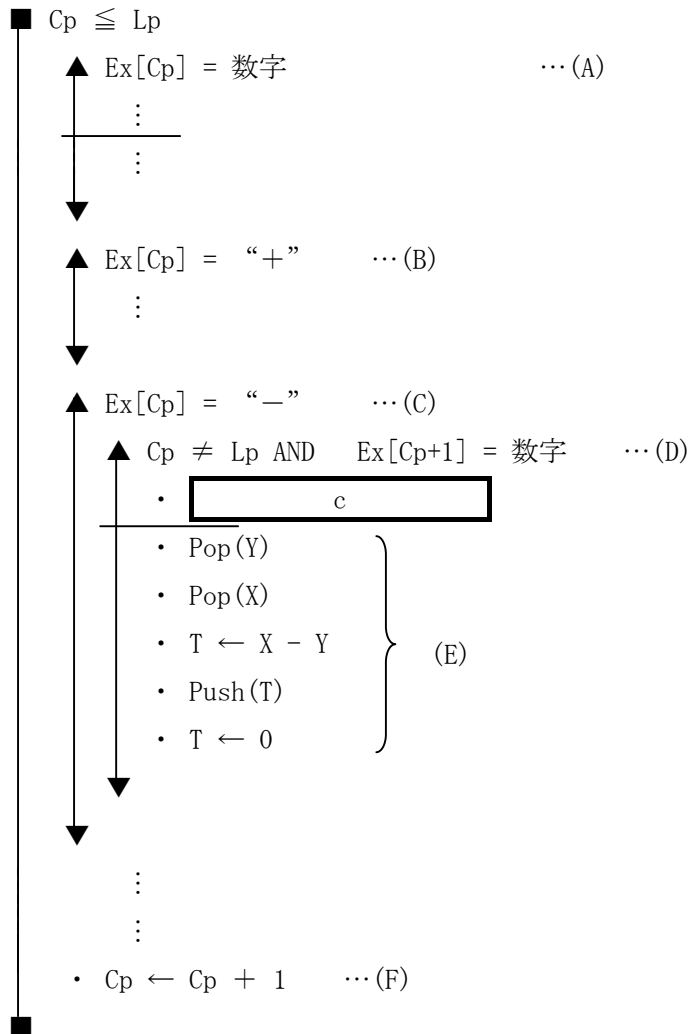


図 6

図 6 は、本問のプログラムの一部です。

$Ex[Cp] = Ex[3] = \text{“-”}$ ($\neq \text{“+”}$) のため、図 6 の (B) の条件式は成立せず、(C) 以降の処理に分岐します。そして、 $Ex[Cp] = Ex[3] = \text{“-”}$ のため、図 6 の (C) の条件式が成立し、図 6 の (D) の条件式の判断が行われます。

このとき、 $Cp(3) \neq Lp(7)$ であり、かつ $Ex[Cp+1] = Ex[3+1] = Ex[4]$ の文字が数字 (“3”) であるため、図 6 の (D) の条件式が成立し、空欄 c の処理が行われます。この条件式は、【逆ポーランド表記法】の例 5 において示した、「“-”の直後に数字がある場合は負の数を表す文字，“-”の直後に数字がない場合は減算を表す演算子」という判断を行うためのものです。

現在処理している文字 “-” の直後に数字があるため、この “-” は “-345” という数字の並びの先頭の文字、すなわち負の数を表す文字であり、減算を表す演算子ではありません。

せん。よってこのときは、減算処理（図 6 の(E)の処理）は行わず、NegF の値を true にする処理のみを行うべきとわかります。NegF は、これまでの解説の通り、数字の並びの先頭の文字が “-” である場合は true、そうでない場合は false となる変数です。

以上から、空欄c= “NegF ← true (イ)” となります。この処理により、NegFの値がtrue となります。そして、図 6 の(F)の式によってCpが1加算され、4 となります。

なお、図 6 の(D)の条件式が成立しないときは、“-” の文字の直後に数字がないことになり、処理している文字“-” が減算を表す演算子と判明するため、空欄 c の処理を行わず、代わりに減算処理（図 6 の(E)の処理）を行うことになります。

Cp=4~6 のときは、“-345” という数字の並びの “345” の部分が実数値 345.0 に変換され、その値が T に格納されます。そして、Cp=6 のとき、T に格納された実数値 345.0 が負の数（-345.0）に変換され、スタックに積まれます（図 7）。

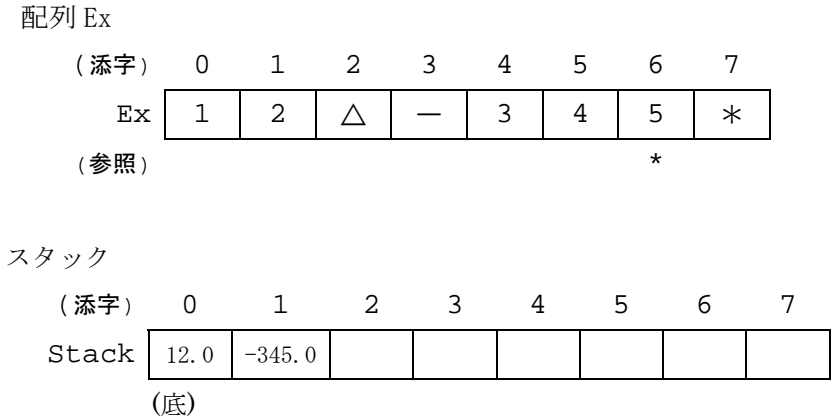


図 7

Cp=7 のとき、Ex[Cp]= “*” となります。このときは、図 8 の部分の処理が行われます。図 8 は、図 3 などと同様に本間のプログラムの一部です。

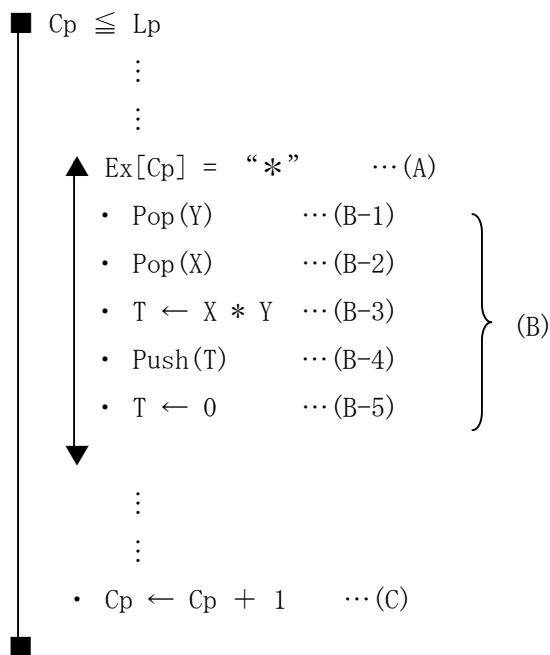


図 8

$Ex[Cp] = “*”$ のため、図 8 の(B)の部分の処理が行われます。(B-1)の式にてスタックからポップされた値が Y に代入され、 $Y = -345.0$ となります。(B-2)の式にてスタックからポップされた値が X に代入され、 $X = 12.0$ となります。スタックから 2 回ポップが行われ、この時点でスタックはいったん空になります (図 9)。

