第60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)
ア	ア	イ	ア	ア	イ	イ	ア

(3) C言語では，宣言指定子 union により共用体を定義することができる。構造体の定義は，宣言指定子 struct により行う。

(6) 列挙型指定子 enum により定義される列挙定数は，整数型で表現可能な値をもつ整数定数式でなければならない。

(7) int 型変数 i の値をキャスト演算子により double 型変数 d に代入する場合，「d = (double)i;」と記述する。

問２

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

& 演算子はビットごとの論理積(空欄 9)を，| 演算子はビットごとの論理和(空欄 10)を，^ 演算子はビットごとの排他的論理和(空欄 11)を，~ 演算子はビットの反転(空欄 12)のビット演算を行うために用いられる。

(13) 変数 a の値は，x の値 0x29AE と y の値 0x705C のビットごとの論理和となる。

x	0	0	1	0	1	0	0	1	1	0	1	0	1	1	1	0	29AE
y	0	1	1	1	0	0	0	0	0	1	0	1	1	1	0	0	705C
x | y	0	1	1	1	1	0	0	1	1	1	1	1	1	1	1	0	79FE

(14) 変数 b の値は，x の値 0x29AE と y の値 0x705C のビットごとの排他的論理和となる。

x	0	0	1	0	1	0	0	1	1	0	1	0	1	1	1	0	29AE
y	0	1	1	1	0	0	0	0	0	1	0	1	1	1	0	0	705C
x ^ y	0	1	0	1	1	0	0	1	1	1	1	1	0	0	1	0	59F2

問３

(15)	(16)	(17)	(18)	(19)
ウ	オ	ア	イ	イ

配列 str は，次のような文字データで初期化されている。

	[0]	[1]	[2]	[3]	[4]	[5]	[6]	[7]	[8]	[9]
str[0]	'C'	'I'	'R'	'C'	'L'	'E'	'¥0'			
str[1]	'S'	'Q'	'U'	'A'	'R'	'E'	'¥0'			
str[2]	'T'	'R'	'I'	'A'	'N'	'G'	'L'	'E'	'¥0'	
str[3]	'P'	'E'	'N'	'T'	'A'	'G'	'O'	'N'	'¥0'	

(15) strlen(str[3]) により，文字列 "PENTAGON" の文字数 8 が変数 len に代入される。したがって，標準出力には「8」と出力される。

(16) strcpy(buf, str[2])により，配列 buf には文字列 "TRIANGLE" が格納される。

	[0]	[1]	[2]	[3]	[4]	[5]	[6]	[7]	[8]
buf	'T'	'R'	'I'	'A'	'N'	'G'	'L'	'E'	'¥0'

strcat(buf, str[1])により，配列 buf の末尾に文字列 "SQUARE" が格納される。

	[0]	[1]	[2]	[3]	[4]	[5]	[6]	[7]
buf	'T'	'R'	'I'	'A'	'N'	'G'	'L'	'E'

	[8]	[9]	[10]	[11]	[12]	[13]	[14]
	'S'	'Q'	'U'	'A'	'R'	'E'	'¥0'

したがって，標準出力には「TRIANGLESQUARE」と出力される。

(17) strcpy(buf, str[0]) により，配列 buf には文字列 "CIRCLE" が格納される。

	[0]	[1]	[2]	[3]	[4]	[5]	[6]
buf	'C'	'I'	'R'	'C'	'L'	'E'	'¥0'

strncpy(buf + 2, str[3], 5) により，"PENTAGON" の先頭から5文字が配列 buf の3文字目から格納された後，buf[7] = '¥0' により要素番号7にナル文字が設定される。

	[0]	[1]	[2]	[3]	[4]	[5]	[6]	[7]
buf	'C'	'I'	'P'	'E'	'N'	'T'	'A'	'¥0'

