

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。

[プログラムの説明]

入退室が制限されているエリア（以下、制限エリアという）の入退室の状況を記録したファイルを読み込んで、入退室状況を印字するプログラムである。

X社では、ICカードを用いた入退室システムを導入して、制限エリアの入退室を管理している。図1に、X社の制限エリアのレイアウトを示す。ドアは5か所（ドア番号11, 12, 21, 22, 31）あり、各ドアの内外にカードリーダーが設置されている。入室時と退室時には、カードリーダーにICカードをかざしてドアを開錠する。各ドアのカードリーダーにICカードをかざす都度、その入退室の状況を記録した1行の入退室レコードが入退室ログに書き込まれる。

制限エリアには、機密に応じたレベルが設定されている。レベルは、外部からその部屋へ入るまでに開錠して通過する必要があるドアの数で表す。ドア番号は、10の位がレベルを表す。入退室できるレベルは、ICカードごとに設定されている。

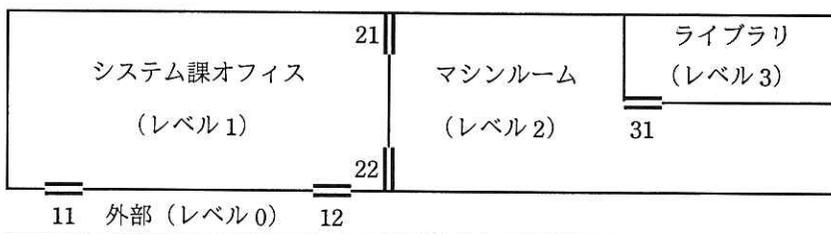


図1 X社の制限エリアのレイアウト

(1) 入退室レコードは33桁の固定長の文字列で、次に示す項目から成る。

項目名	カードID	日付	時刻	ドア番号	入退	可否	名前	改行
桁数	4桁	4桁   2桁   2桁	2桁   2桁   2桁	2桁	1桁	1桁	10桁	1桁
データ例	S001	2015   10   18	09   00   00	11	I	A	Sys Op Lib	\n

- ① カード ID は、IC カードを識別する ID であり、英数字から成る。
  - ② 日付と時刻は、カードリーダーに IC カードをかざした日付と時刻である。
  - ③ 入退は、“I”（入室）又は“O”（退室）である。
  - ④ 可否は、“A”（開錠）又は“R”（拒否）である。
  - ⑤ 名前は、IC カードの使用者を識別する名前であり、英数字と空白から成る。
  - ⑥ レコードの終端には、改行文字が付いている。
- (2) 入退室ログから印字したい日付・時間帯の入退室レコードを抽出し、カード ID、日付及び時刻（先頭の 18 桁）で昇順に整列したファイルを Access.Log とする。図 2 に、Access.Log のレコードの例を示す。

```
S001;20151018;090000;11;I;A;Sys Op Lib
S001;20151018;091004;21;I;A;Sys Op Lib
S001;20151018;092008;31;I;A;Sys Op Lib
S001;20151018;093012;31;O;A;Sys Op Lib
S001;20151018;094016;21;O;A;Sys Op Lib
S001;20151018;095020;11;O;A;Sys Op Lib
V001;20151018;110048;12;I;A;Visitor 01
V001;20151018;111052;22;I;R;Visitor 01
V001;20151018;112056;12;O;A;Visitor 01
```

注記 破線は項目の区切りを示す。  
改行文字は表示していない。

図 2 Access.Log のレコードの例

- (3) Access.Log から読み込んだ 1 レコードごとに、カード ID、名前、日時、ドア番号、入退、可否から成る 1 行を、図 3 に示す様式で印字する。日時以降の項目は、IC カードをかざしたドアのレベルに応じて 19, 39, 59 桁目のいずれかから印字する。

1 ...	19 ...	39 ...	59 ...
カード ID 名前	レベル 1	レベル 2	レベル 3
S001 Sys Op Lib	10-18 09:00 11 IN	10-18 09:10 21 IN	10-18 09:20 31 IN
			10-18 09:30 31 OUT
		10-18 09:40 21 OUT	
	10-18 09:50 11 OUT		
V001 Visitor 01	10-18 11:00 12 IN	10-18 11:10 22 (R)IN	
	10-18 11:20 12 OUT		

注記 見出し行は、あらかじめ印字されているものとする。

図 3 図 2 のレコードの例を用いた印字結果

① カード ID 及び名前は、同一カード ID が続くとき、先頭の行だけに印字する。

② 日時は、“月-日時:分”の形式で印字する。

③ 入退は、“1”なら“IN”と、“0”なら“OUT”と印字する。

④ 可否は、“A”なら何も印字せず、“R”なら“IN”又は“OUT”の直前に“(R)”を印字する。

(4) 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。また、ライブラリ関数 `strcpy(s1, s2)` は、文字列 `s2` を `s1` に複写する。

[プログラム]

