

次の問 7 から問 11 までの 5 問については、この中から 1 問を選択し、選択した問題については、「選択」タブを開いて番号を選択して解答してください。

**問 7** 次の C プログラムの説明及びプログラムを読んで、設問 1, 2 に答えよ。

B 社では、セキュリティ管理のために、システムファイルから定期的に、システムの運用・保守用の利用者 ID 一覧を作成し、ファイルで保管している。

これらの利用者 ID に付加できる特権には、システムファイルの更新などができるシステム特権（以下、特権 S という）及びバックアップ作業などのために全てのファイルを参照できるオペレーション特権（以下、特権 O という）の 2 種類がある。

B 社では、セキュリティ管理強化の一環で、利用者 ID の追加・削除や特権の付加・解除が、申請に基づいて正しくシステムに反映されているかどうかを検証することになった。そのために、最新及び 1 世代前の利用者 ID 一覧を比較して、この間の利用者 ID 及び特権の登録内容の差異を印字するプログラムを作成する。

〔プログラムの説明〕

- (1) 最新の利用者 ID 一覧をファイル NewFile から、1 世代前の利用者 ID 一覧をファイル OldFile から、それぞれ読み込む。レコードは利用者 ID の昇順に整列されている。
- (2) 1 レコードは、利用者 ID (1 ~ 8 桁)、利用者名 (1 ~ 10 桁)、属性 (1 桁) 及び最終使用日 (8 桁) の 4 項目から成る。各項目は、空白文字で区切られている。

- ① 利用者 ID 及び利用者名は、英数字から成る文字列である。
- ② 属性は、次に示す内容の 8 ビット長のビット列 

0	1	0	0	s	o	g	r
---	---	---	---	---	---	---	---

 である。

上位 4 ビット： 固定値 0100  
 ビット s： 特権 S が付加されていれば 1, 付加されていなければ 0  
 ビット o： 特権 O が付加されていれば 1, 付加されていなければ 0  
 ビット g 及び r：その他の属性

- ③ 最終使用日は、数字から成る文字列で、その利用者 ID で最後にログインした年月日を表す。登録後、一度もログインしていない場合は、全桁が“0”である。
- (3) 利用者 ID 及び特権の登録内容の差異は、次のように印字する。
  - ① NewFile 中であって OldFile 中がない利用者 ID の場合  
 利用者 ID, 利用者名の後に“利用者 ID 追加”と印字する。この利用者 ID に特権 S が付加さ

れていれば“特権S付加”，特権Oが付加されていれば“特権O付加”を追加印字する。

②OldFile中においてNewFile中になく利用ユーザーIDの場合

利用ユーザーID，利用ユーザー名の後に“利用ユーザーID削除”と印字する。この利用ユーザーIDに特権Sが付加されていれば“特権S解除”，特権Oが付加されていれば“特権O解除”を追加印字する。

③NewFile及びOldFileの両方にあり，特権Sと特権Oの少なくとも一方の付加状況が変わった利用ユーザーIDの場合

利用ユーザーID，利用ユーザー名の後に“特権S付加”，“特権S解除”，“特権O付加”，“特権O解除”の該当する全てを印字する。

(4) 入力ファイルNewFile及びOldFileのデータ例を図1に，図1のデータ例を用いた実行結果を図2に，それぞれ示す。

ファイルNewFileのデータ例				ファイルOldFileのデータ例			
利用ユーザーID	利用ユーザー名	属性	最終使用日	利用ユーザーID	利用ユーザー名	属性	最終使用日
AE001	UserE1	41	20140419	AE001	UserE1	40	20140419
AE002	UserE2	48	20141014	AE002	UserE2	44	20140712
AE003	UserE3	48	20140716	AE003	UserE3	4C	20140716
AP005	UserP5	46	20141015	AP004	UserP4	45	20140715
AP006	UserP6	44	00000000	AP005	UserP5	44	20140713

