

第47回

C言語プログラミング能力認定試験

3 級

解答時における注意事項

1. 次の表に従って解答してください。

問題番号	問 1 ～ 問 6
選択方法	6 問必須
試験時間	6 0 分

2. HB の黒鉛筆を使用してください。訂正する場合は，あとが残らないように消しゴムできれいに消し，消しくずを残さないでください。なお，ボールペンや万年筆等で記入した場合は，採点されません。
3. マークシート（解答用紙）の所定の欄に，級種，会場コード，受験番号を記入しマークしてください。また，会場名，氏名及びフリガナ，性別を所定の位置に記入してください。
4. 解答は，次の例題にならって，「解答マーク欄」にマークしてください。

〔例題〕 日本の首都はどこか。

ア 東京 イ 京都 ウ 大阪 エ 福岡

正しい答えは“ア 東京”ですから，次のようにマークしてください。

例題 ☒ ☐ ☐ ☐

指示があるまで開いてはいけません。
試験終了後，問題冊子を回収します。

受験会場	
受験番号	
氏 名	

問 1 ～問 6 は、すべて必須問題です。全問について解答してください。

各設問の答えは、解答群の中から一つだけ選び、括弧中の設問番号に対応したマークシートの解答番号の「解答マーク欄」にマークしてください。なお、二つ以上マークした場合には不正解になります。

問 1 C 言語の特徴に関する次の記述の正誤を、解答群の中から選べ。

- (1) C 言語は、特定の計算機のアーキテクチャと密接に依存しているので、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」を書くことが困難である。
- (2) C 言語では、注釈文を記述することはできない。
- (3) C 言語では、`#include` 前処理命令によって読み込む対象のソースファイル内に、さらに `#include` 前処理命令を記述することはできない。
- (4) 整数定数 `12` と `012` は、異なる値である。
- (5) 標準ライブラリ関数 `toupper` は、ヘッダファイル `ctype.h` で宣言されている。
- (6) `unsigned int` は、符号無し整数型を表す。
- (7) `switch` 文では、「`case` 定数式」で指定した定数式のいずれにも一致しないときの処理を記述することはできない。
- (8) C 言語には、文字列を連結する演算子はない。

解答群

ア 正しい

イ 誤り

問2 データ型及び配列に関する次の記述中の に入れる適切な字句を、解答群の中から選べ。

```
char str1[] = "PARITYBIT";  
char str2[] = {'C', 'H', 'E', 'C', 'K', 'S', 'U', 'M'};  
int pb[][8] = {  
    {4, 3, 1, 5, 9}, {8, 4, 5, 2, 1, 7, 3}, {1, 2, 4, 7, 3, 6}  
};
```

配列 **str1** は (9) 個の要素が確保され、**str1[5]**は (10) である。

配列 **str2** は (11) 個の要素が確保される。

配列 **pb** は (12) × 8 の配列であり、**pb[1][2]**は (13) である。

(9) の解答群

ア 7 イ 8 ウ 9 エ 10 オ 11

(10) の解答群

ア 'B' イ 'I' ウ 'R' エ 'T' オ 'Y'

(11) の解答群

ア 7 イ 8 ウ 9 エ 10 オ 11

(12) の解答群

ア 3 イ 4 ウ 5 エ 6 オ 7

(13) の解答群

ア 0 イ 2 ウ 3 エ 5 オ 8

問3 次の while 文及び do ~ while 文を使ったプログラムを実行したとき, printf 関数によって出力される値を, 解答群の中から選べ。

<プログラム>

```
#include <stdio.h>

int main(void)
{
    int x, y, v;
    int data[] = {3, 9, 1, 5, 2, 8, 4, 6, 7};

    x = 5;
    y = 8;
    v = 0;
    while (x < y) {
        v += 1;
        x += v;
    }
    printf("%d\n", x);          ... (14)
    printf("%d\n", v);          ... (15)

    x = 5;
    y = 2;
    v = 3;
    while (x >= y) {
        v += 1;
        x -= v;
    }
    printf("%d\n", x);          ... (16)
    printf("%d\n", v);          ... (17)

    x = 1;
    y = 2;
    v = 3;
    do {
        v += 1;
        x -= v;
    } while (x > y);
    printf("%d\n", x);          ... (18)
    printf("%d\n", v);          ... (19)
```

```

x = 7;
y = 0;
v = -3;
while (x > y) {
    v -= data[x - y];
    do {
        x -= 1;
        v += 8;
    } while (v < 0);
    y += 3;
}
printf("%d\n", y);          ... (20)
printf("%d\n", v);          ... (21)

return 0;
}

```

(14) の解答群

ア	5	イ	8	ウ	9	エ	11	オ	12
---	---	---	---	---	---	---	----	---	----

(15) の解答群

ア	0	イ	1	ウ	2	エ	3	オ	4
---	---	---	---	---	---	---	---	---	---

(16) の解答群

ア	-4	イ	-1	ウ	0	エ	1	オ	5
---	----	---	----	---	---	---	---	---	---

(17) の解答群

ア	3	イ	4	ウ	5	エ	6	オ	7
---	---	---	---	---	---	---	---	---	---

(18) の解答群

ア	-3	イ	-2	ウ	-1	エ	0	オ	1
---	----	---	----	---	----	---	---	---	---

(19) の解答群

ア	3	イ	4	ウ	5	エ	6	オ	7
---	---	---	---	---	---	---	---	---	---

(20) の解答群

ア	0	イ	3	ウ	6	エ	9	オ	12
---	---	---	---	---	---	---	---	---	----

(21) の解答群

ア	-1	イ	6	ウ	7	エ	14	オ	17
---	----	---	---	---	---	---	----	---	----

問4 インクリメント演算子及びデクリメント演算子に関する次の記述中の に入れる適切な字句を、解答群の中から選べ。なお、(24)～(27)の解答は重複して選んでもよい。

