

問題 1 基数変換に関する文章中の に入れるべき適切な値または語句を選べ。

2 進数 1 ビットでは 0 と 1 の 2 種類の値を表すことができ、4 ビットでは、0000 ~ 1111 の (1) 種類を表すことができる。また、1000 種類を表すためには少なくとも (2) ビット必要になる。

2 進数 8 ビットでは 256 種類を表すことができ、整数を 2 の補数で負の値を表すと次の表のようになる。2 の補数とは、指定されたけた数より 1 けた多い最小値からの差で負数を表すもので、8 ビットで整数を表す場合は 2 進数で 100000000 からの差を用いる。しかし、2 進数の除算は誤りやすいので、一般的に 100000000 と 8 ビットの最大値 11111111 の差が 1 であることに着目して、けた借りの無い 11111111 からの差に 1 を加算することで 2 の補数を求める。また、正負で表現できる種類は同数で、0 を正の方に含めている。

基数変換表

	10 進数	2 進数
正の値	<input type="text"/> (3)	0 1 1 1 1 1 1 1
	0	0 0 0 0 0 0 0 0
	<input type="text"/> (4)	1 1 1 1 1 1 1 1
負の値	<input type="text"/> (5)	1 0 0 0 0 0 0 0

2 進数の右 2 ビットを 10 進数の小数点以下の値に割り当てると 2 進数では 00,01, 10,11 の 4 種類を用いて少数点以下の値を表すため、10 進数 0.1~0.9 の中で正しく 2 進数に変換できる値は (6) の 1 つしか無く、他の値では (7) が生じる。

(1)の解答群

ア . 4 イ . 1 6 ウ . 1 2 8 エ . 2 5 6 オ . 5 1 2

(2)の解答群

ア . 8 イ . 9 ウ . 1 0 エ . 1 1 オ . 1 2

(3)~(5)の解答群

ア . - 1 イ . + 1 ウ . - 1 2 7 エ . + 1 2 7
オ . - 1 2 8 カ . + 1 2 8 キ . - 2 5 5 ク . + 2 5 5
ケ . - 2 5 6 コ . + 2 5 6

(6)の解答群

ア . 0.1 イ . 0.2 ウ . 0.3 エ . 0.4 オ . 0.5
カ . 0.6 キ . 0.7 ク . 0.8 ケ . 0.9

(7)の解答群

ア . アンダフロー イ . オーバフロー ウ . けた落ち
エ . 情報落ち オ . 丸め誤差

問題2 次のオペレーティングシステムに関する設問に答えよ。

<設問1> タスクの制御に関する記述中の に入れるべき最も適切な字句を解答群から選べ。

現在のオペレーティングシステム（以下、OS という）には、複数タスクを並行して動作させる、いわゆるマルチタスク機能がサポートされている。タスクには、実行状態、実行可能状態、待ち状態の3つの状態があるが、これらの状態は、割り込みの発生によって遷移する。

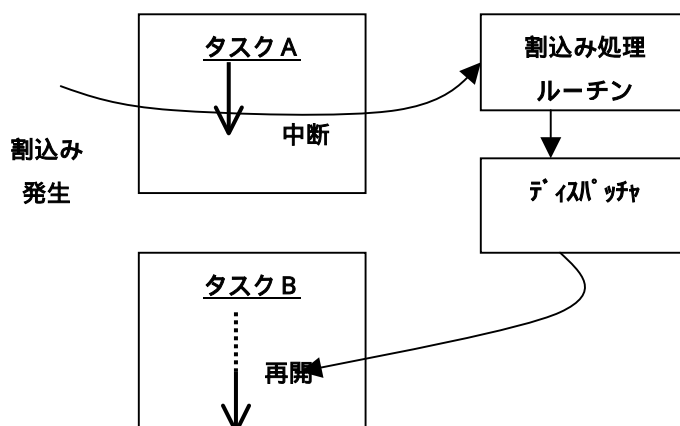


図1 割り込みによるタスクの切替え

図1は、割り込みの発生によって、実行中のタスクAの処理が中断され、別のタスクBの処理が再開される様子を示している。

割り込みが発生すると、今までタスクAに割り当てられていたCPUはOSの割り込み処理ルーチンに割り当てられる。この時点でタスクAの状態は、実行状態から (1) に遷移する。

割り込み処理ルーチンは、割り込みの種類ごとに必要な処理を行って、OSのディスパッチャを起動する。ディスパッチャは、あらかじめ決められたルールに従って、次にCPUを割り当てるタスクを決定する。これをタスクのスケジューリングという。

特定のタスクを優先して実行させたいような環境では、各タスクの優先度に基づいてCPUを割り当てる優先度順スケジューリングが用いられる。また、すべてのタスクをできる限り公平に実行させたいような環境では、 (2) スケジューリングが用いられる。

ディスパッチャによってCPUを割り当てられたタスクの状態は、 (3) に遷移する。

(1) (3) の解答群

ア . 実行状態 イ . 実行可能状態 ウ . 待ち状態 エ . 休止状態

(2) の解答群

ア . 到着順 イ . 実行時間順 ウ . ラウンドロビン エ . LRU

< 設問 2 > 仮想記憶システムに関する記述中の に入れるべき最も適切な字句を解答群から選べ。

仮想記憶システムは、実装されている主記憶装置（以下、主記憶という）の容量より大きなプログラムを実行できるようにする機能のことで、Windows や UNIX など多くのオペレーティングシステム（以下、OS という）で提供されている。

仮想記憶システムにおけるプログラム実行の基本的な考え方は、プログラム全体を主記憶上に配置するのではなく、 (4) のみを主記憶上に配置して、それ以外の部分は仮想記憶装置（以下、仮想記憶という）に配置する、というものである。主記憶上に配置されていないプログラムを実行するには、その部分を仮想記憶から主記憶に転送する必要がある。この転送作業を (5) という。転送の単位は、通常 4 キロバイトが用いられる。

仮想記憶システムでは、主記憶上のアドレス（実アドレス）と仮想記憶上のアドレス（仮想アドレス）の 2 つのアドレスを扱う。利用者から見ると、プログラムは仮想アドレスの空間で動作しているように見える。一方、CPU が命令を実行するには、主記憶上の実アドレスが必要になるので、仮想アドレスを実アドレスに変換するアドレス変換作業が必要になる。この作業は、 (6) 行う必要がある。アドレス変換作業を高速に実行するしくみとして「動的アドレス変換機構」(DAT ; Dynamic Address Translator) が用意されている。

