

第45回
C言語プログラミング能力認定試験
3 級

正答・解説

第45回正答 解説

<解説>

問 1

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

- (3) 標準ライブラリ関数 `isxdigit` は、16 進数字かどうかを判定する。
- (5) C 言語では、ループ本体の各実行の後に継続条件の評価を行う `do` 文がある。
- (7) `int` 型変数 `a` に対して、「`a -= 5;`」と等価な記述は「`a = a - 5;`」である。

問 2

(9)	(10)	(11)	(12)	(13)
オ	ア	イ	イ	エ

C 言語における四則演算は、`+`（加算演算子）、`-`（減算演算子）、`*`（空欄 9）（乗算演算子）、`/`（除算演算子）により記述する。また、`++`（インクリメント演算子（空欄 10））と `--`（デクリメント演算子（空欄 11））により、被演算数に `1` を加える、被演算数から `1` を減ずることができる。

- (12) 「`a = i++`」では、変数 `a` に変数 `i` の値「5」が代入された後、変数 `i` の値が `1` 加算されて「6」となる。したがって、変数 `a` には「5」が格納される。
- (13) 「`b = ++i`」では、変数 `i` の値「6」が `1` 加算されて「7」となった後、変数 `b` に変数 `i` の値が代入される。したがって、変数 `b` には「7」が格納される。

問 3

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

- (14) `i = 0` のとき、`weight[i]` の値は `30` なので、「`weight[i] <= 50`」は「真」となり、`price[i]` には `80` が格納される。しかし、その次の「`weight[i] <= 200`」も「真」となるため、`price[i]` には `120` が格納される。三つ目の「`weight[i] > 200`」は「偽」となり、`price[i]` の値は変化しない。したがって、標準出力には `price[0]` の値 `120` が出力される。
- (15) `i = 1` のとき、`weight[i]` の値は `180` なので、「`weight[i] <= 50`」は「偽」となり、`price[i]` の値は初期値の `0` のまま変化しない。その次の「`weight[i] <= 200`」は「真」となり、`price[i]` には `120` が格納される。三つ目の「`weight[i] > 200`」は「偽」となり、`price[i]` の値は変化しない。したがって、標準出力には `price[1]` の値 `120` が出力される。
- (16) `i = 0` のとき、`weight[i]` の値は `30` なので、「`weight[i] <= 50`」は「真」となり、`price[i]` には `80` が格納され、その後の `else if` 以降の処理は実行されない。

したがって、標準出力には `price[0]` の値 80 が出力される。

- (17) `i = 1` のとき、`weight[i]` の値は 180 なので、「`weight[i] <= 50`」は「偽」となり、次の `else if` の条件式が評価される。次の「`weight[i] <= 200`」は「真」となるため、`price[i]` には 120 が格納される。したがって、標準出力には `price[1]` の値 120 が出力される。
- (18) `i = 0` のとき、`age[i][0]` の値は 30、`age[i][1]` の値は 28 なので、「`age[i][0] + age[i][1] >= 100`」は「偽」となり、`else` 句の `for` 文が実行される。
 `j = 0` のとき、「`age[i][j] <= 10`」は「偽」、「`age[i][j] >= 60`」も「偽」となり、`price[i]` には 1000 が加算され、`price[i]` は 1000 となる。`j = 1` のときも、「`age[i][j] <= 10`」は「偽」、「`age[i][j] >= 60`」も「偽」となり、`price[i]` には 1000 が加算され、`price[i]` は 2000 となる。したがって、標準出力には `price[0]` の値 2000 が出力される。
- (19) `i = 1` のとき、`age[i][0]` の値は 7、`age[i][1]` の値は 40 なので、「`age[i][0] + age[i][1] >= 100`」は「偽」となり、`else` 句の `for` 文が実行される。
 `j = 0` のとき、「`age[i][j] <= 10`」は「真」となり、`price[i]` には 300 が加算され、`price[i]` は 300 となる。`j = 1` のとき、「`age[i][j] <= 10`」は「偽」、「`age[i][j] >= 60`」も「偽」となり、`price[i]` には 1000 が加算され、`price[i]` は 1300 となる。したがって、標準出力には `price[1]` の値 1300 が出力される。
- (20) `i = 2` のとき、`age[i][0]` の値は 60、`age[i][1]` の値は 10 なので、「`age[i][0] + age[i][1] >= 100`」は「偽」となり、`else` 句の `for` 文が実行される。
 `j = 0` のとき、「`age[i][j] <= 10`」は「偽」、「`age[i][j] >= 60`」は「真」となるため、`price[i]` には 750 が加算され、`price[i]` は 750 となる。`j = 1` のとき、「`age[i][j] <= 10`」は「真」となり、`price[i]` には 300 が加算され、`price[i]` は 1050 となる。したがって、標準出力には `price[2]` の値 1050 が出力される。
- (21) `i = 3` のとき、`age[i][0]` の値は 58、`age[i][1]` の値は 49 なので、「`age[i][0] + age[i][1] >= 100`」は「真」となり、`price[i]` には 1200 が代入される。したがって、標準出力には `price[3]` の値 1200 が出力される。

