

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。

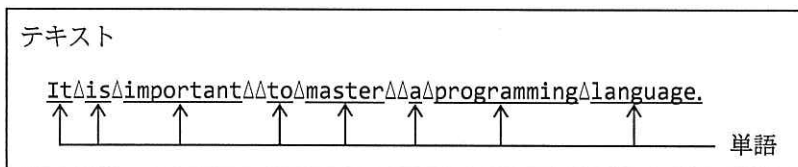
[プログラムの説明]

関数 `format_text` は、印字したときに単語が行末で切れないようにテキストを編集してファイルに出力するプログラムである。

(1) テキストに含まれる文字は、次のものである。

- ① 英字 A～Z, a～z
- ② 数字 0～9
- ③ 記号 !"#\$%&'()*+,-./:;<=>?[]^_`{|}~
- ④ 空白文字
- ⑤ 改行文字

(2) 単語は、空白文字及び改行文字を含まない文字列であり、空白文字又は改行文字で区切られている。単語の文字数は20以下とする。図1に、テキストに含まれる単語の例を示す。



注記 “Δ” は空白文字を表す。

図1 テキストに含まれる単語の例

(3) 改行文字を除き、1行分として印字できる文字数(以下、最大文字数という)をプログラムの引数で与えるが、その値は40以上とする。単語の途中で最大文字数を超える場合は、単語の直前に改行文字を出力し、その単語が次の行の先頭に印字されるようにする。

(4) テキスト中の空白文字及び改行文字は、そのまま出力する。ただし、連続する

[プログラム]

(行番号)

```
1 #include <stdio.h>
2 #define WLEN_MAX 20 /* 単語の最大長 */
3 void format_text(char *, char *, int);
4 void format_text(char *in_file, char *out_file, int width) {
5     FILE *ifp, *ofp;
6     int ch, /* 入力した文字 */
7         lpos = 0, /* 出力処理をしている行での出力済み文字数 */
8         sp = 0; /* strに格納されている文字数 */
9     char str[WLEN_MAX + 2]; /* 単語を含む出力用文字列 */
10
11     ifp = fopen(in_file, "r");
12     ofp = fopen(out_file, "w");
13
14     while ((ch = fgetc(ifp)) != EOF) {
15         if (ch == '\n') {
16             str[sp++] = ch;
17             str[sp] = '\0';
18             fputs(str, ofp);
19             a;
20             sp = 0;
21         } else if (ch == ' ') {
22             lpos += sp;
23             if (lpos >= width) {
24                 str[sp++] = b;
25                 lpos = 0;
26             }
27             str[sp] = '\0';
28             fputs(str, ofp);
29             fputc(ch, ofp);
30             lpos++;
31             c;
32         } else {
33             if (d) {
34                 fputc('\n', ofp);
35                 lpos = 0;
36             }
37             str[sp++] = ch;
38         }
39     }
40     str[sp] = '\0';
41     fputs(str, ofp);
42
43     fclose(ifp);
44     fclose(ofp);
45 }
```

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 `lpos = 0` イ `lpos = sp` ウ `lpos = width`
エ `lpos++` オ `lpos += sp`

bに関する解答群

- ア `'\0'` イ `'\n'` ウ `' '` エ `ch`

cに関する解答群

- ア `sp = 0` イ `sp = lpos` ウ `sp++`
エ `str[lpos] = ch` オ `str[sp++] = ch`

dに関する解答群

- ア `lpos > 0` イ `lpos >= width`
ウ `lpos > sp` エ `sp > 0`
オ `(lpos + sp) > 0` カ `(lpos + sp) >= width`

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。

テキストの編集処理のうち、行頭の空白文字に関する処理を変更することになった。これに合わせて関数 `format_text` に、処理を追加する。ここで、プログラム中の a ~ d には正しい答えが入っているものとする。

(1) テキストの編集処理の変更内容は、次のとおりである。

- ① 入力テキスト中の1文字以上の連続する空白文字列で、印字したときに行頭に来る部分は出力しない。
- ② 最初に出力する単語の場合、又は行の先頭に来る単語の場合で直前に出力した単語の最後の文字が“.”であったときは、単語の直前に空白文字を一つ出力する。

(2) 処理を変更したプログラムを実行したときは、図3に示すようになる。

(入力テキスト)
The Information Technology Engineers Examination was first administered in 1969. In 1970, it became a national examination. Since its commencement, the examination has played an important role in the development of IT engineers.
(出力テキスト)
The Information Technology Engineers Examination was first administered in 1969. In 1970, it became a national examination. Since its commencement, the examination has played an important role in the development of IT engineers.
-----+-----+-----+-----+-----+ 1 10 20 30 40 50 印字したときの文字位置

注記 “Δ” は空白文字を, “↓” は改行文字を表す。

図3 処理を変更し, 最大文字数を 50 とした場合の例

(3) 処理の変更に対応するために, プログラムを表1のとおりに変更する。

表1 プログラムの変更内容

処置	変更内容
行番号 8 と 9 の間に追加	<code>int lch = '.';</code>
行番号 26 と 27 の間に追加	<code>if (<input type="text" value="e"/>) {</code>
行番号 28 と 29 の間に追加	<code>}</code>
行番号 34 と 35 の間に追加	<code>if ((lpos == 0) && (sp == 0) && (lch == '.')) { fputc(' ', ofp); lpos++; } <input type="text" value="f"/>;</code>

eに関する解答群

ア `lpos == 0`

イ `lpos == sp`

ウ `lpos != 0`

エ `lpos != sp`

オ `sp == 0`

カ `sp != 0`

fに関する解答群

ア `lch = ' '`

イ `lch = '.'`

ウ `lch = ch`

エ `str[sp++] = ' '`

オ `str[sp++] = lch`

C