第44回

C言語プログラミング能力認定試験

3　級

正答・解説

サーティファイ
情報処理能力認定委員会

第44回正答　解説

＜解説＞

問１

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

(1) C言語は，特定の計算機のアーキテクチャとは独立であるので，異なるハードウェアで動作させることのできる「移植可能なプログラム」をコーディングすることもできる。

(3) C言語では，注釈（/*と*/で囲まれた文字列）を入れ子にすることはできない。

(6)「`int array[10] = {0, 1, 2};`」と宣言された配列 `array` の要素数は 10 となる。

問２

(9)	(10)	(11)	(12)	(13)
イ	ア	イ	ア	ア

(9) 数値定数 `0500` は 8 進定数の表現であり，10 進数の `320` となる。

(11)「`char s[] = "DEC";`」と等価になる宣言は，「`char s[] = {'D', 'E', 'C', '¥0'};`」である。

問３

(14)	(15)	(16)	(17)	(18)	(19)	(20)	(21)
エ	イ	エ	ア	ウ	オ	イ	エ

(14)，(15) 変数 `i` は `7` に初期化された後，`i--`により，`7`，`6`，`5`，…と 1 ずつ減算される。`for` ループの継続条件が `i > 2` であるから，変数 `i` が `2` となったとき `for` ループを終了する。変数 `a` は `0` に初期化された後，`for` ループの繰返し処理で `data[i]`の値が加算される。したがって，変数 `i` は `2`，変数 `a` は `32` となる。

繰返し回数	1	2	3	4	5	6
i	7	6	5	4	3	2
data[i]	8	7	9	2	6	
a	8	15	24	26	32	

(16)，(17) 変数 `i` は `1` に初期化された後，`i += 2` により，`1`，`3`，`5`，…と `2` ずつ加算される。`for` ループの継続条件が `i < 6` であるから，変数 `i` が `6` 以上になったとき `for` ループを終了する。変数 `a` は `0` に初期化された後，`for` ループの繰返し処理で `data[i - 1]`の値が加算される。したがって，変数 `i` は `7`，変数 `a` は `11` となる。

繰返し回数	1	2	3	4
i	1	3	5	7
data[i - 1]	5	4	2	
a	5	9	11	

(18)，(19) 変数 a は 0 に初期化される。変数 i は 0 に初期化された後，i++により，0，1，2，…と 1 ずつ加算される。for ループの継続条件が i < 6 であるから，変数 i が 6 になったとき for ループを終了する。ただし，data[i] + data[i + 1] < data[i + 2]が「真」になる場合，変数 a に data[i + 2]の値が代入され，そこで for ループを終了する。したがって，変数 i は 3，変数 a は 9 となる。

繰返し回数	1	2	3	4
i	0	1	2	3
data[i]	5	3	4	6
data[i + 1]	3	4	6	2
data[i + 2]	4	6	2	9
a	0	0	0	9

(20)，(21) 外側の for ループでは，変数 i は 0 に初期化された後，i += 2 により，2 ずつ加算される。for ループの継続条件が i < 7 であるから，変数 i が 7 以上になったとき for ループを終了する。一方，内側の for ループでは，変数 j は i + 1 に初期化された後，j++により，1 ずつ加算される。for ループの継続条件が j < 8 であるから，変数 j が 8 になったとき for ループを終了する。変数 a は 0 に初期化された後，内側の for ループの繰返し処理で，data[i] - data[j] > 0 が「真」になる場合，data[i] - data[j]の値が加算される。また，変数 b は 0 に初期化された後，内側の for ループの繰返し処理で，data[i] - data[j] > 0 が「真」になる場合，1 が加算される。したがって，変数 a は 8，変数 b は 4 となる。

繰返し回数	1	2	3	4	5	6	7	
i	0	0	0	0	0	0	0	0
j	1	2	3	4	5	6	7	8
data[i]	5	5	5	5	5	5	5	
data[j]	3	4	6	2	9	7	8	
a	2	3	3	6	6	6	6	
b	1	2	2	3	3	3	3	

繰返し回数	8	9	10	11	12		13	14	15		16		
i	2	2	2	2	2	2	4	4	4	4	6	6	8
j	3	4	5	6	7	8	5	6	7	8	7	8	
data[i]	4	4	4	4	4		2	2	2		7		
data[j]	6	2	9	7	8		9	7	8		8		
a	6	8	8	8	8		8	8	8		8		
b	3	4	4	4	4		4	4	4		4		

