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1～3に答えよ。

二つの英単語を辞書順で比較する関数 diccmp を作成した。

一般に，英大文字，英小文字，記号文字を含む英単語を一定の順に並べる方法として，次の例のように，文字列中の文字コード順に並べる方法の他に，英和辞典や書籍の欧文索引のような順序で並べる方法（以下，辞書順という）がある。

例　4個の単語“A.M.”，“CPU”，“all”及び“best”を並べる方法

文字コード順：“A.M.” ＜ “CPU” ＜ “all” ＜ “best”

辞書順：　　“all” ＜ “A.M.” ＜ “best” ＜ “CPU”

検索などの場合は，英単語を辞書順に整列しておくと検索の効率がよく，精度も向上する。

関数 diccmp は，英大文字，英小文字，“-”及び“.”を含む二つの英単語の大小を辞書順で比較する。

〔プログラムの説明〕

関数 diccmp は，二つの英単語の大小を辞書順で比較する。その引数と返却値は，次のとおりである。

引数：	word1, word2	英単語を格納した文字列	
返却値：	負の値	word1＜word2	（例：“map” ＜ “May”）
	0	word1＝word2	（例：“Mr.” ＝ “Mr.”）
	正の値	word1＞word2	（例：“U.S.” ＞ “US”）

(1) 引数 word1 と word2 は，いずれも次の条件を満たしている。

① 各文字は，英大文字（“A” ～ “Z”），英小文字（“a” ～ “z”），“-” 又は “.” のいずれかである。英大文字と英小文字を合わせて，英字という。

② 文字列の長さは，1 以上 30 以下である。

③ 文字列の先頭の文字は，英字である。

④ 文字列中の “-” 及び “.” の直前の文字は，英字である。

C

(2) 辞書順での比較の方法は，次のとおりである。

① 各引数の文字列から，基本文字列と文字情報列を生成する。基本文字列とは，引数の文字列から英字だけを順に取り出して，大文字を小文字に変換した文字列である。文字情報列とは，基本文字列中の各文字に対応する 6 種類の文字情報（大文字，小文字，大文字 “-” 付き，小文字 “-” 付き，大文字 “.” 付き，小文字 “.” 付き）を表す情報の列である。ここで，“-” 及び “.” は，直前の英字に属する文字情報とみなす。

例 引数の文字列：“Fri.” → 基本文字列：

f	r	i

文字情報列：

大文字	小文字	小文字 “.” 付き

② 各引数の基本文字列同士を関数 strcmp で比較し，異なっていれば，その返却値（負の値 又は 正の値）を返す。一致していれば，各引数の文字情報列同士を関数 strcmp で比較し，その返却値（0，負の値 又は 正の値）を返す。

③ 6 種類の文字情報について，その値の大小関係はプログラム中で定めている。

(3) プログラム中で使用しているライブラリ関数の概要は，次のとおりである。

isalpha(c)：c が英字のとき 0 以外の値を返し，それ以外のとき 0 を返す。
islower(c)：c が英小文字のとき 0 以外の値を返し，それ以外のとき 0 を返す。
strcmp(s1, s2)：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。
tolower(c)：c が英大文字のときその文字に対応する英小文字を返し，それ以外のとき c を返す。

〔プログラム〕

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

int  diccmp(char *, char *);
void diccnv(char *, char *, char *);

int diccmp(char *word1, char *word2) {
    int  rc;

    char char1[31], char2[31], attr1[31], attr2[31];
    diccnv(word1, char1, attr1);
    diccnv(word2, char2, attr2);
    rc = strcmp(char1, char2);
    if (rc == 0)
        rc = strcmp(attr1, attr2);
    return rc;
}

void diccnv(char *wordx, char *charx, char *attrx) {
    int  ch, cpos, wpos;

    cpos = 0;
    wpos = 0;
    while (wpos < 30 && (ch = wordx[wpos++]) != '\0') {
        if (isalpha(ch)) {
            if (islower(ch)) {
                charx[cpos] = ch;
                attrx[cpos] = '0';
            }
            else {
                charx[cpos] = tolower(ch);
                attrx[cpos] = '4';
            }
            ch = wordx[wpos];
            if (ch == '-' || ch == '.') {
                if (ch == '-')
                    attrx[cpos] += 1;
                else
                    attrx[cpos] += 2;
                wpos++;
            }
            cpos++;
        }
    }
    charx[cpos] = '\0';
    attrx[cpos] = '\0';
}
```

設問1　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。

引数 word1 の内容を“A.D.”，word2 の内容を“ad-”として，関数 diccmp を実行した。関数 diccmp 中の return 文を実行する時点で，attr1 の内容は“［　a　］”，attr2 の内容は“［　b　］”，rc の内容は［　c　］となる。

a，b に関する解答群

ア　01	イ　02	ウ　11	エ　22
オ　41	カ　42	キ　55	ク　66

c に関する解答群

ア　0	イ　正の値	ウ　負の値

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。

関数 diccmp を用いて，4 個の単語“CO”，“Co.”，“co-”及び“co.”を相互に比較したとき，その大小関係は，［　d　］となる。

d に関する解答群

ア　“CO” ＜ “Co.” ＜ “co-” ＜ “co.”	イ　“CO” ＜ “Co.” ＜ “co.” ＜ “co-”
ウ　“Co.” ＜ “CO” ＜ “co-” ＜ “co.”	エ　“co-” ＜ “co.” ＜ “Co.” ＜ “CO”
オ　“co.” ＜ “co-” ＜ “CO” ＜ “Co.”	カ　“co.” ＜ “co-” ＜ “Co.” ＜ “CO”

設問3　次の表1中の［　　］に入れる正しい答えを，解答群の中から選べ。

このプログラムは，引数 word1 及び word2 がプログラムの説明中の (1) に示した条件①～④を満たしているものとして作成している。しかし，引数が条件を満たさない場合のプログラムの動作についても確認しておきたい。そこで，引数が条件を満たさない場合のプログラムの動作を，表1にまとめた。

なお，文字列中の文字は，全て1バイト文字とする。

表1　引数が条件を満たさない場合のプログラムの動作

条件	条件を満たさない場合	プログラムの動作
①	文字列中に，英字，“-”，“.” 以外の文字がある。	
②	文字列の長さが0である。	[e]。
	文字列の長さが31以上である。	[f]までを有効とし，後続の文字を無視する。
③	先頭の文字が英字でない。	[g]。
④	“-” 及び “.” の直前の文字が英字でない。	

注記　網掛けの部分は表示していない。

eに関する解答群

ア　空文字列として扱い，プログラムは正常に終了する

イ　配列に何も値を設定せずに比較をするので，予期できない結果となる

ウ　配列の定義範囲外への書込みが発生するので，予期できない結果となる

エ　プログラムが終了しない

fに関する解答群

ア　30個目の英字

イ　30個目の英字（直後の文字が “-” 又は “.” の場合はその文字）

ウ　30文字目

エ　30文字目（30文字目が英字で31文字目が “-” 又は “.” の場合は31文字目）

gに関する解答群

ア　その文字が “-”，“.” 以外の場合はその文字を無視するが，“-” 又は “.” の場合は配列の定義範囲外への書込みが発生するので，予期できない結果となる

イ　その文字を無視する

ウ　配列の定義範囲外への書込みが発生するので，予期できない結果となる

エ　プログラムが終了しない