第56回

C言語プログラミング能力認定試験

2　級

正答・解説

サーティファイ
情報処理能力認定委員会

＜解説＞

問１

(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)
イ	ア	ア	イ	ア	ア	イ	イ

(1) 標準ライブラリ関数 rand は，ヘッダ stdlib.h で宣言されている。
(4) s1 の文字列が s2 の文字列より大きい場合，「strcmp(s1, s2)」の返却値は 0 より大きい整数となる。
(7) 論理 OR 演算子（||）を使用して複数の条件式を記述した場合，真偽の判定は左から右へ順に行われ，一つでも真と判定されると，それ以降の条件式は判定されない。
(8) ヘッダファイルの指定には前処理指令である#include を使用するが，ヘッダファイルの中に，さらに#include を使用することができる。

問２

(9)	(10)	(11)	(12)	(13)	(14)
ア	イ	エ	エ	ウ	エ

(9) sizeof 演算子は，演算対象として指定したデータ型(空欄 9)や変数，配列などのメモリ上のサイズをバイト数で求める。
(10) sizeof(int)は，メモリ上におけるデータ型 int のサイズ（バイト数）である。int 型のサイズは 32 ビット(4 バイト)であることから，標準出力には 4 と出力される。
(11) sizeof data[2]は，メモリ上における配列 data の要素 data[2]のサイズ（バイト数）である。配列 data の各要素は int 型であるから，標準出力には 4 と出力される。
(12) sizeof data は，メモリ上における配列 data のサイズ（バイト数）である。配列 data は要素 4 の int 型の配列(4 バイト×4＝16 バイト)であることから，標準出力には 16 と出力される。
(13) strlen(mountain)は，文字列 mountain の文字数を返すので，標準出力には 4 と出力される。
(14) sizeof mountain / sizeof(char)は，配列 mountain （char 型）のメモリ上の全サイズを，char 型のメモリ上のサイズで割った値である。すなわち，配列 mountain の要素数と等しくなる。ここで，配列 mountain の要素数は {'F', 'u', 'j', 'i', '¥0'} の 5 であることから，標準出力には 5 と出力される。

問３

(15)	(16)	(17)	(18)	(19)
イ	オ	オ	エ	ウ

(15)

	16 進数	2 進数
a	5afd	0101 1010 1111 1101
b	e439	1110 0100 0011 1001
a & b	4039	0100 0000 0011 1001

(16)

	16 進数	2 進数
a	5afd	0101 1010 1111 1101
b	e439	1110 0100 0011 1001
a | b	fefd	1111 1110 1111 1101

(17)

	16 進数	2 進数
b	e439	1110 0100 0011 1001
~b	1bc6	0001 1011 1100 0110
b ^ ~b	ffff	1111 1111 1111 1111

(18)

	16 進数	2 進数
a	5afd	0101 1010 1111 1101
a << 4	afd0	1010 1111 1101 0000

(19)

	16 進数	2 進数
b	e439	1110 0100 0011 1001
b << 3	21c8	0010 0001 1100 1000

	16 進数	2 進数
a	5afd	0101 1010 1111 1101
b << 3	21c8	0010 0001 1100 1000
a ^ (b << 3)	7b35	0111 1011 0011 0101

問 4

(20)	(21)	(22)	(23)	(24)	(25)
ア	ア	ウ	ウ	ウ	イ

(20) ～ (22) main 関数は，二つの仮引数をもつ場合，次のように定義できる。

```
int main(int argc, char(空欄 20) *argv[])
{
        ⋮
    return 0;
}
```

仮引数 argc には，実行時にコマンド行で指定されたプログラム名と引数の個数の合計値(空欄 21)が格納されて渡され，配列 argv の要素には，プログラム名と各引数の文字列のポインタが格納されて渡される。ここで，argv[argc]には NULL(空欄 22)が格納されている。