(4) の解答群

ア . プログラムの命令部分 イ . プログラムのデータ部分
ウ . プログラムのメイン関数 エ . プログラムの実行に必要な部分

(5) の解答群

ア . ページイン イ . ページアウト ウ . スワッピング
エ . スラッシング オ . フラグメンテーション

(6) の解答群

ア . プログラムの実行を開始するときにまとめて
イ . ひとつ一つの命令を実行するたびに
ウ . 主記憶上のデータを更新する命令を実行するたびに
エ . 機械語のプログラムを生成するときに

問題3 次のプロセッサアーキテクチャに関する設問に答えよ。

<設問1> プロセッサの動作に関する記述中の [] に入れるべき適切な字句を解答群から選べ。

プロセッサはプログラムに従って処理を実行する。プログラムは人間が判読しやすいプログラミング言語で記述するが、プロセッサが直接判読できる (1) に変換されハードディスクなどの補助記憶に記憶される。命令の実行はプログラムを補助記憶からコンピュータ内部のメモリにロードし実行する。命令の実行はメモリに格納された1命令をプロセッサに取り出し解読する (2) サイクル、解読した命令を実行するサイクルの2つのサイクルで命令を実行する(図1参照)。

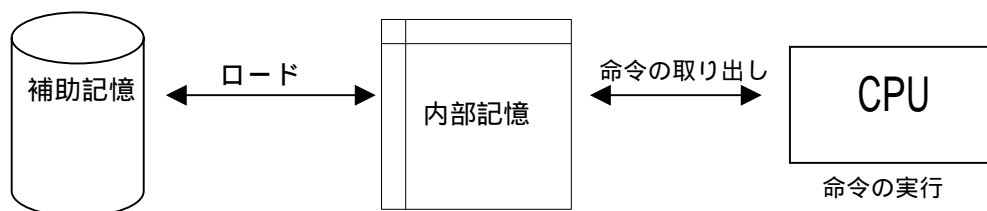


図1 プロセッサの動作

(1)の解答群

- ア．アセンブラ イ．機械語 ウ．高級言語 エ．低水準言語

(2)の解答群

- ア．エクスキュージョン イ．フェッチ ウ．マシン エ．基本

<設問2> プロセッサの構成に関する記述中の [] に入れるべき適切な字句を解答群から選べ。

プロセッサの基本機能として記憶機能、演算機能、制御機能がある。記憶機能はレジスタ群、演算機能は (3) (演算ユニット)、制御機能は制御部で構成される。レジスタ群は、次に実行される命令が格納されているアドレスが記憶されているプログラムカウンタ、メモリから取り出した命令を記憶する命令レジスタ、命令実行後の状態を記憶するフラグレジスタ、演算に使用するデータを記憶する (4)、など利用する用途により種類は異なる。

(3)の解答群

- ア．ALU イ．CISC ウ．MPU エ．RISC オ．SIMD

(4)の解答群

- ア．アキュムレータ イ．インデックスレジスタ ウ．スタックポインタ
エ．ベースレジスタ オ．アドレスレジスタ

<設問3> プロセッサのプログラム実行手順に関する記述中の[]に入れるべき適切な字句を解答群から選べ。

プログラム実行の手順は次のとおりとし、実行直前の内部メモリの状態を図2とする。なお、命令の形式は、命令部16ビット、アドレス部16ビットの固定長とし、アドレス指定方式は、直接アドレス指定方式とする。

[プログラム実行手順]

プログラムカウンタが示すアドレスから命令を命令レジスタに取り出す。

プログラムカウンタに で取り出した命令の長さをプログラムカウンタに加え、次に取り出すアドレスをプログラムカウンタに格納する。

デコーダにより命令を解読し、命令表に従って命令を実行する。

アドレス	データ	
0000	0A01	0101
0001	0A02	0102
0002	0A01	0103
0003	0A01	0102
0004	0A02	0103



0101	0000	0000
0102	0000	0003
0103	0000	0002

(注) 数値は16進数とする

図2 内部メモリの状態

表 命令の種類(抜粋)

命令	16進表記	命令の種類
LD(ロード)	0A01	実効アドレスの示す内容を汎用レジスタに設定する。
JMP(分岐)	0A02	実効アドレスの示す内容をプログラムカウンタに設定する。

プログラムカウンタが0000のとき、プログラムを実行すると命令レジスタの命令部は0A01となり、プログラムカウンタは0001となる。次にプログラムを実行すると命令レジスタの命令部は[(5)]となり、プログラムカウンタは[(6)]となる。

(5) の解答群

ア . 0A01 イ . 0A02 ウ . 0A03

(6) の解答群

ア . 0000 イ . 0001 ウ . 0002 エ . 0003 オ . 0004
カ . 0101 キ . 0102 ク . 0103

解 答

問題 1 (1) イ (2) ウ (3) エ (4) ア
 (5) オ (6) オ (7) オ

問題 2 (1) イ (2) ウ (3) ア (4) エ
 (5) ア (6) イ

問題 3 (1) イ (2) イ (3) ア (4) ア
 (5) イ (6) エ

問題1 次の流れ図の説明と流れ図を読み，設問に答えよ。

[流れ図の説明]

配列 $DT(i)$ ($i=1,2,\dots,10$) に格納されている値の昇順での順位を配列 $R(j)$ ($j=1,2,\dots,10$) の対応する位置に求めるものである。順位は1から始まり，同じ値がある場合は同順位とし，次の順位を欠番とする。

[流れ図]

