次の**問9**から**問13**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，答案用紙の選択欄の(選)をマークして解答してください。

なお，2問以上マークした場合には，はじめの1問について採点します。

問9　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 中にあって OldFile 中にない利用者 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。

② OldFile 中にあって NewFile 中にない利用者 ID の場合

利用者 ID，利用者名の後に“利用者 ID 削除”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 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。

③ NewFile 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者 ID の場合

利用者 ID，利用者名の後に“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

利用者 ID	利用者名	属性	最終使用日
AE001	UserE1	41	20140419
AE002	UserE2	48	20141014
AE003	UserE3	48	20140716
AP005	UserP5	46	20141015
AP006	UserP6	44	00000000

ファイル OldFile のデータ例

利用者 ID	利用者名	属性	最終使用日
AE001	UserE1	40	20140419
AE002	UserE2	44	20140712
AE003	UserE3	4C	20140716
AP004	UserP4	45	20140715
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

利用者 ID	利用者名	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	利用者 ID 削除　特権 O 解除
AP006	UserP6	利用者 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 & BitO) > (OldAttr & BitO))
              printf("  特権 O 付加");
           if ((NewAttr & BitO) < (OldAttr & BitO))
              printf("  特権 O 解除");
        }
        [      c      ]
α    }
     else {
        if (      d      ) {
           printf("\n%-8s  %-10s  利用者 ID 追加", NewID, NewName);
           if ((NewAttr & BitS) == BitS)  printf("  特権 S 付加");
           if ((NewAttr & BitO) == BitO)  printf("  特権 O 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 & BitO) == BitO)  printf("  特権 O 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 + BitO)) != 0x00

イ　((NewAttr | OldAttr) & (BitS + BitO)) != 0x00

ウ　(NewAttr & (BitS + BitO)) != (OldAttr & (BitS + BitO))

エ　(NewAttr | (BitS + BitO)) != (OldAttr | (BitS + BitO))

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 + BitO)) != 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。

(1)　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。

(2)　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。

①　NewFile 中にあって，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。

② NewFile 及び OldFile の両方にあって，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）

(3) 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

エ　strcmp(NewID, OldID) == 0

オ　strcmp(NewID, OldID) == 0 && (NewAttr & BitR) == BitR

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