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

FILE *logFile;
int logEOF = 0;
char cardID[5] = "----", date[9] = "-----", time[7] = "-----",
    door[3] = "--", dir[2] = "-", act[2] = "--",
    name[11] = "-----";
① → char lastID[5] = "----";

void getRecord();
② → void putRecord();

void main() {
    logFile = fopen("Access.Log", "r");
    getRecord();
    while (logEOF != EOF) {
③ →     putRecord();
④ →     getRecord();
⑤ →     }
    fclose(logFile);
}

void getRecord() {
    if (fscanf(logFile, "%4c%8c%6c%2c%1c%1c%10c\n",
                cardID, date, time, door, dir, act, name) == EOF)
        a ;
    /* ファイルの終わりに達したとき fscanf は EOF (負の整数定数) を返す */
}
```

```

void putRecord() {
    int putSpace;

    if (strcmp(cardID, lastID) == 0)
        printf("%18s", " ");
    else {
        printf("\n%4s %10s ", cardID, name);
        ⑥ →  ;
        ⑦ → }
        putSpace =  ;
        while (0 < putSpace--) {
            printf("%20s", " ");
        ⑧ → }
        printf("%.2s-%.2s %.2s:%.2s %2s ",
            date+4, date+6, time, time+2, door);
        if (strcmp(act, "R") == 0) printf("%s", "(R)");
        if (strcmp(dir, "I") == 0) printf("%s\n", "IN" );
        else printf("%s\n", "OUT");
    }
}

```

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, bに関する解答群

- |                       |                          |
|-----------------------|--------------------------|
| ア cardID = lastID     | イ lastID = cardID        |
| ウ logEOF = 0          | エ logEOF = EOF           |
| オ strcpy(cardID, " ") | カ strcpy(cardID, lastID) |
| キ strcpy(lastID, " ") | ク strcpy(lastID, cardID) |

cに関する解答群

- |                     |                 |
|---------------------|-----------------|
| ア door[0] - '0' - 1 | イ door[0] - '0' |
| ウ door[0] - '0' + 1 | エ door[0] - '3' |

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。

プログラムを変更して、入退室順序の整合性を検査する処理を追加する。

各入退室者は、最初は外部（レベル 0）にいて、最後は外部に出るものとする。正しく入退室していれば、制限エリアのレベルは 0 から始まり、ドアを通過する都度、レベルが 1 ずつ変化し、最後は 0 に戻るはずである。しかし、抽出した日付・時間帯よりも前（後）に入室（退室）している場合や、IC カードをかざさずに人の後についてドアを通過した場合などには、順序の不整合が起きる。

なお、カードリーダーに IC カードをかざしてドアが開錠した場合は、必ず入退室するものとする。

- (1) 順序の不整合が起きた場合は、本来入退室の記録があるべき行のレベル 1 の印字位置に “\*\*\*\*\* Level x-->y” と印字する。これは、入退室者がこの前後にレベル  $x$  から  $y$  へ移動したはずであるが、その記録がないことを意味する。
- (2) 図 4 に、不整合がある入力レコードの例を示す。図 4 のレコードを用いて変更後のプログラムを実行すると、印字結果は、図 5 のようになる。

T001	20151018	100024	11	I	A	Test Use 1
T001	20151018	102028	31	I	A	Test Use 1
T001	20151018	104032	21	O	A	Test Use 1

注記 破線は項目の区切りを示す。  
改行文字は表示していない。

図 4 不整合がある入力レコードの例

1 ...		19 ...	39 ...	59 ...
カード ID 名前	レベル 1	レベル 2	レベル 3	
T001 Test Use 1	10-18 10:00 11 IN			
	***** Level 1-->2			
				10-18 10:20 31 IN
	***** Level 3-->2			
		10-18 10:40 21 OUT		
	***** Level 1-->0			

注記 見出し行は、あらかじめ印字されているものとする。

図 5 不整合がある入力レコードの例（図 4）のときの印字結果

プログラムを、次のように変更する。

- (1) プログラムに、次の文を追加する。

追加位置	追加する文
行①の直後	int level = 0;
行②の直後	void checkLevel(); void clearLevel();
<input type="text" value="d"/> の直後	checkLevel();
<input type="text" value="e"/> の直後	clearLevel();

注記 level は、入退室者が現在いる制限エリアのレベルを表す変数である。

- (2) プログラムの末尾に、次の二つの関数を追加する。

```
void checkLevel() {
    int afterLevel, beforeLevel, doorLevel;

    doorLevel = door[0] - '0';
    beforeLevel = doorLevel;
    afterLevel = doorLevel;
    if (strcmp(dir, "I") == 0)
        beforeLevel = doorLevel - 1;
    else
        afterLevel = doorLevel - 1;
    if (strcmp(act, "R") == 0)
        afterLevel = beforeLevel;
    if (level != beforeLevel)
        printf("***** Level %d-->%d\n%18s",
               , , " ");
    level = afterLevel;
}

void clearLevel() {
    if (logEOF == EOF || (strcmp(cardID, lastID) != 0))
        if (level > 0) {
            printf("%18s***** Level %d-->0\n", " ", level);
            level = 0;
        }
}
```

d, eに関する解答群

ア 行③

イ 行④

ウ 行⑤

エ 行⑥

オ 行⑦

カ 行⑧

f, gに関する解答群

ア afterLevel

イ beforeLevel

ウ doorLevel

エ level

C