注記1 属性の値（網掛け部分）は，16進数で表示している。

注記2 見出し行は，各ファイルには含まれないものとする。

図1 入力ファイルNewFile及びOldFileのデータ例

利用ユーザーID	利用ユーザー名	登録内容の差異
AE002	UserE2	特権S 付加 特権O 解除
AE003	UserE3	特権O 解除
AP004	UserP4	利用ユーザーID 削除 特権O 解除
AP006	UserP6	利用ユーザーID 追加 特権O 付加

注記 見出し行は，事前に印字されているものとする。

図2 図1のデータ例を用いた実行結果

(5) ライブラリ関数 strcmp(s1, s2) は，文字列 s1 と s2 を比較し，s1 < s2 のとき負の値を，s1 = s2 のとき 0 を，s1 > s2 のとき正の値を，それぞれ返す。

〔プログラム〕

```

#include <stdio.h>
#include <string.h>

#define BitS 0x08
#define BitO 0x04
#define BitR 0x01

FILE *NewFile, *OldFile;

```

```

int          NewEof,      OldEof;
char         NewID[9],   OldID[9];
char         NewName[11],OldName[11];
unsigned char NewAttr,   OldAttr;
char         NewDate[9], OldDate[9];

void ReadNewRecord() {

    fscanf(NewFile, "%s %s %c %s", NewID, NewName, &NewAttr, NewDate);
    if (feof(NewFile) != 0) {
        NewEof = EOF;          /* EOF: 負の整数定数 */
        strcpy(NewID, "\xFF"); /* \xFF: 8ビット符号の最大値 */
    }
}

void ReadOldRecord() {

    fscanf(OldFile, "%s %s %c %s", OldID, OldName, &OldAttr, OldDate);
    if (feof(OldFile) != 0) {
        OldEof = EOF;
        strcpy(OldID, "\xFF");
    }
}

void main() {

    NewFile = fopen("NewFile", "rb");
    OldFile = fopen("OldFile", "rb");
    NewEof = 0;
    OldEof = 0;
    ReadNewRecord();
    ReadOldRecord();
    while ( [ a ] ) {
        if (strcmp(NewID, OldID) == 0) {
            if ( [ b ] ) {
                printf("\n%-8s  %-10s", NewID, NewName);
                if ((NewAttr & BitS) > (OldAttr & BitS))
                    printf("  特権S 付加");
                if ((NewAttr & BitS) < (OldAttr & BitS))
                    printf("  特権S 解除");
                if ((NewAttr & Bit0) > (OldAttr & Bit0))
                    printf("  特権0 付加");
                if ((NewAttr & Bit0) < (OldAttr & Bit0))
                    printf("  特権0 解除");
            }
            [ c ]
        }
    }
}

```



```

else {
    if (  ) {
        printf("\n%-8s  %-10s  利用者ID 追加", NewID, NewName);
        if ((NewAttr & BitS) == BitS) printf(" 特権S 付加");
        if ((NewAttr & Bit0) == Bit0) printf(" 特権0 付加");
        ReadNewRecord();
    }
    else {
        printf("\n%-8s  %-10s  利用者ID 削除", OldID, OldName);
        if ((OldAttr & BitS) == BitS) printf(" 特権S 解除");
        if ((OldAttr & Bit0) == Bit0) printf(" 特権0 解除");
        ReadOldRecord();
    }
}
}
}
fclose(OldFile);
fclose(NewFile);
}

```

**設問 1** プログラム中の  に入れる正しい答えを、解答群の中から選べ。

**a に関する解答群**

- ア (NewEof != EOF) && (OldEof != EOF)
- イ (NewEof != EOF) || (OldEof != EOF)
- ウ NewEof != OldEof
- エ NewEof == OldEof

**b に関する解答群**

- ア ((NewAttr & OldAttr) & (BitS + Bit0)) != 0x00
- イ ((NewAttr | OldAttr) & (BitS + Bit0)) != 0x00
- ウ (NewAttr & (BitS + Bit0)) != (OldAttr & (BitS + Bit0))
- エ (NewAttr | (BitS + Bit0)) != (OldAttr | (BitS + Bit0))