問４

(22)	(23)	(24)	(25)	(26)	(27)
オ	イ	ア	ウ	エ	オ

C言語の等値演算子には，二つの値が等しいとき「真」となる `==` と，二つの値が等しくないとき「真」となる `!=`（空欄22）がある。

算術演算子には，`+`(加算)，`-`(減算)，`*`(空欄 23)(乗算)，`/`(除算)，`%`(剰余算）がある。ここで，被除数，除数ともに整数による除算では，小数点以下が切り捨てられた整数値（空欄 24）が商となるため，整数型変数 `a`，`b`（`b ≠ 0`）について，`a % b` の結果は `a - (a / b) * b`（空欄 25）の結果と同値となる。

論理演算子には，`&&`（空欄26）(論理積)，`||`（論理和)，`!`（空欄27）(論理否定)がある。

問５

(28)	(29)	(30)	(31)	(32)
イ	ア	ア	エ	エ

(28) `"%03d"`という書式を指定すると，10 進数の右寄せ 3 桁で出力され，余った桁には`'0'`が出力される。

(29) `"%5o"`という書式を指定すると，8 進数の右寄せ 5 桁で出力され，余った桁には空白が出力される。

(30) `"%6s"`という書式を指定すると，6 文字分の領域に右寄せで出力され，余った桁には空白が出力される。

(31) `"%-3s"`という書式を指定すると，3 文字分の領域に左寄せで出力され，余った桁には空白が出力される。ここで，指定した文字列が 3 文字より長い場合は，文字列の末尾まで出力される。

(32) `"%-7.4s"`という書式を指定すると，文字列の先頭 4 文字が 7 文字分の領域に左寄せで出力され，余った桁には空白が出力される。

問６

(33)	(34)	(35)	(36)	(37)
イ	エ	ア	エ	ウ

(33) ＜処理手順＞(2)の「暗号化前文字列の文字を先頭から末尾まで順に 1 文字ずつ取り出す」ための while 文の継続条件である。while 文の繰返し処理内の「if (instr[i] == tbl[row][col])」から，変数 i は暗号化前文字列を走査するための添字であることが判断でき，繰返し処理の前で 0 に初期化され，繰返し処理の最後の「i++」で走査位置を次の文字に移動している。文字列の走査は，文字列の末尾に格納されている'¥0'が見つかるまで行えばよいから，継続条件は「instr[i] != '¥0'」となる。

(34) ＜処理手順＞(4)の「求めた文字位置の行を行補正値で，列を列補正値で補正する」処理である。空欄直後の「cnvCol = (col + ofsCol) % COLS」により列位置を補正しているのと同様に，行位置を補正した値を変数 cnvRow に代入する。変数 cnvRow に代入する値は，暗号化前文字と一致する変換前テーブル内の行（row）に行補正値（ofsRow）を加算し，補正後の値が配列要素の境界を超えた場合を考慮して，テーブル行数（ROWS）で割った余りを求めればよい。したがって，「(row + ofsRow) % ROWS」となる。

(35) ＜処理手順＞(3)の「一致する文字がない場合は，不正な文字としてその文字を標準出力に表示する」ための判定条件である。暗号化前文字と一致する文字が変換前テーブルにない場合は，「ch = cvt[cnvRow][cnvCol]」による変数 ch への暗号化後文字の代入が行われず初期値（'¥0'）のままとなる。したがって，「ch == '¥0'」となる。

(36) ＜処理手順＞(4)の「変換後テーブルの文字を求め，暗号化後文字列に追加する」処理である。変換後テーブルの文字は，変数 ch に格納されている。また，暗号化後文字列（encstr）に追加する位置は変数 j で管理しており，暗号化後文字列に変数 ch を代入した後インクリメントする。したがって，「encstr[j++] = ch」となる。

(37) ＜処理手順＞(5)の「暗号化後文字列を標準出力に表示する」処理は，「printf("暗号化後文字列 : %s¥n", encstr)」で行っている。ここで，標準関数 printf の"%s"書式指定により文字列として正しく表示するためには，encstr の末尾に文字列の終端を表す文字('¥0')を格納しておかなければならない。したがって，「encstr[j] = '¥0'」となる。