配列 Ex

(添字)	0	1	2	3	4	5	6	7
Ex	1	2	△	—	3	4	5	*
(参照)								*

スタック

(添字)	0	1	2	3	4	5	6	7
Stack								
(底)								

図 9

そして、(B-3)の式にて T に $X * Y$ の値 ($12.0 \times (-345.0) = -4140.0$) が代入されます。この値が、(B-4)の式にてスタックにプッシュされます (図 10)。

配列 Ex

(添字)	0	1	2	3	4	5	6	7
Ex	1	2	△	—	3	4	5	*
(参照)								*

スタック

(添字)	0	1	2	3	4	5	6	7
Stack	-4140.0							
(底)								

図 10

さらに、(B-5)の式にて T がゼロクリアされます。図 10 から、 $12.0 \times (-345.0)$ という計算を正しく行った値がスタックに乗っているため、プログラムの最後の “・Pop(Ret)” という処理により、この値を Ret にポップして計算結果として返します。

・空欄 d, e

〔プログラムの説明〕の(6)から、スタックの操作位置を示す変数 Sp の初期値は 0 となっています。関数 Push において、空の状態のスタックに引数 T の値をプッシュするとき、Stack[0] (スタックの底) に T の値を格納するためには、先に “・Stack[Sp] ← T” という処理を行い、その後に “・Sp ← Sp + 1” という処理を行うべきです。

先に “・Sp ← Sp + 1” という処理を行ってしまうと、Sp の値が先に 0 から 1 増加して 1 となり、その後に “・Stack[Sp] ← T” という処理が行われるため、Stack[0] には何も格納されないまま、Stack[1] に T の値が格納されてしまい、Stack[0] の領域が無駄になります。

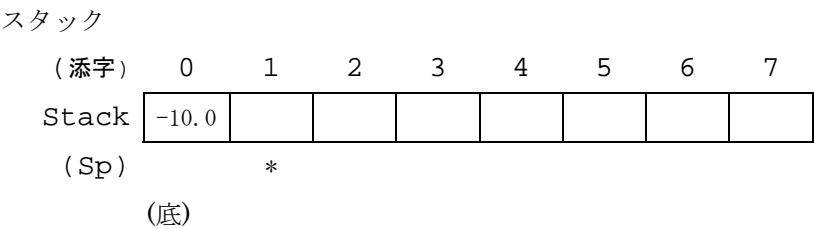
また、関数 Push では “ $Sp \leq MAX$ ” という条件式の判断を行い、 $Sp \leq MAX$ であれば空欄 d の処理を行うことになります。すなわち、 $Sp = MAX$ のときに、空欄 d の処理が行われる可能性があります。 $Sp = MAX$ のときに “・Sp ← Sp + 1” という処理を先に行うと、Sp の値が MAX + 1 となり、配列 Stack の最大要素番号を超えた位置に T の値が格納されてしまいます。

以上から、先に “・Sp ← Sp + 1” という処理を行うのは誤りであり、先に “・Stack[Sp] ← T” という処理を行うべきとわかります。

したがって、空欄 d は オ となります。それ以外の解答群は、先に Sp の値を加算したり、Stack[Sp] ではなく T に代入していたり、Sp の値を加算ではなく減算したりしているため、不適切です。

空欄 d = オとなっているため、例えば空のスタック ($Sp = 0$ の状態) に -10.0 という値をプッシュすると、先に “・Stack[Sp] ← T” という処理が行われて、Stack[0] に -10.0 が

格納されます。その後，“ $\cdot Sp \leftarrow Sp + 1$ ”という処理が行われ， Sp の値が 1 加算されて 1 となります。よって，図 11 のようになります。



((Sp)= Sp の値 (操作位置) を “*” で示す欄)

図 11

このスタックからポップする場合は，先に “ $\cdot Sp \leftarrow Sp - 1$ ” という処理を行って Sp の値を 1 減算して 0 とし，その後に “ $\cdot T \leftarrow Stack[Sp]$ ” という処理を行って， $Stack[Sp] = Stack[0]$ の値を取り出すのが適切となります。先に $Stack[Sp]$ の値を取り出すと， $Stack[Sp] = Stack[1]$ の値が取り出されてしまい，ポップによって返される値が不定となってしまいます。

以上から，空欄 e = エ となります。それ以外の解答群は，先に $Stack[Sp]$ の値を取り出したり， Sp の値を減算ではなく加算したりしているため，不適切です。

問 9

解答：

設問 1 ア

設問 2 イ

設問 3 ア

設問 4 イ

解説：

< 概要 >

本問のプログラムにおける各関数について、その処理の概要を以下に示す。

関数 rcheck：

自分の向いている方向に対して、右側の升目の状態を戻り値として返す。

	x-1	x	x+1
y-1	M[y-1][x-1]	M[y-1][x] (左向きなら ここが右側)	M[y-1][x+1]
y	M[y][x-1] (下向きなら ここが右側)	M[y][x]	M[y][x+1] (上向きなら ここが右側)
y+1	M[y+1][x-1]	M[y+1][x] (右向きなら ここが右側)	M[y+1][x+1]

関数 fcheck :

自分の向いている方向の升目の状態を戻り値として返す。

	x-1	x	x+1
y-1	M[y-1][x-1]	M[y-1][x] (上向きなら ここ)	M[y-1][x+1]
y	M[y][x-1] (左向きなら ここ)	M[y][x]	M[y][x+1] (右向きなら ここ)
y+1	M[y+1][x-1]	M[y+1][x] (下向きなら ここ)	M[y+1][x+1]

関数 go :

自分の向いている方向に進む。

このプログラムでは、以下のように迷路を進んでいる。

プログラム 21～24 行目 :

自分の右側が通路か、終点ならば、方向を右に変え (23 行目) その方向に進む。

プログラム 26～28 行目 :

自分の正面が通路か、終点ならば、そのままの向きで進む。そうでなければ、向きを左に変える。

すなわち、

1. 右側が開いているならば、右に進路を変えて進む。
 2. そうでなければ直進を試みる。
 3. 右にも進めず、直進もできなければ、左に向いて 1.に戻る。
- という一連の順序で迷路を辿っていることになる。