したがって，標準出力には「CIPENTA」と出力される。

(18) strcmp(str[1], "SQUAR") により，文字列 "SQUARE" と文字列 "SQUAR" が比較される。strcmp の比較結果は非0（不一致）となるので，strncpy(buf, str[1], 2) および buf[2] = '¥0' が実行されて，配列 buf には文字列 "SQ" が格納される。したがって，標準出力には「SQ」と出力される。

（19）strcpy(buf, str[3])により，配列 buf には文字列 "PENTAGON" が格納される。

	[0]	[1]	[2]	[3]	[4]	[5]	[6]	[7]	[8]
buf	'P'	'E'	'N'	'T'	'A'	'G'	'O'	'N'	'¥0'

strchr(str[2], 'G') は，"TRIANGLE" の先頭から文字 'G' を検索して，その位置 str[2][5]を指し示すアドレス値を返却する。

	[0]	[1]	[2]	[3]	[4]	[5]	[6]	[7]	[8]
str[2]	'T'	'R'	'I'	'A'	'N'	'G'	'L'	'E'	'¥0'

よって，strcpy(buf + 5, strchr(str[2], 'G'))は，buf[5]の位置から文字列 "GLE" が格納される。

	[0]	[1]	[2]	[3]	[4]	[5]	[6]	[7]	[8]
buf	'P'	'E'	'N'	'T'	'A'	'G'	'L'	'E'	'¥0'

したがって，標準出力には「PENTAGLE」と出力される。

問４

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

(20) ～ (23) 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(空欄 20) argc, char(空欄 21) *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 22)が格納されて渡される。また，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 23)が格納されている。

(24) argv[1]には，1 番目の引数「-size」が格納されているので，標準出力には-size と出力される。

(25) argv[2]には，2 番目の引数「1024」が格納されているので，標準出力には 4 と出力される。

問５

(26)	(27)	(28)	(29)
ウ	エ	ウ	ア

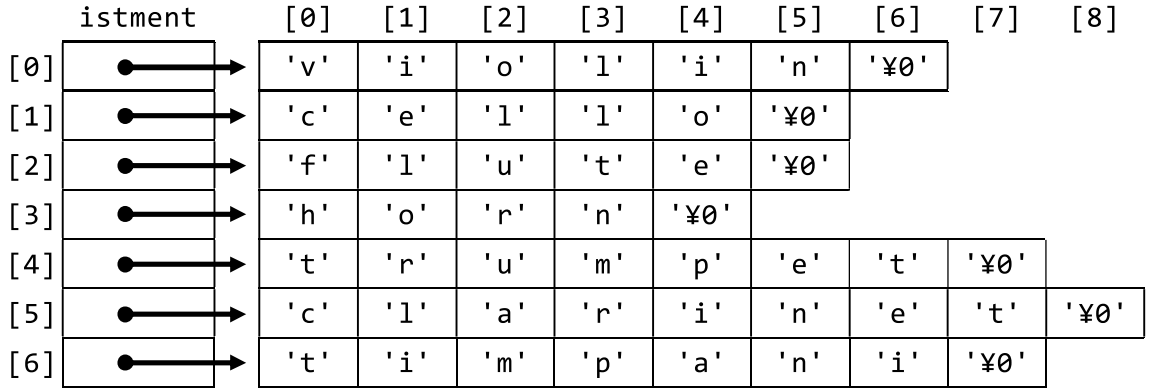
プログラムでファイルを使用する際には，関数fopenを呼び出し，引数で指定したファイル名のファイルをストリームに結び付ける。関数fopenの返却値の型は FILE * (空欄26)で，関数の呼出しに失敗した場合の戻り値は NULL(空欄27)となる。

ファイルに対する処理が終了した後，ストリームとファイルの結び付けを解除するには，関数fopenの返却値 (空欄28)を引数に指定して関数fclose(空欄29)を呼び出す。