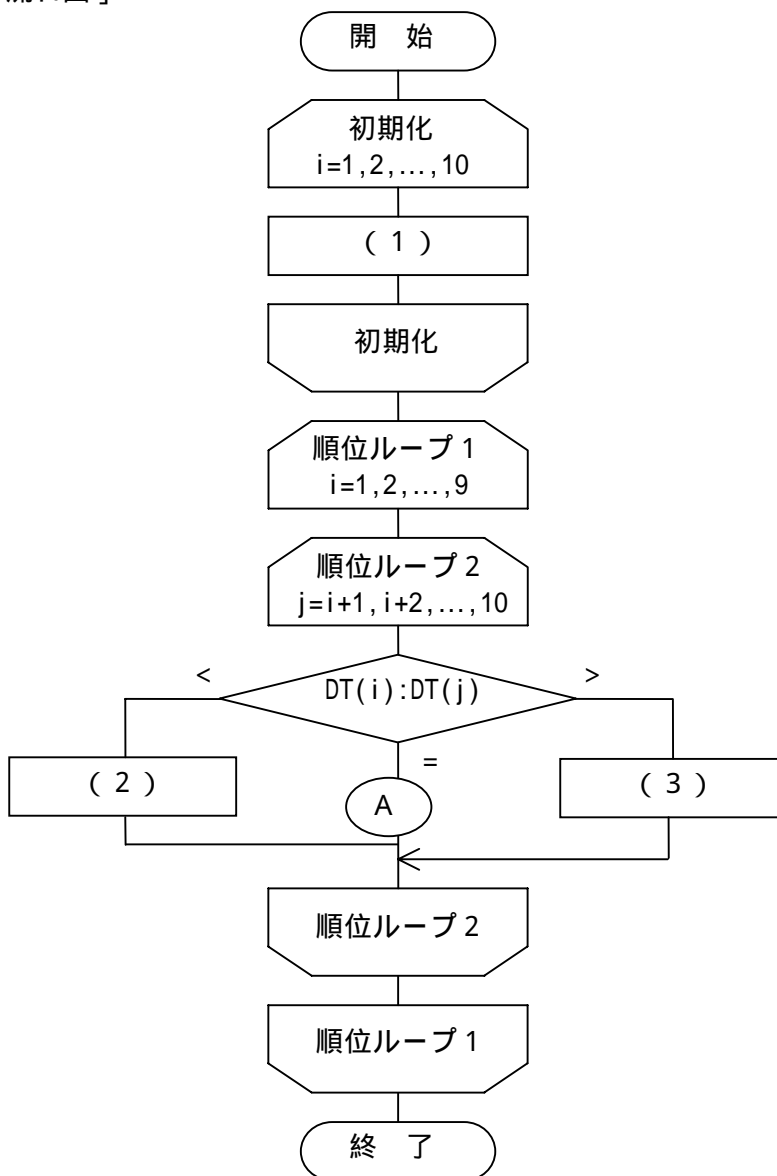


図 順位を求める流れ図

<設問1> 流れ図中の に入るべき適切な字句を解答群から選べ。

<設問2> この流れ図で $DT(i)$ と $DT(j)$ の比較を何回行うか, (4) の解答群から選べ。

<設問3> 配列 DT の内容が次のようであった場合、流れ図中の A を何回通過するか, (5) の解答群から選べ。

	(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)	(10)
DT	8	5	12	7	6	9	35	2	8	25

(1) ~ (3) の解答群

ア. $R(i) = 0$ イ. $R(j) = 0$ ウ. $R(i) = 1$
エ. $R(j) = 1$ オ. $R(i) = R(i) + 1$ カ. $R(j) = R(j) + 1$

(4) の解答群

ア. 9回 イ. 10回 ウ. 20回 エ. 25回 オ. 34回
カ. 35回 キ. 44回 ク. 45回 ケ. 54回 コ. 55回

(5) の解答群

ア. 0回 イ. 1回 ウ. 2回 エ. 3回 オ. 4回
カ. 5回 キ. 6回 ク. 7回 ケ. 8回 コ. 9回

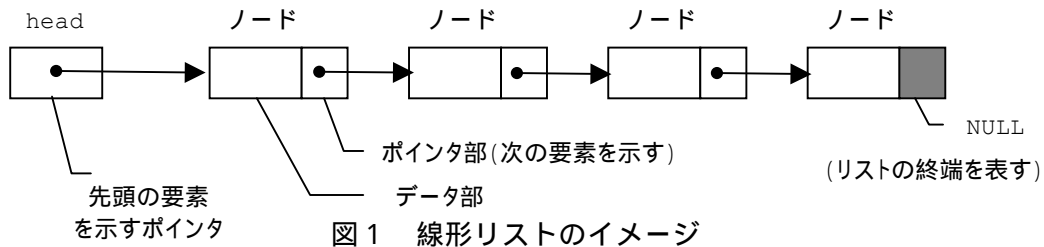
問題2 次の線形リストに関する説明を読んで、設問に答えよ。

[線形リストの説明]

線形リストとは、図1のようにデータ部とポインタ部からなる要素(ノード)をポインタで鎖状につなげたものである。ここでは、次の様なルールでリストを構成する。

head は、リストの先頭ノードのアドレスを保持する。リストにデータがない初期状態では NULL とする。

各ノードは、文字列データを保持するデータ部と、次のノードが格納されているアドレスを保持するポインタ部からなり、リストの終端のノードのポインタ部は NULL とする。



ここでは新たに確保した領域のアドレスは new に格納されているものとし、
 あるノードのデータ部は アドレス.data (例 new.data)
 あるノードのポインタ部は アドレス.pointer (例 new.pointer)
 と表現するものとして考える。また、確保された領域のデータ部およびポインタ部は初期状態では不定であるものとする。

<設問1> 次の文中の に入れるべき適切な字句を解答群から選べ。

リストに1件もデータが登録されていないとき、head は NULL となっているはずであるから、1件目のデータを登録するには、

- 手続き1
- 手続き2
- 手続き3 new.data リストに登録するデータ

の順に処理すればよい。

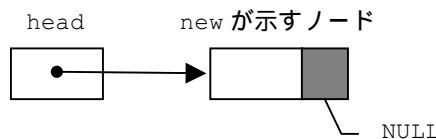


図2 1件目のデータ登録イメージ

(1) ~ (2) の解答群

- ア .head ← NULL
- イ .head ← new
- ウ .head ← new.pointer
- エ .new.pointer ← head

2 件目以降のデータの登録処理は、設問 2 ~ 設問 4 で

- A) リストの先頭に登録する場合
 - B) リストの末尾に登録する場合
 - C) リストの先頭および末尾以外に登録する場合
- の 3 つの場合に分けて考える。

< 設問 2 > 次の文中の に入れるべき適切な字句を解答群から選べ。

- A) 2 件目以降のデータをリストの先頭に登録する場合、
- 手続き 1
 - 手続き 2
 - 手続き 3 new.data ← リストに登録するデータ
- の順に処理すればよい。

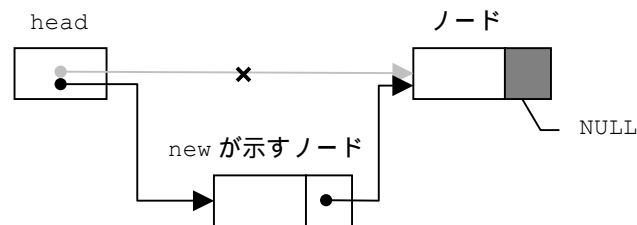


図 3 リストの先頭へのデータ登録イメージ

(3) ~ (4) の解答群

- ア .head ← new
- イ .head ← new.pointer
- ウ .new.pointer ← NULL
- エ .new.pointer ← head

<設問3> 次の文中の に入れるべき適切な字句を解答群から選べ。

B) 2件目以降のデータをリストの末尾に登録する場合、2件目以降の登録には、登録前の終端のノードを探す必要がある。現在注目しているノードの領域を示すポインタを here とすると、リストの先頭からポインタ部が NULL のノードを探し、見つかったら