<設問 1>

前述の探索の順序に従うと、始点→①→②→①→③→⑤→⑥→⑤→⑦→終点 という順

番で迷路を辿ることになる。

<設問 2>

設問 1 と同様に，この探索の順序に従うと，

始点→①→②→③→④→①→②→③→④→…となり，終点にたどり着けず永久に①→②→③→④→①→…という順番で回りつづけることになる。

<設問 3>

プログラム中の，21 行目から 28 行目までの箇所において，21 行目，26 行目，28 行目の選択処理によって進行方向を決定している。よって，21 行目の if 文の直前では移動する前の位置が，if 文の直後では移動後の位置が得られることになる。

よって，始点を表示するには，移動前の位置が得られる if 文の直前に `printf` 文を追加すればよい。また，終点が判明するのは 29 行目の直後 (`while` ループから抜けた後) なので，29 行目の直後にも `printf` 文を追加する。

<設問 4>

`(dir+1) % 4` という処理を `dir` に加えると，`dir` の値は以下のように変化する。

<code>dir=1</code> (右)	…	<code>2</code> (下)
<code>dir=2</code> (下)	…	<code>3</code> (左)
<code>dir=3</code> (左)	…	<code>0</code> (上)
<code>dir=0</code> (上)	…	<code>1</code> (右)

よって，向きが右周りに変わる。逆に `(dir+3) % 4` という処理を `dir` に加えると，`dir` の値は以下のように変化する。

<code>dir=1</code> (右)	…	<code>0</code> (上)
<code>dir=2</code> (下)	…	<code>1</code> (右)
<code>dir=3</code> (左)	…	<code>2</code> (下)
<code>dir=0</code> (上)	…	<code>3</code> (左)

向きが左回りに変わることがわかる。関数 `rcheck` を関数 `lcheck` (左が開いているかを判定する関数) に変更したのだから，右回りではなく左回りで迷路を解くアルゴリズムに変更したと推察できる。よって，23 行目を左回り処理 (`(dir+3) % 4`)，28 行目を右回り処理 (`(dir+1) % 4`) に変更すればよいことになる。

問 10

解答：

設問 1 a カ b オ c エ d エ e オ

設問 2 ア

解説：

<概要>

マスタファイルをトランザクションファイルによって更新する処理である。この処理を行うためには乱呼び出し（レコードを先頭から順に呼び出すのではなく、順不同で呼び出すこと）を用いる。

トランザクションファイルによるマスタファイル更新処理では、マスタファイルのすべてのレコードが一律に更新されるわけではなく、トランザクションにおいて追加や変更・削除等の処理をされなかった商品のレコードは更新処理の対象外となる。よって、マスタファイルの全てのレコードを順呼び出しで順番に処理することはできないからである。

<設問 1 空欄 a,b>

この箇所は、ファイルの乱呼び出しにおいて定石とも言える定番の処理である。空欄 a,b の直後に乱呼び出しの **READ** 文があるので、**MASTER-FILE** からどのレコードを読み出して処理するかを特定するためのキーが必要になる。この処理では、商品コードがそれに該当する。

よって、更新情報が記録されているトランザクションファイル（**T**ファイル）の商品コード（**GOODS-TRA**）を、マスタファイルの商品コード（**GOODS-MAS**）に転記する。

<設問 1 空欄 c>

READ 文で無効キーが通知された場合、マスタファイル中に処理対象となるレコードは存在しないので、この状態で実行されるのは追加処理である。空欄 b の 4 行先に、**WRITE** 文があり、これがマスタファイルへの追加処理である。

よって、この箇所には、プログラムの説明(3)－①（**M**ファイルへの追加）に相当する処理（解答群エ）が適切である。解答群イなどは誤りなので注意が必要である（**T**ファイルの商品コード、分類コード、価格および名称がすべて記述されていないため）。

<設問 1 空欄 d>

この個所では、レコードの削除を実行するかどうかを判定している。この条件が真ならば削除を実行しているので、明らかにプログラムの説明(3)－③に相当する処理（解答群エ）が適切といえる。

<設問 1 空欄 e>

この条件が偽の場合は、ファイルの変更を行うことになる。変更処理では M ファイルを 1 回目に古い内容で、2 回目に新しい内容で出力している。M ファイルを変更する REWRITE 文の実行時には、M ファイルのレコード内容は新しい内容で出力しなければならないので、REWRITE 文直前の PERFORM 文によって、新しい内容を PRINT2（作業領域）に転記することになる。

よって、この PERFORM 文の前の WRITE 文で行う 1 回目の印字で古い内容を印字するため、M ファイルの内容を PRINT2 に転記する処理（解答群オ）が適切である。

<設問 2>

本問において、M ファイルには以下のキーが存在する。

M ファイル：

- (1) 49-0001-00011
- (2) 49-0021-00901
- (3) 49-0031-00002

これに対して、以下に示す T ファイルの各レコードについて比較し、適切な処理を決定する。

T ファイル：

49-0001-00011
49-0001-00012 000500NAME-4
49-0021-00901 001200
49-0022-00003A0102C000100NAME-5

1 件目：

M ファイルに存在し、更新情報が空白なので、「削除」される。

2 件目：

M ファイルには存在しないが、更新情報に空白個所があるため追加処理はできず、「エラー」となる。

3 件目：

M ファイルに存在し、更新情報に空白個所があるのでその箇所を「変更」する。

4 件目：

M ファイルに存在せず、また更新情報が全て存在するので、「追加」が適切である。これらに当てはまるのは解答群アの印字結果のみである。

問 11

解答：

設問 1 a カ b ウ c オ d ア

設問 2 ウ

設問 3 ウ

解説：

<概要>

プログラム 1: プログラムのメイン部分である。メソッド `main` において問題文の(1)～(5)の処理を行う。

プログラム 2: インタフェース `IComp` を宣言している。`IComp` はメソッド `compareTo` を宣言している。`compareTo` の処理内容の定義は、`IComp` を実装しているクラス `Computer` において行っている。

プログラム 3: `IComp` を実装している (空欄 a) クラス `Computer` を定義している。`Computer` はメンバ変数 `name`, `frequency`, `memory`, `price` やメソッド `getName`, `setPrice`, `toString`, `getDiscountRate`, および実装したインタフェース `IComp` からのメソッド `compareTo` を定義している (空欄 b)。

プログラム 4: `Computer` から継承したクラス `Workstation` を定義している。

プログラム 5: `Computer` から継承したクラス `PersonalComputer` を定義している。

プログラム 6: 値引き価格計算や整列を行うクラス `PriceUtility` を定義している (空欄 c, d)。

<設問 1 空欄 a>

プログラム 2, 3 よりクラス `Computer` はインタフェース `IComp` を実装する。インタフェース実装のキーワードが `implements` であり、その後には実装するインタフェース名 `IComp` (解答群カ) が入る。