(23) argcには，実行時にコマンド行で指定されたプログラム名と引数の個数の合計値が格納されているから，標準出力には 5 と出力される。

(24) argv[2]には，2 番目の引数「AZ」が格納されているから，標準出力には AZ と出力される。

(25) argv[4]には，4 番目の引数「2021」が格納されているから，標準出力には 0 と出力される。

問５

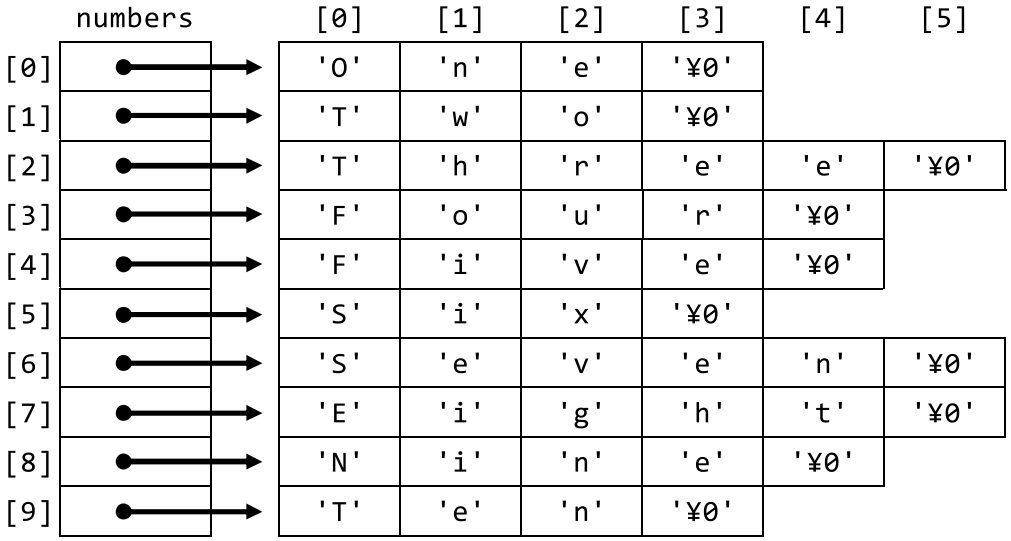
(26)	(27)	(28)	(29)
エ	ウ	ア	ア

プログラムでファイルを使用する際には，fopen(空欄 26) 関数を呼び出し，ファイル名で指定したファイルにストリームを結びつける。fopen(空欄 26) 関数の呼び出しに失敗した場合の戻り値は，NULL(空欄 27) となる。ストリームから 1 文字読み込むには，fgetc(空欄 28) 関数を呼び出す。ストリームがファイルの終わりに達している場合，fgetc (空欄 28) 関数は EOF(空欄 29) を返却する。

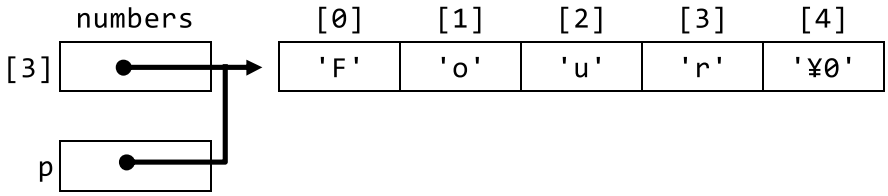
問６

(30)	(31)	(32)	(33)	(34)
イ	オ	ア	イ	オ

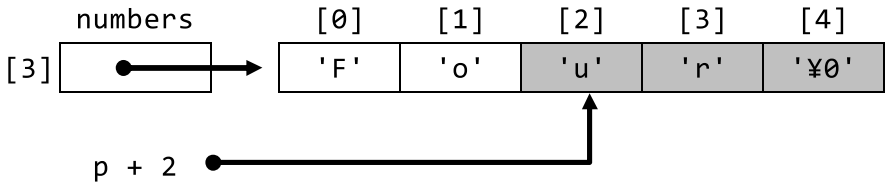
char型ポインタを要素に持つ配列numbersの各要素は以下のように初期化される。



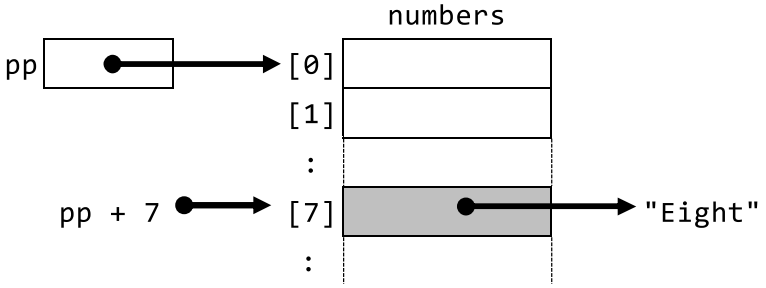
(30)「p = numbers[3]」により，p は"Four"を指し示すことになる。したがって，標準出力には「Four」と出力される。



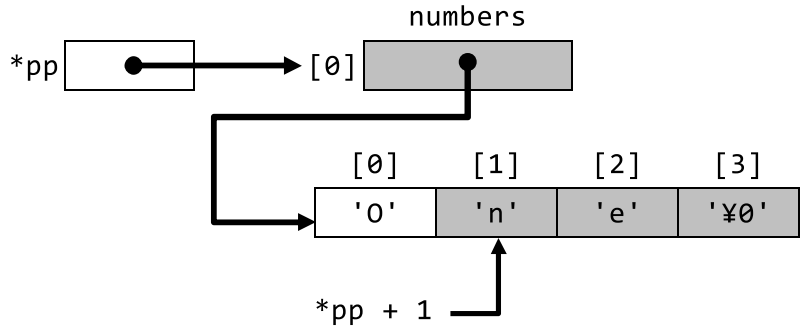
(31)「p + 2」は numbers[3][2]の位置を指し示すことになる。したがって，標準出力には「ur」と出力される。