手続き1

手続き2

手続き3 new.data ← リストに登録するデータ

の順に処理すればよい。

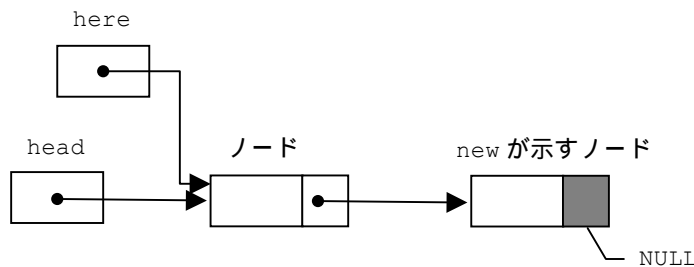


図4 リストの末尾へのデータ登録イメージ

(5)~(6)の解答群

- ア .new.pointer ← here
- イ .new.pointer ← here.pointer
- ウ .here.pointer ← new
- エ .here.pointer ← new.pointer

<設問4> 次の文中の に入れるべき適切な字句を解答群から選べ。

C) 2件目以降のデータをリストの先頭および末尾以外に登録する場合も、登録位置を探す必要がある。また、登録の際に直前のノードのポインタ部の変更が必要となるため、直前のノードの領域を示すポインタが必要になる。現在注目しているノードの領域を示すポインタを here、直前のノードの領域を示すポインタを prev として、here が示すノードの前に登録するには、

手続き1

手続き2

手続き3 new.data ← リストに登録するデータ

の順に処理すればよい。

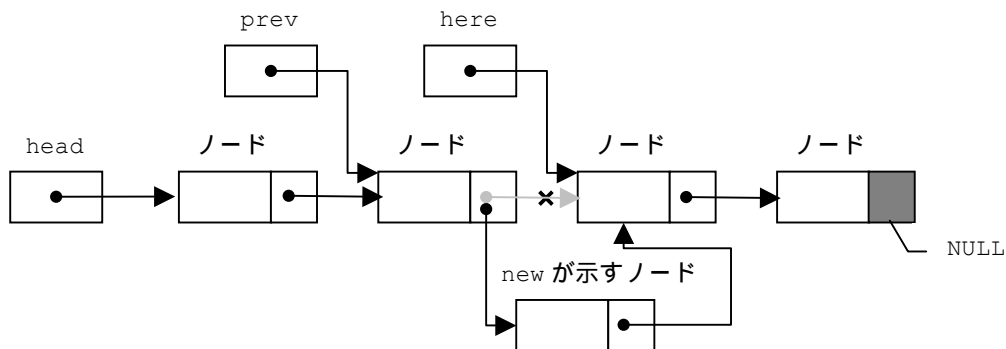


図5 リストの中間へのデータ登録イメージ

(7)~(8)の解答群

- ア .prev.pointer ← new
- イ .prev.pointer ← new.pointer
- ウ .new.pointer ← here.pointer
- エ .new.pointer ← prev.pointer

<設問5> 次の文中の に入れるべき適切な字句を解答群から選べ。

このように場合分けして考えてみると、リストの先頭に登録する場合には head を、中間に登録する場合は直前のノードのポインタ部を、末尾に登録する場合は終端のノードのポインタ部をそれぞれ意識する必要があり、リストのどの位置に登録するかによって処理手順が異なってくることに気づく。

そこで、リストに新たなデータを登録する際、登録位置によって処理手順が異なってしまうことを避けるために、リストの先頭と末尾にダミーノードを用意する。先頭と末尾のダミーノードのデータ部には、登録するノードのデータ部と比較したときに、必ず小さいと判定される特殊な値（ここでは $-\infty$ と表す）と、必ず大きいと判定される特殊な値（ここでは $+\infty$ と表す）を設定しておく。

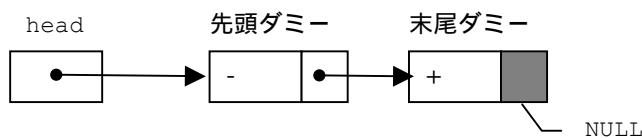


図6 ダミーノードを使ったリストの初期状態イメージ

このようにダミーノードを用意しておくと、リストのどの位置に登録する場合でも、同じ処理手順を使うことができる。現在注目しているノードの領域を示すポインタを here、直前のノードの領域を示すポインタを prev として、here が示すノードの前に登録するには、

- 手続き 1

(9)

 手続き 2

(10)

 手続き 3 new.data ← リストに登録するデータ

の順に処理すればよい。

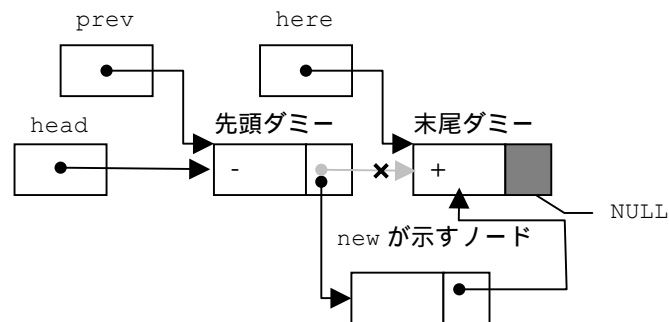


図7 ダミーノードを使ったリストへのデータ登録イメージ

(9) ~ (10) の解答群

- ア .prev.pointer ← new
- イ .prev.pointer ← new.pointer
- ウ .new.pointer ← here.pointer
- エ .new.pointer ← prev.pointer

問題 次のプログラムは、指定された年月日から、指定日数後の年月日を求めるものである。説明を読んで、設問に答えよ。

[プログラムの仕様]

1. 処理の概要

標準入力から年月日と指定日数を入力する。値が無効なときは、エラーメッセージを標準出力に出力する。値が有効なときは、入力された年月日から指定日数後の年月日を求めて出力する。なお、うるう年は考慮しない。

2. 関数

```
int check(int yy, int mm, int dd, int day);
```

機能：yy 年 mm 月 dd 日および day が示す指定日数が、有効かどうかを判断する。

年月日は、年が 0 以下のときやあり得ない月日のときに無効とする。

指定日数は、0 以下のときに無効とする。

返却値：年月日と指定日数が共に有効のときは 0

年月日が無効のときは 1

指定日数が無効のときは 2

年月日と指定日数が共に無効のときは 3

返却値の int 型データには、次のように下位 2 ビットに対してビット単位で値を設定する。

未使用(0にしておく)		: 年月日が無効のとき 1
-------------	--	---------------

: 指定日数が無効のとき 1

```
void day_cal(int *cyy, int *cmm, int *cdd, int yy, int mm, int dd,
int day);
```