<設問 1 空欄 b>

インタフェース `IComp` ではメソッド `compareTo` の引数は `IComp` 型である。`compareTo` の定義部分 (クラス `Computer` 内) において、この引数 a を `Computer` 型のオブジェクト `computer` に代入するには、型変換しなければならない。よって、このキャスト演算子の中には、`Computer` (解答群ウ) が入る。

<設問 1 空欄 c>

プログラム 6 のこの箇所において、値引き後の価格を求めている。値引き後の価格は、もとの価格 (`price` の値) に、 $(1 - \text{値引き率} (\text{getDiscountRate の戻り値}))$ を乗算すれば求

められる。もとの価格の値はメソッド `getPrice` によって返されるので、`getPrice`（解答群オ）が適切である。

<設問 1 空欄 d>

単純挿入法によって配列を降順に整列する場合、挿入すべき要素（`x` とする）の値と、整列済み配列の末尾要素の値とをまず比較の対象とし、`x` の値の方が大きい限り、末尾より一つ前の要素、二つ前の要素、…と、整列済み配列の比較対象要素を先頭に辿りながら、順に比較していく。そして、`x` の値の方が小さくなるか、同じ値となった時点で、比較対象要素の直後に `x` を挿入する。

プログラム 6 のメソッド `sort` では、挿入すべき要素を `temp` に代入しておき、整列済み要素配列 `a` の末尾要素（`a[j]`）から順に、`temp` の価格の値と `a[j]`、`a[j-1]`、…の価格の値とを比較している。よって、ここでは価格を比較するメソッド `compareWith` を呼び出すことになる。

<設問 2>

もともとの `compareWith` ではプログラム 3 の 27～28 行目を見ればわかるとおり、価格の値を比較している。この部分を設問 2 のように変更すると、動作周波数の降順に整列される。ただし、プログラム 6 の 16 行目（整列部分）において、`for` ループの継続条件より、挿入すべき要素（`temp`）と比較対象要素の動作周波数が同じ値のときは、交換を行わず比較対象要素の直後に `temp` を挿入している。よって、動作周波数が同じ値ならば、最初に出現した要素の方が配列の前方に並ぶことになる。

したがって、出力結果は、

1 : Ws1
2 : Ws2
3 : Pc1
4 : Pc2

である。

<設問 3>

クラス `PersonalComputer` の値引率を 0.4 に変更したので、`Pc1` と `Pc2` の値引き後の値が変化する。`Ws1`、`Ws2` には変化はない。

`Pc1` : $180,000 \times (1 - 0.4) = 180,000 \times 0.6 = 108,000$

`Pc2` : $200,000 \times (1 - 0.4) = 200,000 \times 0.6 = 120,000$

よって、解答群ウが適切である。

問 12

解答：

設問 1 a ウ b エ c オ

設問 2 ウ

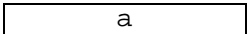
設問 3 イ

解説：

<概要>

本問のプログラムのうち、まず副プログラム **HEXDUMP** と、その処理内容を以下に示す。
なお、行番号は説明のためにつけたものである。

〔プログラム〕

```
1  HEXDUMP  START
2          RPUSH
3  α   ; 設問 2 参照
4  LP1     LAD   GR4,OUTBUF      ; 出力ポインタの初期化
5          LAD   GR3,16         ; 行内語数カウンタの初期化
6          LD    GR5,GR1
7          CALL  HEXCNV         ; アドレスの 16 進変換
8  LP2     LD    GR5,=' '
9          ST    GR5,0,GR4
10         LAD   GR4,1,GR4
11         LD    GR5,0,GR1
12         CALL  HEXCNV         ; 1 語分の 16 進変換
13         LAD   GR1,1,GR1
14         CPL   GR1,GR2
15         
16         SUBA  GR3,=1
17         JNZ   LP2            ; 1 行分 (16 語) 終了?
18         OUT   OUTBUF,OUTLEN
19         JUMP  LP1
20  FIN     LAD   GR5,OUTBUF
21         
22         ST    GR4,LENG
23         OUT   OUTBUF,LENG
24         RPOP
25         RET
```

```

26 OUTBUF DS 84
27 OUTLEN DC 84
28 LENG DS 1
29 END

```

1～5 行目：初期化处理である。

6 行目：GR1 の値（アドレス）を GR5 に代入する。

7 行目：GR5 の値（アドレスの値）を副プログラム HEXCNV によって 4 ビットごとに 16 進変換し、さらに文字に変換して、GR4 が示している出力領域に格納していく。

8～10 行目：GR4 が示す領域に空白文字を格納する。

11 行目：GR1 の値が示すアドレスに格納されている値を GR5 に代入する。

12 行目：GR5 の値を副プログラム HEXCNV によって 4 ビットごとに 16 進変換し、さらに文字に変換して、GR4 が示している出力領域に格納していく。

13～15 行目：GR1 の値を 1 増やす。GR1 と GR2（終端アドレス）を比較し、 $GR1 > GR2$ となった場合は、指定した範囲のメモリの内容をすべて変換したことになるので、20 行目に飛び、プログラム終了準備処理に移る（空欄 a）。

16～17 行目：GR3（一度に 1 行に出力する語数のカウント用のレジスタ）の値を 1 減らす。GR3=0 となった場合は 16 語の出力準備ができたことになるので、18 行目に飛び、出力領域（OUTBUF）の内容を出力する。そうでなければ次の語を変換するために 8 行目に戻る。

18～19 行目：出力領域に入った 16 語分の内容を出力し、アドレスの値の変換処理（6 行目）に戻る。

20 行目以降：ここに処理が移った時点で、出力領域に未出力の内容が残っている可能性がある。出力領域に残った変換内容の語数を計算して（空欄 b）出力し、プログラムを終了する。

本問では、16 進変換処理を以下のように行っている。

例えば、元の値を 0001 1001 0000 1100 とおくと、これを 4 ビットずつ分割し、

```

0001
1001
0000
1100

```

とする。

そして、それぞれを 16 進数値に変換し、さらに文字に変換する。

0001	→	1	→	“1”
1001	→	9	→	“9”
0000	→	0	→	“0”
1100	→	C	→	“C”

