

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。

入力ファイルの内容を、文字及び16進数で表示するプログラムである。

[プログラムの説明]

(1) 関数 dump の引数の仕様は、次のとおりである。

char \*filename 入力ファイルのファイル名

long from 表示を開始するバイト位置

long to 表示を終了するバイト位置 (値が負の場合はファイルの末尾)

ここで、バイト位置は、ファイルの先頭のバイトから順に0, 1, …と数える。

(2) 入力ファイルは、バイナリファイルとして読み込む。入力ファイル中の各バイトの内容(ビット構成)に制約はない。図1に、入力ファイルの例を示す。

(3) 入力ファイル中の各バイトの内容を、文字及び16進数で表示する。図2は図1の入力ファイルの先頭から末尾までの表示例であり、図3は同じファイルのバイト位置17から40までの表示例である。

(4) 表示の様式を、次に示す。説明中の①, ②, …は、図中の網掛け部分を指している。

- ・入力ファイルのバイト位置 from から60バイトずつを、3行1組で表示する。
- ・各組の1行目に各バイトが表す文字を、2行目に各バイトの16進数表示の上位桁を、3行目に同下位桁を、それぞれ表示する。例えば、①のバイトは、文字表示が“i”で、その16進数表示が69である。
- ・バイトの内容が16進数表示で20～7E以外の場合は、そのバイトが表す文字として、②のように“.”を表示する。
- ・各組の1行目の行頭に、その組に表示する最初のバイトのバイト位置を10進数で③の形式で表示する。

・入力ファイルの内容の表示が終わった後、最終行の④の位置には、入力ファイルの終わりに達して終了した場合は“END OF DATA”を、表示を終了するバイト位置に達して終了した場合は“END OF DUMP”を表示する。⑤の位置には、表示した入力ファイルの内容のバイト数を10進数で表示する。

(5) 入力ファイルのファイルサイズ（バイト数）及び引数 from, to の値は、次の式を満たすものとする。

to < 0 の場合：  $to < 0 \leq from < \text{ファイルサイズ} < 2^{31}$

to ≥ 0 の場合：  $0 \leq from \leq to < \text{ファイルサイズ} < 2^{31}$

(6) プログラム中で使用している関数 fgetc(s) は、ストリーム s から 1 文字を読み込んで返す。ファイルの終わりに達しているときは、EOF を返す。

```
int main() {
    /* for testing dump() */
    dump("main.c", 0L, -1L);
    dump("main.c", 17L, 40L);
}
```

注記 各行の行末には、復帰文字(0x0D)及び改行文字(0x0A)がある。

図1 入力ファイルの例

```

③ ① ②
0 int main() {.. /* for testing dump() */.. dump("main.c",
66726666222700222222667276776662676722220022267672266662622
9E40D19E890BDA000FA06F2045349E7045D0890AFDA00045D082D19EE32C

③
60 0L, -1L);.. dump("main.c", 17L, 40L);..}..
234222342300222676722666626222334223342300700
00CC0D1C9BDA00045D082D19EE32C017CC040C9BDADDA

④ ⑤
END OF DATA ... 105 byte(s)
```

図2 図1の入力ファイルの先頭から末尾までの表示例

```

③
17 /* for testing dump() */
222667276776662676722222
FA06F2045349E7045D0890AF

④ ⑤
END OF DUMP ... 24 byte(s)
```

図3 図1の入力ファイルのバイト位置17から40までの表示例

[プログラム]

```
#include <stdio.h>

#define WIDTH 60 /* 行当たり表示バイト数 */
#define MASKCHR '.' /* 16進数表示で20~7E以外の場合の表示用文字 */

void dump(char *filename, long from, long to) {
    FILE *infile;
    int chr, pos = 0;
    long cnt = 0;
    char tblC[256], bufC[WIDTH + 1];
    char tblH[256], bufH[WIDTH + 1];
    char tblL[256], bufL[WIDTH + 1];
    char hex[] = "0123456789ABCDEF";

    for (chr = 0x00; chr <= 0xFF; chr++) {
        if ((0x20 <= chr) && (chr <= 0x7E))
            tblC[chr] = chr;
        else
            tblC[chr] = MASKCHR;
        tblH[chr] = hex[chr >> 4];
        tblL[chr] = hex[];
    }
    bufC[WIDTH] = bufH[WIDTH] = bufL[WIDTH] = '\0';
    infile = fopen(filename, "rb");
    while (((chr = fgetc(infile)) != EOF)
            && ( < 0 &&  <= to)) {
        cnt++;
        if () {
            bufC[pos] = tblC[chr];
            bufH[pos] = tblH[chr];
            bufL[pos] = tblL[chr];
            pos++;
            if () {
                printf("%10ld %s\n%12s%s\n%12s%s\n\n",
                       cnt - WIDTH, bufC, " ", bufH, " ", bufL);
                pos = 0;
            }
        }
    }
}
```

```

if (pos > 0) {
    bufC[pos] = bufH[pos] = bufL[pos] = '\0';
    printf("%10ld %s\n%12s%s\n%12s%s\n",
           cnt - pos, bufC, " ", bufH, " ", bufL);
}
if (chr == EOF)
    printf("END OF DATA ... %ld byte(s)\n", cnt - from);
else
    printf("END OF DUMP ... %ld byte(s)\n", cnt - from);
fclose(infile);
}

```

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。

ここで、b1 と b2 に入れる答えは、b に関する解答群の中から組合せとして正しいものを選ぶものとする。

a に関する解答群

- |               |               |
|---------------|---------------|
| ア chr & 0x0F  | イ chr & 0xF0  |
| ウ chr && 0x0F | エ chr && 0xF0 |

b に関する解答群

	b1	b2
ア	&&	&&
イ	&&	
ウ		&&
エ		

c に関する解答群

- |              |               |                   |
|--------------|---------------|-------------------|
| ア cnt > from | イ cnt >= from | ウ cnt >= from - 1 |
|--------------|---------------|-------------------|

d に関する解答群

- |                    |                |                    |
|--------------------|----------------|--------------------|
| ア cnt == WIDTH - 1 | イ cnt == WIDTH | ウ cnt == WIDTH + 1 |
| エ pos == WIDTH - 1 | オ pos == WIDTH | カ pos == WIDTH + 1 |

設問2 関数 dump の動作に関する次の記述中の  に入れる正しい答えを、  
解答群の中から選べ。

表示結果の最終行（表示が 1 行だけの場合はその行）の表示内容について、  
次の二つのケースを考える。

[ケース1] ファイルサイズ = 100, from = 99, to = 99

この場合、最終行の表示内容は “  e  ” となる。

[ケース2] ファイルサイズ = 0, from = 0, to < 0

この場合、ファイルサイズ及び from の値がプログラムの説明 (5) の条件を満たしていない。このケースについて関数 dump を実行すると、最終行の表示内容は “  f  ” となる。

e, fに関する解答群

ア END OF DATA ... -1 byte(s)

イ END OF DUMP ... -1 byte(s)

ウ END OF DATA ... 0 byte(s)

エ END OF DUMP ... 0 byte(s)

オ END OF DATA ... 1 byte(s)

カ END OF DUMP ... 1 byte(s)