機能：yy 年 mm 月 dd 日の day 日後の年月日を算出し、年は*cyy に、月は*cmm に、日は*cdd にそれぞれ格納する。

[プログラム]

```
#include <stdio.h>

int daytab[] = {0, 31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31};
void day_cal(int *cyy, int *cmm, int *cdd, int yy, int mm, int dd, int day);
int check(int yy, int mm, int dd, int day);

int main()
{
    int yy1, mm1, dd1, yy2, mm2, dd2, code, day;

    printf("年月日 (yy/mm/dd): ");
    scanf("%d/%d/%d", &yy1, &mm1, &dd1);
    printf("指定日数: ");
    scanf("%d", &day);
    code = check(yy1, mm1, dd1, day);
    if ( code & 0x01 )
        printf("年月日が無効です。¥n");
    if ( code & 0x02 )
        printf("指定日数が無効です。¥n");
    if ( code == 0 ) {
        day_cal(&yy2, &mm2, &dd2, yy1, mm1, dd1, day);
        printf("%d/%d/%d + %d = %d/%d/%d¥n",yy1, mm1, dd1, day, yy2, mm2, dd2);
    }
    return code;
}

void day_cal(int *cyy, int *cmm, int *cdd, int yy, int mm, int dd, int day)
{
    *cyy = yy;
    *cmm = mm;
    (2);
    while ( *cdd > daytab[*cmm] ) {
        *cdd -= daytab[*cmm];
        ++*cmm;
        if ( *cmm > 12 ) {
            ++*cyy;
            (3);
        }
    }
}

int check(int yy, int mm, int dd, int day)
{
    int code;

    code = 0;
    if ( yy <= 0 || (4) )
        code = 0x01; // 年月日のエラー
    if ( day <= 0 )
        (5); // 指定日数のエラー
    return code;
}
```

<設問1> 次に示すテストケースを入力したとき，想定される出力結果として正しいものを選び。

年月日	指定日数	出力結果
2000年4月31日	31	<input type="text" value="(1)"/>

(1)の解答群

- ア．2000/4/31 + 31 = 2000/5/31 イ．2000/4/31 + 31 = 2000/6/1
 ウ．年月日が無効です。 エ．指定日数が無効です。
 オ．年月日が無効です。
 指定日数が無効です。

<設問2> プログラム中の に入れるべき適切な字句を解答群から選べ。

(2)の解答群

- ア．*cdd = 0 イ．*cdd = dd
 ウ．*cdd = day エ．*cdd = dd + day

(3)の解答群

- ア．++*cmm イ．*cmm = 0
 ウ．*cmm = 1 エ．*cmm = 12

(4)の解答群

- ア．mm < 1 && mm > 12 || dd < 1 && dd > daytab[mm]
 イ．dd < 1 || dd > daytab[mm] && mm < 1 || mm > 12
 ウ．mm <= 1 || mm >= 12 || dd <= 1 || dd >= daytab[mm]
 エ．mm < 1 || mm > 12 || dd < 1 || dd > daytab[mm]

(5)の解答群

- ア．code = 0x02 イ．code |= 0x02
 ウ．code = 0x03 エ．code += 0x03

問題 次のプログラムに関する説明を読み各設問に答えよ。

【プログラムの説明】

D A T A 番地の 2 進数をゾーン 1 0 進数形式に変換し、A N S 番地に格納するプログラムである。ゾーン 10 進数形式は 1 バイト(8 ビット)ずつに区切り、10 進数 1 桁を格納する形式である。1 バイトの上位 4 ビットのゾーン部(1111)と下位ビットの数値部に分かれる。最下位バイトの上位 4 ビットには符号が入り、正のときは(1100)、負のときは(1101)が入る。

ゾーン部	数値部	符号	数値部
(1111)	10 進数 2 桁目	(1100)または (1101)	10 進数 1 桁目

(例) 2 進数で (1111 1111 1111 1101)₂ は符号つき 1 0 進数では - 3 である。したがって、ゾーン 1 0 進数形式で表現すると (1111 0000 1101 0011)₂ となる。

(アルゴリズム)

数値の正負を調べ、負なら(1101)₂を、正なら(1100)₂を符号に格納する。また、負なら、データを正に変換する。

10 で割った商を 10 進数の 2 桁目に、余りを 10 進数 1 桁目に格納する。

(プログラム)

```

PGM START
  LD   GR0,DATA
  JPL  L1
  JZE  L1
  XOR  GR0,=#FFFF ;
  ADDA GR0,=1      } 符号を変える
  ( 1 )
  JUMP L2
L1   LD   GR1,=#000C
L2   LAD  GR2,0
LOOP CPA  GR0,=10
     JMI  L3
     ( 2 )
     LAD  GR2,1,GR2
     JUMP LOOP
L3   LD   GR3,=#F000
     SLL  GR2,8
    
```

```

OR   GR3,GR2
SLL  GR1,4
OR   GR3,GR1
( 3 )
ST   GR3,ANS
RET
DATA DS 1
ANS  DS 1
END

```

<設問1> プログラム中の に入れるべき適切な命令を解答群より選べ。

(1)~(3)の解答群

```

ア LAD GR1,#000D   イ LD  =#GR1,000D   ウ LAD GR1,1,GR1
エ AND GR3,GR0     オ OR  GR3,GR0     カ SLL GR1,12
キ ADDA GR0, =10   ク LAD GR0,10     ケ SUBA GR0,=10

```

<設問2> DATA番地に $(FFC1)_{16}$ が格納されてこのプログラムが実行されたとき、プログラム実行後のGR0の値を解答群から選べ(10進数)。

(4)の解答群

```

ア 3   イ 6   ウ 8   エ 10  オ 12

```

解 答

共通問題

問題 1 (1)ウ (2)カ (3)オ (4)ク (5)イ

問題 2 (1)エ (2)イ (3)エ (4)ア (5)イ (6)ウ
(7)エ (8)ア (9)エ (10)ア

選択問題

C (1)ウ (2)エ (3)ウ (4)エ (5)イ

アセンブラ (1)イ (2)ケ (3)オ (4)ア

問題1 次のネットワークと IP アドレスに関する説明を読み,設問に答えよ。

[ネットワークと IP アドレスの説明]

インターネットやイントラネットなどの TCP/IP プロトコルを利用したネットワークでは,接続されたコンピュータや通信機器一つひとつに IP アドレスが割り当てられている。現在広く普及している IPv4 方式では,8 ビットずつ 4 つに区切られた 32 ビットの表現が使われ,“ 210.145.108.18 ” などのように,0 から 255 までの 10 進数の数値を 4 つ並べて表現する。