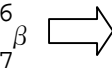
これらを，出力領域に 1 語ずつ順番に格納すればよい。

上記の処理を行っている，副プログラム **HEXCNV** の処理内容は以下のとおりである。
行番号は説明のためにつけたものである。

```

1  HEXCNV  START
2          LAD  GR6,0 ; シフト量を 0 に初期設定
3  LP      LD   GR7,GR5
4          SLL  GR7,0,GR6 ; 変換対象を上位 4 ビットに設定
5          

|   |
|---|
| c |
|---|


6  

|                    |
|--------------------|
| LD  GR7,CTABLE,GR7 |
| ST  GR7,0,GR4      |


7
8          LAD  GR4,1,GR4
9          LAD  GR6,4,GR6
10         CPA  GR6,=16 ; 1 語分の 16 進変換終了？
11         JMI  LP
12         RET
13  CTABLE  DC   '0123456789ABCDEF'
14         END

```

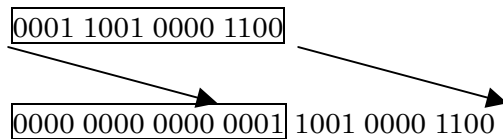
2 行目：初期化处理である。

3 行目：GR5 の値（16 進数に変換する内容）を GR 7 に代入し，以後の変換処理は GR7 に対して行う。直接 GR5 の値に 16 進変換処理を施さないのは，元の値を保存しておくためである。

4 行目：GR7 の値を GR6 の値と同じビット数分，左に論理シフトする。GR6 の値は 0 から始まるため，最初は 0 ビットシフトされる。

5 行目：GR7 の値を 12 ビット右に論理シフトすることにより，GR7 の先頭 4 ビットの値を取り出す（空欄 c）。

さきほどの例を用いると，元の値＝0001 1001 0000 1100 なので，これを 12 ビット右に論理シフトすると，



先頭 4 ビット (0001) のみが右端に残り，それ以外はオーバフローし，GR7 から消滅する。

6～7 行目：GR7 の値（2 進数値）を文字に変換し，GR4 が示す領域に格納する。

上の例では，GR7=0000 0000 0000 0001=1 なので，GR7 に (CTABLE+1) 番地の内容，すなわち “1” という文字が格納される。

8～9 行目：GR4 の値を 1 増やし，GR6 の値を 4 増やす。

10 行目以降：GR6 の値が 16 以上ならば，変換が終了したので元のプログラムに戻る。

なお，2 回目のループでは，GR6 の値=4 なので，以下のように変換が行われる。

GR7		
	0001 1001 0000 1100	
4 行目	1001 0000 1100 0000	(4 ビット左論理シフト)
5 行目	0000 0000 0000 1001	(12 ビット右論理シフト)

6～7 行目 GR7=0000 0000 0000 1001=9 なので，GR7 に (CTABLE+9) 番地の内容，すなわち “9” という文字が格納される。

3 回目以降のループも上記と同様に正しく動作する。

<設問 1 空欄 a>

前述のとおり，GR1（変換対象の語を示すアドレス）と GR2（変換する範囲の終端アドレス）を比較し，GR1>GR2 ならばすべての語の変換が終了したので，終了処理に入る。空欄 a の直前で「CPL GR1, GR2」という処理を行っているので，JPL 命令によって GR1>GR2 であるかどうか判定できる。よって，「JPL FIN」が適切である。

<設問 1 空欄 b>

例えば，GR1 と GR2 の初期値がそれぞれ 1000，1017 であるとする，1000 番地から 1017 番地までの 18 語を出力することになる。

16 語変換するごとに，それらを 1 行で出力するので，最初の行では 1000～1015 番地が出力される。2 行目では，1016～1017 番地のみ出力される。

この例の最初の行のように，変換した 16 語分を出力する場合は，出力文字数は決まって

いる（空白文字も含めて 84 文字分）ので、18 行目の OUT 命令で出力できる。しかしこの例の 2 行目のように、1 行に出力される語数が定まっていない場合は、出力のために出力文字数を算出しなければならない。21～23 行目で、その処理を行っている。

```
20  FIN  LAD   GR5,OUTBUF
21      

|   |
|---|
| b |
|---|


22      ST    GR4,LENG
23      OUT   OUTBUF,LENG
```

23 行目より、LENG が出力文字数と考えられる。GR4 は変換した文字を格納するアドレスを示しているので、その値から OUTBUF のアドレスの値を減算すれば、出力文字数を算出できる。

20 行目で GR5 に OUTBUF のアドレスの値を読み込んでいるので、21 行目では、

GR4-GR5 → GR4

という処理を行えばよいことになる。よって、「 SUBL GR4, GR5 」が適切である。

<設問 1 空欄 c>

前述のとおり、GR7 の値を右に 12 ビット論理シフトする処理が入る。

よって、「 SRL GR7, 12 」が適切である。

<設問 2>

先頭アドレスと終端アドレスが問題文の指定のようなパラメタ領域に格納されている場合は、パラメタ領域を指している GR1 を用いて、副プログラム HEXDUMP の最初（位置 α）で GR1, GR2 にそれぞれ先頭アドレス、終端アドレスをロードすればよいことになる。

よって、

LD GR2, 1, GR1

LD GR1, 0, GR1

という処理を行えばよい。

（先に LD GR1, 0, GR1 を行くと、GR1 に格納されていたパラメタ領域のアドレスが失われてしまうことに注意）

<設問 3>

変更後のプログラムを示す。ここで、行番号は説明のために加えたものである。

```
1      CPA   GR7,=9
```

```

2          JPL  AF
3          LAD  GR7,#0030,GR7 ; 数字 0 の文字コードを加算
4          JUMP CONT
5  AF      
6  CONT    ST   GR7,0,GR4

```

この処理では、変更前のプログラムで使用した領域 **CTABLE** を用いる代わりに、文字コード表を用いて数値から文字への変換を行っている。

COMET II の文字コード表では、**#0030**～**#0039** までの値がそれぞれ文字の“0”～“9”に対応している。すなわち、変換する数値が 0 ならば $\#0030 + 0 = \#0030 = \text{“0”}$ ，1 ならば $\#0030 + 1 = \#0031 = \text{“1”}$ ，…，9 ならば $\#0030 + 9 = \#0039 = \text{“9”}$ となるので、変換する数値に **#0030** を加えればよいことになる。

よって変換する数値（**GR7** の値）が 9 以下ならば、行番号 3 に分岐し、数値 + **#0030** を加算すれば変換が終了することになる。

しかし、変換する数値（**GR7** の値）が 10 以上の場合は、**#0030** を加えても“**A**”や“**B**”という文字にはならない（文字コード表を参照）ので、別の値を加えることになる

文字“**A**”～“**F**”までは、**#0041**～**#0046** にそれぞれ対応しているので、変換する数値に **#0030** ではなく、**#0037** を加えると適切な変換ができる。