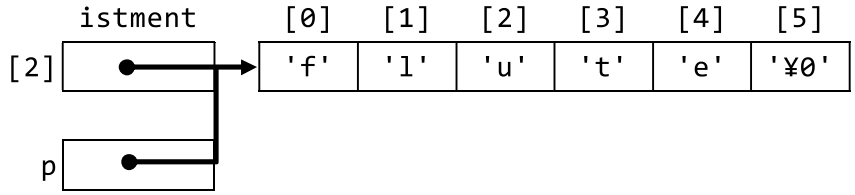
問６

(30)	(31)	(32)	(33)	(34)
イ	オ	オ	ウ	エ

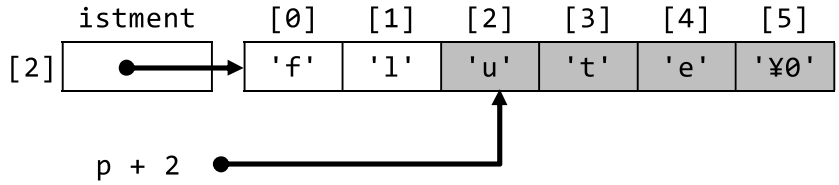
char型ポインタを要素にもつ配列istmentの各要素は以下のように初期化される。



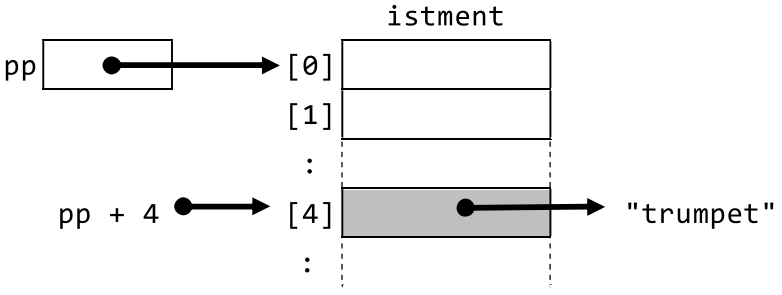
(30)「p = istment[2]」により，p は "flute" を指し示す。したがって，標準出力には「flute」と出力される。



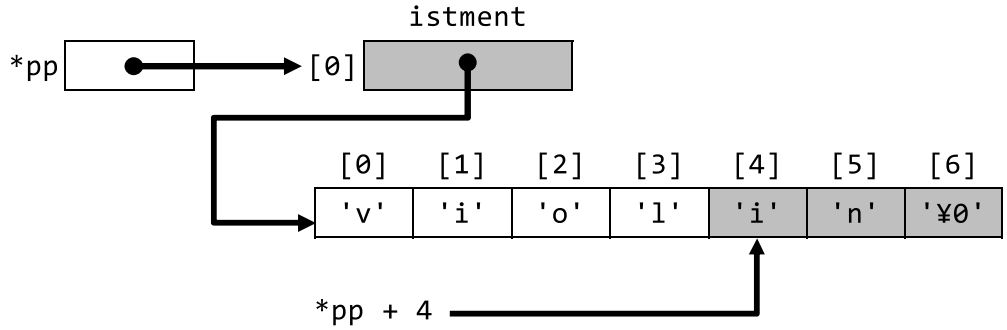
(31)「p + 2」は istment[2][2]の位置を指し示す。したがって，標準出力には「ute」と出力される。



(32)「pp = istment」により，pp は配列 istment の先頭要素を指し示すことになるので，「pp + 4」は istment[4]を指し示す。したがって，「*(pp + 4)」は文字列 "trumpet" を指し示すので，標準出力には「trumpet」と出力される。



(33)「*pp」はistment[0]を指し示す。したがって，「*pp + 4」は文字列"violin"の5文字目の位置を指し示すので，標準出力には「in」と出力される。



(34) 外側のfor文による繰返し処理において，pは順にistment[0]～istment[4]を指し示す。内側のwhile文による繰返し処理において，istment[0]～istment[4]の各文字列の先頭から末尾までの文字を*pで参照し，その文字が 'l' と等しかったらcntをインクリメントしている。すなわち，cntは，"violin"，"cello"，"flute"，"horn"，"trumpet"の文字列内に存在する文字 'l' の数となる。文字 'l' は，"violin" 内に1個，"cello" 内に2個，"flute" 内に1個の計4個存在するので，標準出力には「4」と出力される。