**c に関する解答群**

- ア ReadNewRecord();
- イ ReadNewRecord();  
ReadOldRecord();
- ウ ReadOldRecord();

**d に関する解答群**

- ア (NewAttr & (BitS + Bit0)) != 0x00
- イ NewAttr > OldAttr
- ウ strcmp(NewID, OldID) < 0
- エ strcmp(NewID, OldID) > 0

## 設問 2 次の記述中の に入れる正しい答えを、解答群の中から選べ。

利用者 ID が有効に使用されているかどうかを検証するために、一定期間未使用、又は現在使用不可の利用者 ID 一覧を印字するプログラムを作成する。

- 最新の利用者 ID 一覧をファイル NewFile から、一定期間前の世代の利用者 ID 一覧をファイル OldFile から、それぞれ読み込む。
- 次の①、②の少なくとも一方に該当する利用者 ID について、利用者 ID、利用者名の後に、①に該当する場合は“現在使用不可”を、②に該当する場合は“期間中未使用”を印字する。
  - NewFile 中において、属性のビット r が 1 である利用者 ID  
ここで、属性のビット r は、利用者 ID が使用可能なら 0、使用不可なら 1 である。
  - NewFile 及び OldFile の両方において、最終使用日の値が等しい利用者 ID (全桁が“0”同士で等しい場合を含む)
- 図 3 に、図 1 のデータ例を用いた実行結果を示す。

利用者ID	利用者名	使用状況
AE001	UserE1	現在使用不可 期間中未使用
AE003	UserE3	期間中未使用

注記 見出し行は、事前に印字されているものとする。

図 3 図 1 のデータ例を用いた使用状況の印字結果

この処理を実装するためには、プログラム中の while 文のブロック内 (αで示した部分) を次のように変更すればよい。ここで、プログラム中の  a  には、正しい答えが入っているものとする。

```
if (strcmp(NewID, OldID) <= 0) {
    if (( e ) || ( f )) {
        printf("\n%-8s  %-10s", NewID, NewName);
        if ( e )
            printf("  現在使用不可");
        if ( f )
            printf("  期間中未使用");
    }
    ReadNewRecord();
}
else
    ReadOldRecord();
```

e, f に関する解答群

- ア (NewAttr & BitR) != (OldAttr & BitR)
- イ (NewAttr & BitR) == BitR
- ウ strcmp(NewDate, OldDate) == 0

- 工 `strcmp(NewID, OldID) == 0`
- オ `strcmp(NewID, OldID) == 0 && (NewAttr & BitR) == BitR`
- カ `strcmp(NewID, OldID) == 0 && strcmp(NewDate, OldDate) == 0`

## 問 7 午後のカギ

ファイルの突合せ処理を行う C プログラムの問題です。突合せ処理は午後問題でよく出題されるので、内容を理解しておきましょう。

設問 1 突合せ処理の基本的な仕組みの理解がポイントです。

設問 2 空欄 e には“現在使用不可”，空欄 f には“期間中未使用”かどうかを判定する条件式が入ります。

### 設問 1 プログラムの完成

2つのファイルからレコードを1件ずつ読み込んで比較していく処理を、**突合せ処理**といいます。突合せを行う2つのファイルは、共通のキーであらかじめ整列されている必要があります。本問の場合は、ファイル `NewFile` と `OldFile` が、どちらも利用者 ID の昇順に整列されています。

突合せ処理では、2つのファイルからレコードを1件ずつ読み込んで、両者のキーを比較し、その結果に応じて処理を行います。処理が終わったら次のレコードを読み込み、これを2つのファイルの末尾まで繰り返します。

本問の場合は、ファイル `NewFile` から読み込んだレコードの利用者 ID を `NewID`、ファイル `OldFile` から読み込んだレコードの利用者 ID を `OldID` とし、次のように比較しています。