IP アドレスは,ネットワークアドレスとホストアドレスからなる。ネットワークアドレスはネットワークを識別するために使用し,ホストアドレスはそのネットワーク内のホスト(ルータなどのネットワーク機器やネットワークに接続されているコンピュータなど)を識別するために使用する。ネットワークアドレスとホストアドレスを表現する方法の一つに IP アドレスのネットワーク部とホスト部を決められたブロック単位で区切り,ブロックの長さによってクラス A~C として表現する方式がある(図 1)。この方式で,ホストの IP アドレスからネットワークアドレスを求めるには,サブネットマスクを用いる。例えば,クラス B の IP アドレスでは,ネットワークアドレスは上位 16 ビットなので,サブネットマスクは“ 255.255.0.0 ” となる。また,クラスによる方式は,IP アドレス空間の利用に無駄が生じることがあるので,クラス A~C の IP アドレスのホストアドレス部をサブネットアドレスとホストアドレスに細分化して利用する場合が多い。

クラス方式に対し,図 2 に示すようにクラスを使わない CIDR(Classless Inter-Domain Routing) と呼ばれる方式がある。この方式は,ネットワークアドレスとホストアドレスをビット単位で任意の長さで区切ることができるため,IP アドレス空間を有効に利用することができる。CIDR でも,ホストの IP アドレスからネットワークアドレスを求めるには,一般にサブネットマスクを用いる。

なお,クラス方式,CIDR 方式ともに,ホストアドレスとしてすべて「0」及びすべて「1」は使用できない。

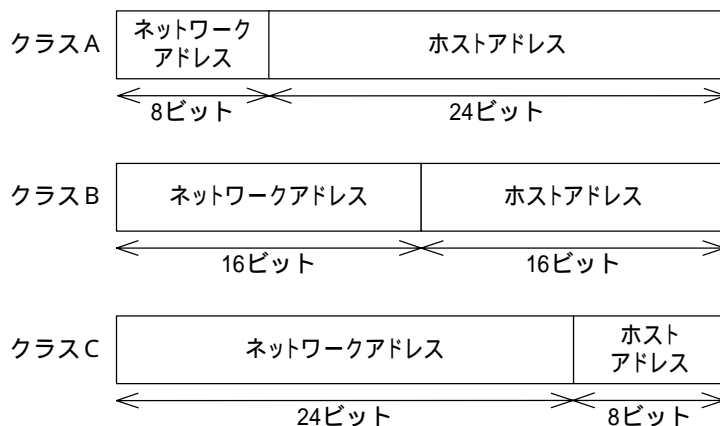
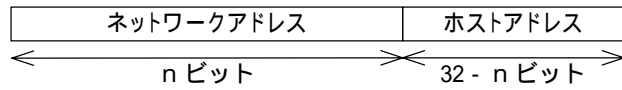
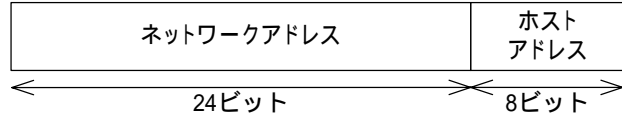


図 1. クラスを使用した IP アドレス構成図

CIDR表記：AAA.BBB.CCC.DDD/n nはネットワークアドレスを表すビット長を示す。



【例1】AAA.BBB.CCC.DDD/24



ネットワークアドレス24ビット，ホストアドレス8ビット（クラスCに相当）。

図2. CIDRを使用したIPアドレス

<設問1> クラスA~Cの方式を用いたネットワークとIPアドレスに関する次の記述中の に入れるべき適切な字句を解答群から選べ。

クラスAを使用して5,000台のホスト（ネットワーク機器も含む）を管理する場合を考える。表現できるホストアドレスは (1) 個に対し，実際に使用するホストアドレスは5,000個なのでIPアドレス空間に大幅な無駄が生じることになる。

また，クラスCを使用して500台のホストを管理するには，一つのネットワークでは最大でも254個しかホストアドレスを管理できないため，最低限 (2) つのクラスCのIPアドレスが必要となり， (2) つのネットワークで管理することになる。このとき，ネットワークどうしを接続するために，1台以上の (3) が必要になり，設備費用が増え， (3) 用のIPアドレスも必要になる。一方，CIDRを使用すると，ホストアドレス部は最低限 (4) ビットあれば500個のホストアドレスを表現できるので，IPアドレス空間に無駄が生じない。

(1)の解答群

- | | | | |
|-----------------|-----------------|-----------------|-----------------|
| ア. 2^8 | イ. 2^{16} | ウ. 2^{24} | エ. $2^8 - 1$ |
| オ. $2^{16} - 1$ | カ. $2^{24} - 1$ | キ. $2^{16} - 2$ | ク. $2^{24} - 2$ |

(2), (4)の解答群

- | | | | | |
|------|------|------|------|-------|
| ア. 1 | イ. 2 | ウ. 3 | エ. 4 | オ. 5 |
| カ. 6 | キ. 7 | ク. 8 | ケ. 9 | コ. 10 |

(3)の解答群

- | | | | |
|-----------|---------|---------|--------|
| ア. L2スイッチ | イ. ブリッジ | ウ. リピータ | エ. ルータ |
|-----------|---------|---------|--------|

<設問 2> IP アドレスとサブネットマスクに関する次の記述中の□□□□に入れるべき適切な数値または字句を解答群から選べ。

クラス C の IP アドレス “192.168.10.0” が割り当てられていて、接続するホスト数が 100 台のネットワークについて考える。このネットワークを一つのネットワークとして 100 台のホストを直接、接続する場合、サブネットマスクは “□(5)□” となる。

一方、100 台のホストを 20 台ずつ 5 つのサブネットワークに分割する場合、図 3 に示すように、クラス C の IP アドレスのホストアドレス 8 ビットをサブネットアドレス部とホストアドレス部に分割して使用する。サブネットワークは 5 つなので、サブネットアドレス部は最低限 3 ビットあれば、5 つのサブネットアドレスを表現できる。また、ホストアドレス部は 5 ビットとなるので、最大 □(6)□ 台のホストを接続でき、20 台のホストを接続するには十分である。このとき、適切なサブネットマスクは、“□(7)□” となる。

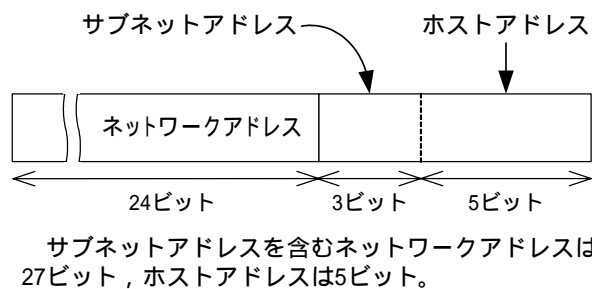


図 3 . ホストアドレスを分割したクラス C の IP アドレス

次に、クラス B の IP アドレス “179.170.0.0” が割り当てられているネットワークにおいて、CIDR を用いて 20 個のサブネットワークに分割する場合を考える。