よって、空欄には「 **LAD GR7,#0037,GR7** 」が適切である。

問 13

解答：

設問 1	a	ウ	b	エ	c	ウ	d	イ	e	ア
設問 2	f	オ	g	カ	h	カ	i	ア	j	カ
設問 3	k	ウ	l	ウ	m	イ	n	ウ	o	エ

解説：

この問題では稼働率計算などの知識が多少必要とされますが、表計算中には複雑な計算式や読解が困難な関数などは登場しないため、比較的容易に解答可能です。

・ 設問 1

この設問を適切に解答するためには、機器を二重化した場合の稼働率および故障率に関する知識が必要です。

・ 確率，事象，余事象

ある事象（出来事）P が発生する確率を p とすると、事象 P が発生することの逆の事象（余事象という）は、「事象 P が発生しない」ことになり、その確率（事象 P が発生しない確率）は、全体の確率（1）から事象 P が発生する確率 p を除いた、 $1-p$ となります。

逆に、事象Qが起こらない確率を q とすると、「事象Qが起こらない」ことの逆は「事象Qが発生する」ことになるため、事象Qが発生する確率 $=1-q$ となります。

ある機器の稼働率 $=X$ とすると、その機器が稼働しない確率は $(1-X)$ となります。機器を二重化すると、いずれか一方の機器が稼働していれば、全体として稼働しているものとみなされます。逆に、両方の機器が稼働していなければ、二重化した機器全体として稼働していないものとみなされます。

二重化した機器（単独での稼働率 $=X$ とする）が全体として稼働する事象を事象Qとみなした場合、「二重化した機器が稼働しない」（事象Qが起こらない）確率を q とすると、上記の説明より $q = (1-X) \times (1-X) = (1-X)^2$ となります。よって、事象Qが発生する確率 $=1 - \text{事象Qが起こらない確率} = 1 - q = 1 - (1-X)^2$ となります。

問題文表 1 の機器 1 を例にとって考えます。機器 1 の単一稼働率を X とすると、 $X=97.0\% = 0.97$ ですから、 $(1-X) = 0.03$ となり、

- ・ 二重化した機器 1 が稼働しない（故障する）確率

$$= (1-X)^2 = 0.03^2 = 0.0009$$

- ・ 二重化した機器 1 が稼働する確率（二重化後稼働率）

$$= 1 - (1-X)^2 = 1 - 0.0009 = 0.9991 = 99.91\%$$

となります。他の機器 2～4 も、同様にして二重化後稼働率を計算可能です。

- ・ 空欄 a, b

単一損失額を求めるには、機器を二重化せず単一にしていた場合の故障時間（単一故障時間）を求める必要があります。単一の機器が故障する（稼働しない）確率は、先の「確率、事象、余事象」の説明より、 $(1 - \text{単一稼働率})$ となります。

この確率を、対象期間（5 年間＝43,800 時間）に乗じたものが、単一故障時間になります。よって、空欄 a＝ $(1 - \text{単一稼働率})$ （ウ）となります。機器 1 について検算すると、 $43,800 \times (1 - 0.97) = 43,800 \times 0.03 = 1,314$ となり、問題文表 1 と比較して正しいことがわかります。

二重化後損失額を求めるには、機器を二重化した場合の故障時間（二重化後故障時間）を求める必要があります。二重化した機器が故障する（稼働しない）確率は、先の説明より $(1 - \text{単一稼働率})^2$ となります。

この確率を、対象期間（5 年間＝43,800 時間）に乗じたものが、二重化後故障時間になります。よって、空欄 b＝ $(1 - \text{単一稼働率})^2$ （エ）となります。こちらも機器 1 について検算すると、 $43,800 \times (1 - 0.97)^2 = 43,800 \times 0.0009 = 39,42$ となり、問題文表 1 と比較して正しいことがわかります。

・空欄 c, d

問題文の「機器別二重化効果額」および「機器別二重化投資効果額」の式に従って、各機器の機器別二重化効果額および機器別二重化投資効果額を計算した結果を、表 1 に示します。

表 1

機器	機器購入額 (千円)	単一損失額 (千円)	二重化後 損失額(千円)	保守費	機器別 二重化 効果額	機器別 二重化 投資効果額
機器 1	5000	26280	788	4000	21492	16492
機器 2	4000	8760	88	3200	5472	1472
機器 3	15000	35040	1402	12000	21638	6638
機器 4	10000	17520	350	8000	9170	-830

この表 1 より、機器別二重化効果額が最も高い（金額が多い）機器は機器 3（ウ）、最も低い（金額が少ない）機器は機器 2（イ）となります。機器 2 は単一故障時間が短いため、それに比例して単一損失額自体がもともと少なく、二重化した場合の改善効果が薄いことが読み取れます。

また、機器別二重化投資効果額が最も高い機器は、機器 1（ア）となります。

・設問 2

設問 1 では、故障 1 時間につき発生する損失額（時間損失額）を固定（20 千円）にしていましたが、この額が大きくなるほど単一損失額と二重化後損失額との差が大きくなるため、二重化のための投資を行うべきであると評価される機器が増えることになります。問題文図 1 のワークシートでは、時間損失額を 10 千円から 30 千円まで変化させた場合の、各機器の二重化投資効果額を求め、二重化のための投資を行うかを判断しています。

	…	D	E	F	G	H
⋮	…	⋮	⋮	⋮	⋮	⋮
2	…	単一 故障時間 (時間)	二重化後 故障時間 (時間)	時間損失額（千円／時間）		
3				10	20	30
4				二重化投資効果額（千円）		
5	…	1,314	39.42	3,745.8	16,491.6	29,237.4
6	…	438	4.38	−2,863.8	1,472.4	5,808.6
7	…	1,752	70.08	−10,180.8	6,638.4	23,457.6
8	…	876	17.52	−9,415.2	−830.4	7,754.4

図 1

図 1 は、問題文図 1 のワークシートの一部を説明のために取り出したものです。設問 2 の空欄 f～h は、機器別二重化投資効果額の式などを変形することで求められます。

・空欄 f～h

問題文に示された機器別二重化投資効果額の式は、

$$\text{機器別二重化投資効果額} = \text{機器別二重化効果額} - \text{機器購入額} \quad \cdots \quad (*)$$

となります。この式の「機器別二重化効果額」は、当該機器についての「単一損失額－二重化後損失額－保守費」で表せるため、

$$(*) = \text{単一損失額} - \text{二重化後損失額} - \text{保守費} - \text{機器購入額} \quad \cdots \quad \text{式 1}$$