#### ① `NewID < OldID` の場合

`NewID` は、`NewFile` 中にあって、`OldFile` 中に見えない利用者 ID です。たとえば `NewID` が「AE001」、`OldID` が「AE002」だった場合、`OldFile` に「AE001」のレコードは存在しません。

この場合、「AE001」は新たに追加された利用者 ID であることがわかります。プログラムは“利用者 ID 追加”の処理をして、次のレコードの突合せのために、`NewFile` から次のレコードを読み込みます。

#### ② `NewID > OldID` の場合

`OldID` は、`OldFile` 中にあって、`NewFile` 中に見えない利用者 ID です。たとえば `OldID` が「AE002」、`NewID` が「AE003」だった場合、`NewFile` に「AE002」のレコードは存在しません。

この場合、「AE002」は削除された利用者 ID であることがわかります。プログラムは“利用者 ID 削除”の処理をして、次のレコードの突合せのために、`OldFile` から次のレコードを読み込みます。

#### ③ `NewID = OldID` の場合

`NewFile` と `OldFile` の両方に、同じ利用者 ID のレコードがあります。この場合は、特権が変更されていないかどうかを調べて、変更されていればそれに合わせた処理を行い、`NewFile` と `OldFile` の両方から、次のレコードを読み込みます。

空欄 a：空欄の前後は次のとおりです。

```
while (  ) {
    if (strcmp(NewID, OldID) == 0 {
        ...
    }
}
```

空欄 a には、`while` 文に指定する繰返し条件が入ります。このプログラムでは、2つのファイルからレコードを1件ずつ読み込みます。2つのファイルを両方とも末尾まで読み込んだら、繰返しは終了です。

1件分のレコードの読み込みは、関数 `ReadNewRecord` と、関数 `ReadOldRecord` とによ

て行います。ファイルの末尾に達すると、関数 ReadNewRecord は変数 NewEof に定数 EOF を設定します。同様に、関数 ReadOldRecord は、変数 OldEof に定数 EOF を設定します。

ファイル NewFile を最後まで読み込んだ場合

```
if (feof(NewFile) != 0) {
    NewEof = EOF; ←変数NewEofにEOFを設定
    strcpy(NewID, "\xFF");
}
```

変数 NewEof と変数 OldEof の両方に EOF が設定されたら、繰返しは終了です。言い換えると、変数 NewEof と変数 OldEof のどちらか一方が EOF でない間は、繰返しを継続しなければなりません。これを条件式で表すと、

```
(NewEof != EOF) || (OldEof != EOF)
```

となります。これが、空欄 a に入る条件式になります。空欄 a は  です。

空欄 b：空欄の前後は次のとおりです。

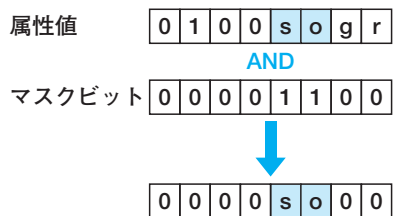
```
if (strcmp(NewID, OldID) == 0) {
    if (  ) {
        printf("\n%-8s  %-10s", NewID,
                NewName);
        if ((NewAttr & BitS) > (OldAttr
                                & BitS))
            printf("  特権 S 付加 ");
        if ((NewAttr & BitS) < (OldAttr
                                & BitS))
            printf("  特権 S 解除 ");
        if ((NewAttr & Bit0) > (OldAttr
                                & Bit0))
            printf("  特権 0 付加 ");
        if ((NewAttr & Bit0) < (OldAttr
                                & Bit0))
            printf("  特権 0 解除 ");
    }
    
}
```