初めに、クラス B のホストアドレス部 16 ビットをサブネットアドレスとホストアドレスに分割する。次に、ホストアドレス部 16 ビットのうち、20 個のサブネットワークアドレスを表すための必要最低限のビット数をサブネットワークのネットワークアドレスとして利用する。このとき、サブネットワークの IP アドレスは “179.170.0.0” ~ “179.170.248.0” の範囲となり、表現できるサブネットワーク数は最大 □(8)□ 個となる。これらのサブネットワークに適切なサブネットマスクは “255.255.248.0” である。20 個のサブネットワークのうち、ある一つのサブネットワークアドレスを CIDR 表記で表すと “179.170.168.0/ □(9)□” と表現できる。

(5), (7)の解答群

ア . 255.255.0.0	イ . 255.255.224.0
ウ . 255.255.248.0	エ . 255.255.255.0
オ . 255.255.255.168	カ . 255.255.255.192
キ . 255.255.255.224	ク . 255.255.255.248

(6), (8)の解答群

ア . 14	イ . 16	ウ . 20	エ . 24	オ . 28
カ . 30	キ . 31	ク . 32	ケ . 62	コ . 64

(9)の解答群

ア . 11	イ . 16	ウ . 21	エ . 27
--------	--------	--------	--------

<設問3> 次のネットワークの構成と問題に関する記述中の に入れるべき最も適切な字句を解答群から選べ。

図3は、クラスBのIPアドレスを基に、LAN-1～LAN-3の3つのサブネットワークからなるネットワークの構成図である。LAN-1～LAN-3の各サブネットワークには、次のようなサブネットアドレスが割り当てられている。例えば、LAN-1のホストアドレスは“172.16.1.1”～“172.16.15.254”となる。

LAN-1 : 172.16. 1.0/20

LAN-2 : 172.16.16.0/20

LAN-3 : 172.16.32.0/20

図3中の“172.16.XXX.YYY”は、ホストまたはルータのIPアドレスを示す。あるホストが同じネットワークにない別のホストにアクセスする際は、自身が接続されているネットワークのルータをデフォルトゲートウェイとする。例えば、ホストAのデフォルトゲートウェイはルータ1であり、デフォルトゲートウェイのIPアドレスは、“172.16.1.1”である。

このネットワークで、次の～に示す問題が発生した。なお、ルータ1,2のルーティング情報は正しく設定されている。

ホストAはホストB, D, Fと通信できるが、ホストHとは通信できない。
ホストEはホストDと通信できるが、ホストA, Fとは通信できない。
ホストGはホストFと通信できるが、他のすべてのホストとは通信できない。

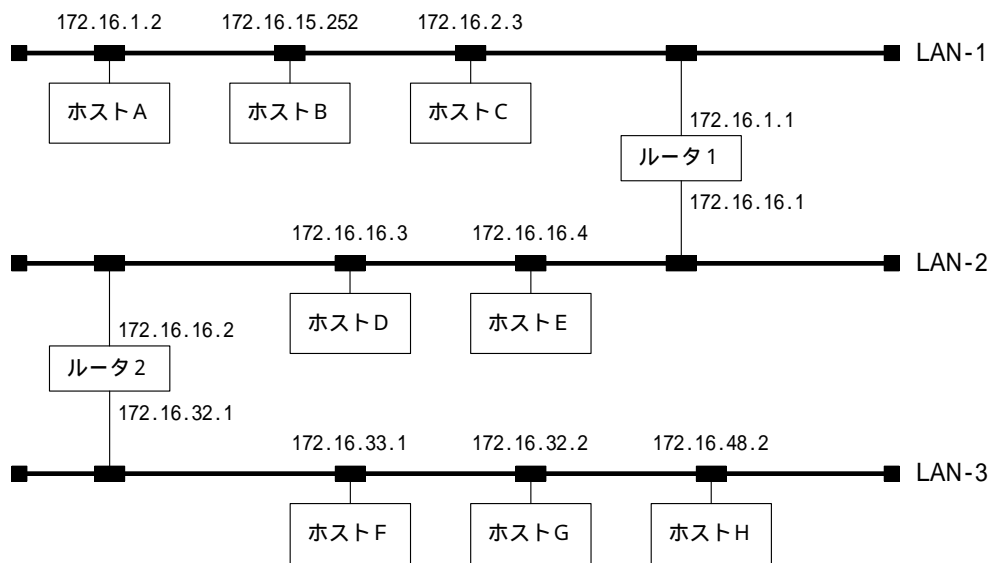


図3 ネットワーク構成図

は、ホスト H の IP アドレスが誤っていたことが原因であった。そこで、ホスト H の IP アドレスを (10) に変更したところ、ホスト A と通信できるようになった。

は、ホスト E のデフォルトゲートウェイは正しく設定されていたが、 (11) の設定が誤っていたことが原因であった。そこで、 (11) の設定を正しく変更したところ、ホスト A と通信できるようになった。

は、ホスト G のデフォルトゲートウェイの設定が誤っていたことが原因であった。そこで、ホスト G のデフォルトゲートウェイを (12) に変更したところ、他のすべてのホストと通信できるようになった。

(10)の解答群

- | | |
|-------------------|-------------------|
| ア . 172.16.33.1 | イ . 172.16.47.254 |
| ウ . 172.16.47.255 | エ . 172.16.48.1 |

(11)の解答群

- | | |
|----------------|--------------|
| ア . サブネットアドレス | イ . サブネットマスク |
| ウ . ネットワークアドレス | エ . ホストアドレス |

(12)の解答群

- | | |
|-----------------|-----------------|
| ア . 172.16.16.1 | イ . 172.16.16.2 |
| ウ . 172.16.32.1 | エ . 172.16.32.2 |

問題2 次の関係データベースに関する記述を読み各設問に答えよ。

J人材派遣会社では、派遣スタッフの情報を管理するため、リレーショナルデータベースを使っている。主な管理内容としては、次のものがある。

- ・ 資格取得者に関する各種一覧表の作成
- ・ 資格取得者の登録，削除，更新などのメンテナンス
- ・ 資格名称の登録，削除などのメンテナンス
- ・ 派遣先企業の登録，削除，更新などのメンテナンス

派遣スタッフ管理データベースで使用する表は、次のとおりである。

[各表の説明]

太字・下線の列名は主キーであることを示す。

スタッフ

スタッフ番号	氏名	性別	年齢	郵便番号	住所	支社番号	...
---------------	----	----	----	------	----	------	-----