と置き換えることができます。さらに、問題文の〔損失額の計算方法〕より、

$$\text{単一損失額} = \text{単一故障時間} \times \text{時間損失額} \quad (\text{設問 1 では 20 千円で固定})$$

$$\text{二重化後損失額} = \text{二重化後故障時間} \times \text{時間損失額} \quad (\text{同上})$$

となることから、

$$\begin{aligned} \text{式 1} &= \text{単一故障時間} \times \text{時間損失額} - \text{二重化後故障時間} \times \text{時間損失額} \\ &\quad - \text{保守費} - \text{機器購入額} \end{aligned}$$

$$= (\text{単一故障時間} - \text{二重化後故障時間}) \times \text{時間損失額} - \text{保守費} - \text{機器購入額}$$

となります。

問題文のセル F5 の式 “(f - g) * h - \$C5 - \$B5” と、上の式を照らし合わせると、\$C5＝機器 1 の保守費、\$B5＝機器 1 の機器購入額となっていることから、空欄 f＝機器 1 の単一故障時間、空欄 g＝機器 1 の二重化後故障時間、空欄 h＝時間

損失額であると推測できます。

図 1 より、機器 1 の単一故障時間のセル=D5，二重化後故障時間のセル=E5，列 F で参照すべき時間損失額（10 千円）のセル=F3 となるため、セル F5 に適切な式として、

$$(D5-E5)*F3-\$C5-\$B5 \quad \cdots \quad (\text{式 A とする})$$

という式がまず考えられます。ただし、この式 A は右方向および下方向の各セルに複写されるため、複写後のセルにとっても、この式が適切かどうかを検証する必要があります。

- ・セル F5 の式 A を下方向に複写する場合

セル F5 の式 A を、一つ下の行のセル F6 に複写すると、複写後の式 A の中のセル指定が 1 行下にずれ、以下のようになります。

$$(D6-E6)*\underline{F4}-\$C6-\$B6$$

この式の下線部は、時間損失額ではなく、不適切なセルを参照しています。よって、式 A のセル指定のうち、“F3”の部分の行指定を、“F\$3”のように絶対参照にしなければならないと判明します。

以上より、セル 5 の式は式 A のままでは不完全で、式 B のようにすることになります。

$$(D5-E5)*F\$3-\$C5-\$B5 \quad \cdots \quad (\text{式 B とする})$$

- ・セル F5 の式 B を右方向に複写する場合

セル F5 の式 B を、一つ右の行のセル G5 に複写すると、複写後の式 B の中のセル指定が 1 列右にずれ、以下のようになります。

$$(\underline{E5}-\underline{F5})*G\$3-\$C5-\$B5$$

この式の下線部の各セル指定は、単一故障時間などではなく、不適切なセルを参照しています。したがって式 B のセル指定のうち、“D5”および“E5”の部分の列指定を、“\$D5”，“\$E5”のように絶対参照にしなければならないと判明します。

なお、時間損失額のセル指定が右に 1 列ずれて“G\$3”となっていますが、この指定は適切です。

以上より、式 B でも不完全であり、式 C のようにすることになります。

$$(\$D5-\$E5)*F\$3-\$C5-\$B5 \quad \cdots \quad (\text{式 C とする})$$

この式 C の状態ならば、セル F5～H8 に複写しても、各機器の単一故障時間，二重化後故障時間，および各時間損失額を適切に参照できます。すなわち、空欄 f=\$D5（オ），空欄 g=\$E5（カ），空欄 h=F\$3（カ）となります。

- ・空欄 i, j

問題文図 1 を参照すれば、これらの空欄は容易に解答できます。

時間損失額が 10 千円の場合（空欄 i）

… 機器 1 の二重化投資効果額のみがプラス（3,745.8）であり、他の機器はすべてマイナスであるため、「機器 1 だけ、二重化のための投資を行う」（ア）が適切です。

時間損失額が 30 千円の場合（空欄 j）

… 機器 1～4 の二重化投資効果額がすべてプラスであるため、「対象機器すべてについて、二重化のための投資を行う」(カ) が適切です。

・設問 3

問題文図 2 のワークシートの範囲が広く、網掛け部分が多いため一見難解に見えますが、問題文の“(作成したワークシートの説明)”に従って各セルに適切な式を考えていくことで、すべての空欄について解答が可能です。また、網掛け部分のセルの大部分は、適切な式を求めなくとも解答する際には特に支障はありません。

このような種の設問を解く場合、空欄のセルに適切な式を求めるには、設問 2 と同様、以下のような手順をとります。

- (1) 空欄のセルに適切と思われる式を、仮に当てはめる。
- (2) その式を指定された方向に複写した際に、複写によって変更された式中のセル指定が複写先のセルにとっても適切かどうかを検証する。適切でない場合は、当該セル指定の列または行の指定を絶対参照に変える。

この方法をもって、空欄 k 以降の解答を考えます。

・空欄 k

先の設問 1 の解説などから、5 年間の単一損失額は、単一故障時間×時間損失額＝43,800 × (1－単一稼働率) × 時間損失額の式で求められます。ただしこの設問 3 では、1 年目から順に 1 年ずつ、1 年分の運転時間＝(43,800÷5)＝8,760 時間分の単一損失額を求めているため、1 年分の単一損失額＝8,760 × (1－単一稼働率) × 時間損失額となります。

問題文図 2 より、

- ・ 1 年分の運転時間 (8,760) … セル C7 に格納されている
- ・ 単一稼働率 … セル C3 に格納されている
- ・ 時間損失額 … セル C9 に格納されている

といえるため、セル C10 にはまず“(1－C3) * C7 * C9)”という式が仮に当てはめられます。

この式を 1 列右のセル D10 に複写する場合を考えると、複写後の式は“(1－D3) * D7 * D9”となり、下線部の指定が適切でないことになります(他のセル指定は、2 年目の運転時間および時間損失額を適切に示している)。よって、先の式の下線部の“C3”の指定を“\$C3”とする必要があります。

以上より、セル C10 に適切な式は“(1－\$C3) * C7 * C9)”となります(空欄 k＝ウ)。

なお、問題文には記載されていませんが、参考までにセル C11、C12 に格納される「ケース 1 の場合の二重化後損失額」および「ケース 1 の場合の二重化効果額」を求める式を示します。

セル C11:

二重化後の故障時間は、(1－二重化後稼働率(セル C4)) で求められる「二重化後に機器 1 が稼働しない(故障する)確率」と、「各年度の運転時間(1 年目の場合はセル C7)」

とを乗じて求められる。この二重化後の故障時間に、各年度の時間損失額（1年目の場合はセル C9）を乗じたものが二重化後損失額となるため、セル C11 に適切な式は“(1- $\$C4$)* $C7$ * $C9$ ”である。

セル C12:

各年度における二重化効果額は、単一損失額（1年目はセル C10）から二重化後損失額（1年目はセル C11）を引き、さらに各年度の保守費用（1年目はセル C8）を引いた金額である。よって、セル C12 に適切な式は“ $C10-C11-C8$ ”である。

・空欄 1

セル C13 には、1年目の二重化効果額（セル C12 を複製したもの）が格納されています。セル D13 には2年目までの二重化効果額の累積額（1年目の二重化効果額+2年目の二重化効果額）が格納されることになるため、セル C13 とセル D12（2年目の二重化効果額）を加算した“ $C13+D12$ ”という式が、このセル D13 に仮に当てはめられます。

この式を1列右のセル E13 に複製すると、複製後の式は“ $D13+E12$ ”となります。この場合、セル D13 の値は2年目までの二重化効果額の累積額（1年目の二重化効果額+2年目の二重化効果額）であり、セル E12 の値は3年目の二重化効果額であるため、両セルを加算した値は“1年目の二重化効果額+2年目の二重化効果額+3年目の二重化効果額”となり、3年目までの二重化効果額の累積額に等しくなります。よって、この複製後の式はセル E13 の式として適切であり、セル指定の一部を絶対参照にする必要はありません。なお、この式を更に右側の列のセルにそのまま複製しても、すべてのセルにおいて各年度までの累積二重化効果額を適切に示す式になります。

すなわち、セル D13 に適切な式は“ $C13+D12$ ”となります（空欄 1=ウ）。

なお、先ほどと同様、問題文には記載されていませんが、参考までにセル C14 に格納される「ケース 1 の場合の二重化投資効果額」を求める式を示します。

セル C14:

各年度における二重化投資効果額は、その年度までの累計二重化効果額（1年目はセル C13 の値）から、機器購入額（セル C2 の値）を引いた値である。よって、セル C14 に適切な式は“ $C13-\$C2$ ”である。

・空欄 m

ケース 2 では、2年目以降の時間損失額は毎年前年比の 15%だけ増加することになります。この場合、ある年の時間損失額=T とすると、その翌年の時間損失額= $T+T\times 0.15$ （増加分）= $1.15\times T$ となるため、翌年の時間損失額=当年の時間損失額 $\times 1.15$ という式で求められます。よって、セル D15 に入ることになる2年目の時間損失額を示す式として、まず“ $C15$ （1年目の時間損失額） $\times 1.15$ ”という式が仮に考えられます。

この式を1列右のセル E15 に複製すると、複製後の式は“ $D15\times 1.15$ ”となります。この

場合、複写後の式はセル D15 の値（2 年目の時間損失額）の 1.15 倍となり、すなわち 3 年目の時間損失額に等しくなります。よって、この複写後の式はセル E15 の式としても適切であり、セル指定の一部を絶対参照にする必要はありません。なお、この式を更に右側の列のセルにそのまま複写しても、すべてのセルにおいて各年度における時間損失額を適切に示す式になります。

すなわち、セル D15 に適切な式は“C15*1.15”となります（空欄 m=イ）。

なお、セル C16～C20 およびセル D19 に格納される、「ケース 2 の場合の単一損失額」、「ケース 2 の場合の二重化後損失額」、「ケース 2 の場合の二重化効果額」、「ケース 2 の場合の累積二重化効果額」および「ケース 2 の場合の二重化投資効果額」を求める式は、ケース 1 と同様にして得られ、それぞれ以下ようになります。

セル C16（ケース 2 の場合の単一損失額）：“(1-\$C3) *C7*C15”

セル C17（ケース 2 の場合の二重化後損失額）：“(1-\$C4) *C7*C15”

セル C18（ケース 2 の場合の二重化効果額）：“C16－C17－C8”

セル C19（ケース 2 の場合の累積二重化効果額（1 年目））：“C18”

セル D19（ケース 2 の場合の累積二重化効果額（2 年目まで））：“C19＋D18”

セル C20（ケース 2 の場合の二重化投資効果額）：“C19－\$C2”

・空欄 n, o

これまでの説明で示した、セル C10～C20 などに格納される式をすべて把握した上で、問題文図 2 のワークシートの網掛け部分をすべて表示した結果を図 2 に示します。

図 2 より、機器 4 の二重化投資効果額は、ケース 1 では 3 年目（セル E14）にてプラスに、ケース 2 では 4 年目（セル F20）にてプラスに転じています。

	A	B	C	D	E	F	G
1	(機器 4 に関する条件)						
2		機器購入額(千円)	10,000				
3		単一稼働率	98.00%				
4		二重化後稼働率	99.96%				
5	(機器 4 の二重化投資効果額算出)						
6	期間		1 年目	2 年目	3 年目	4 年目	5 年目
7	運転時間(時間)		8,760	8,760	8,760	8,760	8,760
8	保守費用(千円)		0	2,000	2,000	2,000	2,000
9	ケース 1	時間損失額(千円/時間)	30	30	30	30	30
10		単一損失額(千円)	5,256	5,256	5,256	5,256	5,256
11		二重化後損失額(千円)	105	105	105	105	105
12		二重化効果額(千円)	5,151	3,151	3,151	3,151	3,151
13		累積二重化効果額(千円)	5,151	8,302	11,453	14,604	17,754
14		二重化投資効果額(千円)	-4,849	-1,698	1,453	4,604	7,754
15	ケース 2	時間損失額(千円/時間)	20	23	26	30	35
16		単一損失額(千円)	3,504	4,030	4,634	5,329	6,129
17		二重化後損失額(千円)	70	81	93	107	123
18		二重化効果額(千円)	3,434	1,949	2,541	3,223	4,006
19		累積二重化効果額(千円)	3,434	5,383	7,924	11,147	15,153
20		二重化投資効果額(千円)	-6,566	-4,617	-2,076	1,147	5,153

図 2

なお、この図 2 のように問題文図 2 の網掛け部分の値をすべて求めなくとも、各ケースでの二重化投資効果額がプラスに転じる年度は以下のようにして求められます。

ケース 1 : セル D13 の、2 年目までの累積二重化効果額

＝問題文図 2 のセル C13 (＝セル C12) とセル D12 の値より、 $5,151 + 3,151 = 8,302$ となる。 $8,302 < 10,000$ (機器購入額) となるため、まだプラスにはならない。

セル E13 の、3 年目までの累積二重化効果額

＝同じく、問題文図 2 のセル D13 (上で求めた値) とセル E12 の値より、 $8,302 + 3,151 = 11,453$ となる。 $11,453 > 10,000$ (機器購入額) となり、この年にプラスになる。

ケース 2 : 問題文図 2 に表示されているセル F19 において、累積二重化効果額の値が始めて 10,000 を超えるため、この年 (4 年目) にプラスになる。