問 4

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

C 言語のプログラムをコンパイラで実行形式へ変換する際、`#define` (空欄22) 指令によりマクロを置き換えたり、`#include` (空欄23) 指令により他のソースファイルを組み込んだりすることができる。ここで、これらの処理は、プリプロセッサ (空欄24) により行われる。

- (25) `TREND` は、`SPORTS` に置換され、`SPORTS` はさらに "TENNIS" という文字列に置換される。

```
char str[] = TREND;
    ↓
char str[] = SPORTS;
    ↓
char str[] = "TENNIS";
```

したがって、`ch = str[1]`により、変数 `ch` には 'E' が格納される。

(26) `ADD1` は `a + b` に置換される。

```
x = ADD1 * 2;
    ↓
x = a + b * 2;
```

したがって、変数 `x` には $8 + 3 * 2 = 14$ が格納される。

(27) `ADD2` は `(a + b)` に置換される。

```
y = ADD2 * 2;
    ↓
y = (a + b) * 2;
```

したがって、変数 `y` には $(8 + 3) * 2 = 22$ が格納される。

問 5

(28)	(29)	(30)	(31)	(32)
エ	エ	エ	オ	イ

- (28) `"%-3d"` という書式を指定すると、10 進数の左寄せ 3 桁で出力され、余った桁には空白が出力される。
- (29) `"%04x"` という書式を指定すると、16 進数の右寄せ 4 桁で出力され、余った桁には '0' が出力される。ここで、`"x"` のように英小文字で書式を指定すると、16 進数の 'A' ~ 'F' の値は英小文字で出力される。
- (30) `"%-6s"` という書式を指定すると、6 文字分の領域に左寄せで出力され、余った桁には空白が出力される。
- (31) `"%5s"` という書式を指定すると、5 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。ここで、指定した文字列が 5 文字より長い場合は、文字列の末尾まで出力される。
- (32) `"%10.5s"` という書式を指定すると、文字列の先頭 5 文字が 10 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。

問 6

(33)	(34)	(35)	(36)	(37)
イ	ウ	イ	イ	エ

- (33) Q1 から質問を開始するために、配列 **yesno** の行位置を管理する変数 **q** を **0** に初期化する。したがって、「**q = 0**」となる。
- (34) 反復処理により質問を継続するのは、配列 **yesno** の行位置を管理する変数 **q** が **0 ~ 5** の範囲 (Q1~Q6) にあるときである。したがって、継続条件は「**q >= 0**」となる。
- (35) 変数 **ans** が **1**、すなわち、回答が「YES」の場合は、**yesno[q][0]**に次の質問番号 (1~6) もしくは決定したコース (-4~0) が格納されている。変数 **q** は配列 **yesno** の行位置 (0~5) を管理しているので、**yesno[q][0]**の要素の値から **1** 減算しなければならない。したがって、「**yesno[q][0] - 1**」となる。ここで、**yesno[q][0]**の要素の値がコース (-4~0) の場合は、変数 **q** の値は「**yesno[q][0] - 1**」により **-5~-1** となり、反復処理を終了する。
- (36) 変数 **ans** が **2**、すなわち、回答が「NO」の場合は、**yesno[q][1]**に次の質問番号 (1~6) もしくは決定したコース (-4~0) が格納されているので、設問(35)と同様に、**yesno[q][1]**の要素の値から **1** 減算した値を変数 **q** に代入すればよい。したがって、「**yesno[q][1] - 1**」となる。
- (37) 反復処理が終了するのは、コースが決定したときである。コースが決定したときの変数 **q** の値には、設問(35)で説明したように、**-5~-1** が格納されている。ここで、反復処理が終了した時点、すなわち、コースが決定したときのコースと変数 **q** の値の関係は、次のとおりである。

- ① コース A のとき、変数 **q** の値は **-1**
- ② コース B のとき、変数 **q** の値は **-2**
- ③ コース C のとき、変数 **q** の値は **-3**
- ④ コース D のとき、変数 **q** の値は **-4**
- ⑤ コース E のとき、変数 **q** の値は **-5**

また、空欄(37)の次の行の **printf** 関数によるコース名の出力において、「**'A' + course**」により文字 '**A**' ~ '**E**' を出力していることから、変数 **course** の値は次のとおりである。

- ① コース A のとき、変数 **course** の値は **0**
- ② コース B のとき、変数 **course** の値は **1**
- ③ コース C のとき、変数 **course** の値は **2**
- ④ コース D のとき、変数 **course** の値は **3**
- ⑤ コース E のとき、変数 **course** の値は **4**

したがって、変数 **course** に代入する値は「**-q - 1**」となる。