支社

支社番号	支社名	郵便番号	所在地
-------------	-----	------	-----

取得資格

スタッフ番号	資格番号	取得日
---------------	-------------	-----

取得日は文字列型で YYYYMMDD 形式である（YYYY は西暦の 4 桁）

例えば，2006 年 9 月 10 日は '20060910' と表現する。

資格

資格番号	資格名	上級資格区分
-------------	-----	--------

上級資格区分は，最上位資格に 1，そうでないものには 0 が入っている。例えば，情報活用試験の場合は 1 級が最上位資格に該当する。

企業

企業番号	企業名	郵便番号	所在地	...
-------------	-----	------	-----	-----

<設問1> 特定資格の取得者を抽出する次の記述中の□□□□□に入れるべき適切な字句を解答群から選べ。

この人材派遣会社では、年内に最上位の資格を3つ以上取得したスタッフに対して奨励金を支給している。対象となるスタッフの番号と取得者数を表示するSQL文は、次のようになる。

```
SELECT スタッフ番号, COUNT(*)  
FROM 取得資格 G, 資格 L  
WHERE G.□(1) = L.□(1) AND □(2)  
GROUP BY スタッフ番号  
HAVING □(3)
```

(1) の解答群

- | | |
|----------|--------|
| ア．スタッフ番号 | イ．支社番号 |
| ウ．資格番号 | エ．取得日 |

(2) の解答群

- | | |
|--------------|---------------|
| ア．上級資格区分 = 0 | イ．上級資格区分 >= 0 |
| ウ．上級資格区分 = 1 | エ．上級資格区分 > 1 |

(3) の解答群

- | | |
|-----------------|-----------------|
| ア．COUNT(*) > 1 | イ．COUNT(*) > 2 |
| ウ．SUM(資格番号) > 1 | エ．SUM(資格番号) > 2 |

<設問 2 > 取得資格の登録に関する次の記述中の に入れるべき適切な字句を解答群から選べ。

このシステムでは、資格試験の科目合格も管理している。例えば、情報システム試験の場合、「基本スキル」と「システムデザインスキル」の両方の資格を取得すると「システムエンジニア認定」を取得したことになる。

科目合格の登録を行った時点で、過去に科目合格していれば正式に認定資格を取得したとして、新たに「取得資格」表へ追加する。

なお、この派遣会社では資格試験が実施された日付を取得日として登録しており、「取得資格」表には 2000 年以降に取得したデータだけを保持している。

今回の処理に必要な条件は、次のとおりである。

- ・ 「システムエンジニア認定」の資格番号は 'J110' である。
- ・ 「基本スキル」の資格番号は 'J111' である。
- ・ 「システムデザインスキル」の資格番号は 'J112' である。
- ・ 「基本スキル」と「システムデザインスキル」の両資格を取得し「システムエンジニア認定」を取得した場合、新たに「システムエンジニア認定」資格を登録する。
- ・ 上記の場合、「基本スキル」と「システムデザインスキル」の科目合格データは削除する。
- ・ 2003 年に制度改正が行われたため 2002 年 12 月 31 日以前の科目合格は無効である。
- ・ 今回の試験は 2006 年 9 月 10 日に実施された。(現在日時は 2006 年 9 月 11 日以降とする)

以上の条件から、次のような処理を実行する SQL 文が必要となる。

- (a) 「作業」表の作成
- (b) 有効期限切れの科目合格を削除
- (c) 対象スタッフの抽出
- (d) 新しい取得資格データの追加
- (e) 科目合格の削除

それぞれの処理に関する SQL 文は、次のようになる。

なお、2006 年 9 月 10 日に実施した試験の合格情報は、既に入力済みである。

(a) 「作業」表の作成

対象となるスタッフの取得資格情報を保持するための一時的な領域

```
(4) TABLE 作業(  
    スタッフ番号          DEC (7),  
    資格番号              CHAR (4),  
    取得日                CHAR (8),  
    上級資格区分          DEC (1)  
)
```

(b) 有効期限切れの科目合格を削除

```
(5) FROM 取得資格  
    WHERE 取得日 <= '20021231'  
    AND 資格番号 IN ('J111', 'J112')
```

(c) 対象スタッフの抽出

科目合格者を「作業」表へ取り出す

```
(6) INTO 作業  
    SELECT スタッフ番号,資格番号,取得日  
    FROM 取得資格 WHERE 資格番号 = 'J111'  
    AND スタッフ番号  
    IN (SELECT スタッフ番号 FROM 取得資格  
        WHERE 資格番号 = (7) )
```

(d) 新しい取得資格データの追加

「作業」表の資格番号と取得日を変更し、「資格取得」表に登録する。

```
(8) 作業  
    SET 資格番号 = (9) ,取得日 = '20060910',  
    上級資格区分 = 1
```

```
(6) INTO 取得資格 SELECT * FROM 作業
```

(e) 科目合格の削除

「資格取得」表から不要な科目合格を削除する。

```
(5) スタッフ番号 FROM 取得資格  
    WHERE スタッフ番号 IN (SELECT スタッフ番号 FROM (10) )  
    AND 資格番号 IN ('J111', 'J112')
```

(4)～(6), (8) の解答群

ア . CREATE イ . DELETE ウ . DROP
エ . INSERT オ . SELECT カ . UPDATE

(7), (9) の解答群

ア . 'J110' イ . 'J111' ウ . 'J112'

(10) の解答群

ア . スタッフ イ . 取得資格 ウ . 作業

<設問3> 派遣スタッフの資格取得状況に関する次の記述中の に入れるべき適切な字句を解答群から選べ。

ある企業にスタッフを派遣するため、企業が希望する資格を取得している人数を調査したところ、求人数に満たないことが分かった。そこで急遽その資格を取得していないスタッフのために養成講座を開くことにした。資格番号が 'J120' である資格の未取得者一覧表を作成する SQL 文は、次のようになる。

```
SELECT 支社名, スタッフ番号, 氏名, 性別, 年齢
FROM スタッフ S, 支社 T
WHERE S.支社番号 = T.支社番号
AND  (11)
      (SELECT * FROM 取得資格 G
       WHERE G. (12) = S. (12)
        AND 資格番号 = 'J120')
```

(11) の解答群

ア . EXISTS イ . NOT EXISTS
ウ . IN エ . NOT IN

(12) の解答群

ア . スタッフ番号 イ . 支社番号
ウ . 資格番号 エ . 支社名

解 答

問題 1	(1)ク	(2)イ	(3)エ	(4)ケ	(5)エ	(6)カ
	(7)キ	(8)ク	(9)ウ	(10)イ	(11)イ	(12)ウ
問題 2	(1)ア	(2)ウ	(3)イ	(4)ア	(5)イ	(6)エ
	(7)ウ	(8)カ	(9)ア	(10)ウ	(11)イ	(12)ア