問７

(35)	(36)	(37)	(38)	(39)
ア	エ	ウ	ア	イ

(35) ＜関数QSortの処理＞(2)の「並べ替え対象配列の先頭位置の要素の値を基準値とする」処理である。先頭位置は引数headで渡されるので，先頭位置の要素の値は，「array[head]」となる。

(36) ＜関数 QSort の処理＞(4)の「並べ替え対象配列の先頭位置＋1 の要素位置から末尾位置までの値を参照」する反復処理である。つまり，変数 i を head + 1 に初期化し，i が tail 以下の間，i をインクリメントする反復条件とすればよい。したがって，「i = head + 1; i <= tail; i++」となる。

(37) 空欄(37)の if 文の条件判定が「真」の場合，＜関数 QSort の処理＞(4-1)の「Sl 配列の要素 lp に比較値を格納」している。したがって，条件判定は「比較値が基準値未満の場合」とすればよく，「array[i] < s」となる。

(38) ＜関数 QSort の処理＞(6)の「(5)で複写した“並べ替え対象配列の末尾”の次要素位置（以下，確定位置という）に，基準値を格納する」処理である。実際に，基準値を格納する処理は，空欄(38)の次の処理「array[t] = s」で行われている。よって，空欄(38)では，t に確定位置を代入すればよい。確定位置は，直前の for 文が終了した時点で，head + i に設定されているから，「t = head + i」となる。

(39) ＜関数 QSort の処理＞(7)の「Sr 配列に格納されている全要素を，並べ替え対象配列の確定位置の次の要素位置から順に複写する」ための反復処理である。反復処理の for 文により j は 0 から rp 未満までインクリメントされるから，Sr 配列の要素 sr[j]を，並べ替え対象配列の確定位置の次の要素位置から順に複写するためには，array の要素番号 head + i + j に代入すればよい。したがって，「array[head + i + j] = sr[j]」となる。

問８

(40)	(41)	(42)	(43)	(44)
ア	ア	ウ	イ	エ

(40) ＜関数 IsProhibited の仕様＞（処理内容）の「ch が数字，または，'-'文字であれば 1 を，そうでなければ 0 を返却する」処理である。空欄（40）の判定条件が「真」なら 1 を返却しているので，「isdigit(ch) || (ch == '-')」となる。

(41) ＜関数 Split の仕様＞（処理内容）(4) の「文字位置が最大文字数と等しければ，分割後文字列配列の行位置，文字位置に'¥0'を格納した後，行位置をインクリメントし，文字位置を 0 に設定する」ための判定条件である。文字位置は m に格納されていて，最大文字数は LINESIZE で定義されている。したがって，「m == LINESIZE」となる。

(42) ＜関数 Split の仕様＞（処理内容）(6) の「走査位置の文字から連続する禁則文字の文字数を求める」処理である。j を i + 1（走査位置 + 1）で初期化した後，while 文で in[j]が'¥0'でない間，空欄（42）の判定が「真」のとき while 文を中断，「偽」のとき j をインクリメントしている。つまり，in[j]が禁則文字である間，j をインクリメントし，in[j]が禁則文字以外であったら while 文を中断することにより，禁則文字の文字数は(j - i)となる。したがって，「IsProhibited(in[j]) == 0」となる。

(43) ＜関数 Split の仕様＞（処理内容）(6) の「求めた文字数分の文字を，出力行の末尾に格納すると最大文字数を超えてしまう場合」の判定処理である。出力行の末尾に格納できる残り文字数は LINESIZE - m で求まり，禁則文字の文字数は(j - i)である。したがって，「(LINESIZE - m) < (j - i)」となる。

(44) ＜関数 Split の仕様＞（処理内容）(9)の「分割後文字列の行数を返却する」処理である。外側の while 文が終了した時点で，分割後文字列を出力している行位置は n に保持されている。分割後文字列の行数は，行位置 + 1 で求められるため，「n + 1」となる。