(32)「pp = numbers」により pp は，配列 numbers の先頭要素を指し示すことになるので「pp + 7」は numbers[7]を指し示す。したがって，「*(pp + 7)」は文字列"Eight"を指し示すので，標準出力には「Eight」と出力される。



(33)「*pp」はnumbers[0]を指し示す。したがって，「*pp + 1」は文字列"One"の2文字目の位置を指し示すので，標準出力には「ne」と出力される。



(34) for文による繰り返し処理において，i = 0のとき，「*pp」はnumbers[0]を指し示しているので，「**pp」は，numbers[0][0]の文字を参照する。同様に「*(pp + 1)」はnumbers[1]を指し示しているので，「**(pp + 1)」は，numbers[1][0]の文字を参照する。for文の繰返し処理内の「pp++」により「*pp」の指し示す配列要素は，numbers[1]，numbers[2]，...と変化する。よって，「**pp == **(pp + 1)」により比較される文字と比較結果は以下のようになり，i = 1のときbreak文により繰り返し処理が終了する。

i	*pp	*(pp + 1)	**pp	**(pp + 1)	比較結果
0	"One"	"Two"	'O'	'T'	偽
1	"Two"	"Three"	'T'	'T'	真

したがって，「*pp」は文字列"Two"を指し示すので，標準出力には「Two」と出力される。

問 7

(35)	(36)	(37)	(38)	(39)
ウ	ウ	イ	エ	ウ

(35) ＜プログラムの説明＞(2)－④の「変換前文字列の末尾文字から先頭文字まで，各文字を参照し，④－１～④－３の処理を繰り返す」反復処理の継続条件である。変換前文字列の末尾文字の要素位置 si は変換前文字列の末尾文字の位置で初期化され，「d = ToNumber(snum[si--])」でデクリメントされる。変換前文字列の先頭文字まで参照する必要があるので，si が-1 になったとき反復処理を終了すればよい。したがって，継続条件は「si >= 0」となる。

(36) ＜プログラムの説明＞(2)－④－１の「対象文字が数値化できなかった場合は，-1 を返却し関数を終了する」ための条件判定である。対象文字が数値化できたとき，すなわち有効な文字であったときは，図１から，関数 ToNumber の返却値が 0 ～（変換前基数 - 1）の範囲であればよく，それ以外の場合は数値化できなかったと

なる。したがって，「d == -1 || d >= sradix」となる。

（37）＜プログラムの説明＞(2)－④－３の「基数重みに変換前基数を乗じた値を，新たな基数重みとする」処理である。したがって「wt *= sradix」となる。

（38）＜プログラムの説明＞(2)－⑥の「変換値が変換後基数以上の間，⑥－１～⑥－２の処理を繰り返す」反復処理の継続条件である。したがって「num >= dradix」となる。

（39）＜プログラムの説明＞(2)－⑥－２の「変換値を変換後基数で割った商を新たな変換値とする」処理である。したがって「num /= dradix」となる。

問８

(40)	(41)	(42)	(43)	(44)
ウ	ア	ア	ウ	ウ

（40）＜関数 ParseRecords の処理の流れ＞(1)「状態を「①H 待ち」に設定する」処理である。したがって「W_HEADER」となる。

（41）＜関数 ParseRecords の処理の流れ＞(case－1)および(case－3)の「メンバーcntを０に初期化」する処理である。したがって「infos[i].cnt = 0」となる。

（42）＜関数 ParseRecords の処理の流れ＞(case－2)の「対象レコードがデータレコードであれば」の判定処理である。したがって「records[j][0] == C_DATA」となる。

（43）＜関数 ParseRecords の処理の流れ＞(case－2)の「トレーラレコードの固有情報（データレコード数）を数値化した値と，情報配列の要素位置のメンバーcnt の値が等しいか」の判定処理である。判定結果が「真」のとき，-2（データレコード数異常）を返却して関数を終了するので，空欄（43）は，「トレーラレコードの固有情報（データレコード数）を数値化した値と，情報配列の要素位置のメンバーcntの値が等しくないとき真となる」条件式である。

したがって「atoi(&records[j][1]) != infos[i].cnt」となる。

（44）＜関数 ParseRecords の処理の流れ＞(4)「状態が「④E 後」でなければ」の判定処理である。判定結果が「真」のとき，-1（レコード構成異常）を返却して関数を終了しているので，空欄（44）は，「状態が「④E 後」でないとき真となる」条件式である。したがって「state != W_AFTER_END」となる。