このブロックでは、「読み込んだレコードの NewID と Old が等しい」=「2つのファイルに同じ利用者 ID のレコードが存在する」場合の処理を行います。

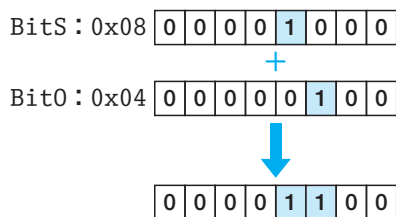
プログラムは、この利用者 ID の特権が変更されて

いる場合はその旨を表示します。空欄 b には、それを確認するための条件式が入ります。

利用者の特権は、8ビット長の属性のビット s とビット o に記録されています。属性からビット s とビット o を取り出すには、次のようなマスクビットを使って、属性値とマスクビットとの AND を取ります。



プログラムでは、定数 BitS が 0x08、Bit0 が 0x04 のように定義されているので、BitS + Bit0 でマスクビット 00001100 が作れます。



利用者 NewID の属性値 NewAttr と、利用者 OldID の属性値 OldAttr から特権ビットを取り出すには、それぞれ次のようにします。

```
NewAttr & (BitS + Bit0)
OldAttr & (BitS + Bit0)
```

この2つの値が異なっていれば、特権に何らかの変更が加えられたこととなります。空欄 b に入る条件式は次のようになります。

```
(NewAttr & (BitS + Bit0)) != (OldAttr
& (BitS + Bit0))
```

以上から、空欄 b は  です。

空欄 c：突合せ処理の説明のところで述べたように、NewFile と OldFile の両方に、同じ利用者 ID のレコードがあった場合、特権が変更されていたときにはそれに応じた処理を行い、その後 NewFile と OldFile の両方から、次のレコードを読み込みます。

したがって空欄 c には、2つのファイルから次のレコードを読み込む処理が入ります。

```
ReadNewRecord();
ReadOldRecord();
```

以上から、空欄 c は **イ** です。

空欄 d : 空欄の前後は次のとおりです。

```
if (  ){
    printf("\n%-8s  %-10s  利用者ID 追加",
           NewID, NewName);
    :
    ReadNewRecord();
}
else {
    printf("\n%-8s  %-10s  利用者ID 削除",
           OldID, OldName);
    :
    ReadOldRecord();
}
```

プログラムは、空欄 d の条件式が真の場合は利用者が追加された場合の処理を、そうでない場合は利用者が削除された場合の処理を実行します。

NewFile から読み込んだ NewID が、OldFile から読み込んだ OldID より小さい場合、NewID は追加された利用者 ID です。したがって空欄 d は、NewID が OldID より小さい場合に真になる

```
strcmp(NewID, OldID) < 0
```

が入ります。空欄 d は **ウ** です。

## 設問 2 プログラムの変更

空欄 e, f : 空欄の前後は次のとおりです。

```
if (  )
    printf(" 現在使用不可 ");
if (  )
    printf(" 期間中未使用 ");
```

ファイル NewFile にある利用者のレコードを調べて、空欄 e の条件式が真なら“現在使用不可”，空欄 f の条件式が真なら“期間中未使用”を印字します。

“現在使用不可”に該当するかどうかは、利用者の属性 NewAttr 中のビット r が 1 かどうかを調べます。これは次のような式で検証できます。

```
(NewAttr & BitR) == BitR
```

以上から、空欄 e は **イ** です。

次に、“期間中未使用”に該当するかどうかは、NewFile と OldFile にある同じ利用者 ID のレコードの最終使用日 NewDate と OldDate を比較し、両者が一致するかどうかを調べます。これは次のような式で検証できます。

```
strcmp(NewID, OldID) == 0 &&
strcmp(NewDate, OldDate) == 0
```

このように、①利用者 ID が等しいと②最終使用日が等しいという 2 つの条件式を、「かつ」を表す演算子 && でつなぎます。以上から、空欄 f は **カ** です。



### ○ 解答 ○

- 設問 1 a - **イ**, b - **ウ**, c - **イ**,  
d - **ウ**
- 設問 2 e - **イ**, f - **カ**