前置インクリメント演算子の結果は、 となるのに対し、後置インクリメント演算子の結果は、 となる。

次のプログラムを実行すると、標準出力には、

a :

b :

d :

i :

と表示される。

<プログラム>

```
#include <stdio.h>

int main(void)
{
    int i, a, b, c, d;

    i = 10;
    a = i++;
    b = ++i;
    c = --i;

    if (c > i)
        d = ++i;
    else
        d = i--;

    printf("a : %d\n", a);
    printf("b : %d\n", b);
    printf("d : %d\n", d);
    printf("i : %d\n", i);

    return 0;
}
```

(22) , (23) の解答群

- ア オペランドの値を増分する前の値
- イ オペランドの値を増分した後の値
- ウ オペランドの値を減分する前の値
- エ オペランドの値を減分した後の値
- オ 不定値

(24) ～ (27) の解答群

- ア 9
- イ 10
- ウ 11
- エ 12
- オ 13

問5 次のプログラムを実行したとき、printf 関数によって出力される値を、解答群の中から選べ。なお、解答群中のΔは空白文字を表す。

<プログラム>

```
#include <stdio.h>

int main(void)
{
    printf("[%4d]¥n", 305);           ... (28)
    printf("[%04x]¥n", 0xF4A);       ... (29)
    printf("[%8s]¥n", "Binary");     ... (30)
    printf("[%3s]¥n", "Text");       ... (31)
    printf("[%7.5s]¥n", "ReadWrite"); ... (32)

    return 0;
}
```

(28) の解答群

ア	[Δ305]	イ	[0305]	ウ	[305]
エ	[305Δ]	オ	[-305]		

(29) の解答群

ア	[ΔF4A]	イ	[Δf4a]	ウ	[0F4A]
エ	[0f4a]	オ	[XF4A]		

(30) の解答群

ア	[ΔΔBinary]	イ	[ΔBinaryΔ]	ウ	[BINARY]
エ	[Binary]	オ	[BinaryΔΔ]		

(31) の解答群

ア	[ΔΔΔText]	イ	[ext]	ウ	[Tex]
エ	[Text]	オ	[TextΔΔΔ]		

(32) の解答群

ア	[ΔΔReadW]	イ	[adWrite]	ウ	[ReadWri]
エ	[ReadWΔΔ]	オ	[ReadWrite]		

問6 次のプログラムの説明を読んで、プログラム中の に入れる適切な字句を、解答群の中から選べ。

＜プログラムの説明＞

標準入力から文字列を入力し、入力した文字列（以下、入力文字列という）を標準出力に表示する。ここで、入力文字列に「%<数字 1 桁>」（「%」の次に 0～9 のうちの 1 文字が続く 2 文字の文字列。以下、キー文字列という）が含まれている場合、この部分を、あらかじめ決められている置換用の文字列（以下、置換文字列という）に置換して表示する。

なお、入力文字列は、最大 79 文字で、A～Z, a～z, 0～9, 及び記号 % で構成される。

キー文字列と置換文字列の対応を、表 1 に示す。

表 1 キー文字列と置換文字列の対応

キー文字列	置換文字列
%0	HOKKAIDO
%1	TOHOKU
%2	HOKURIKU
%3	KANTO
%4	CHUBU
%5	TOKAI
%6	KINKI
%7	CHUGOKU
%8	SHIKOKU
%9	KYUSYU

また、入力文字列中に「%」が 2 文字連続している場合は、「%」1 文字のみを表示する。さらに、入力文字列中の「%」の次の文字が数字でない場合は、「不正なキー文字列 : %<不正文字>」を表示して、プログラムを中断する。ここで、「%<不正文字>」とは、「%」の次に数字 0～9 及び%以外の文字 1 文字が続く 2 文字の文字列のことである。

＜処理手順＞

- (1) 標準入力から入力文字列を入力する。
- (2) 入力文字列の先頭の文字から末尾の文字までを順に走査して、標準出力に表示する文字列（以下、出力文字列という）を作成する。
- (3) 走査の対象文字（以下、走査対象文字という）が「%」で、かつ、次の文字が数字（0～9）であれば、その数字を数値に変換し、その数値に対応する置換文字列を出力文字列とする。
- (4) 走査対象文字が「%」で、かつ、次の文字も「%」であれば、「%」1 文字を出力文字列とする。

- (5) 走査対象文字が「%」で、かつ、次の文字が数字（0～9）でも「%」でもない場合は、
「不正なキー文字列：%<不正文字>」を表示して、プログラムを中断する。
- (6) 走査対象文字が「%」でない場合は、そのままその文字を出力文字列とする。
- (7) 置換処理が正常に終了した場合は、「置換後文字列：<出力文字列>」を表示する。ここで、「<出力文字列>」とは、入力文字列を末尾まで走査し、正常に置換処理が終了した文字列である。

《実行結果例》 （正常に置換できる場合）

入力文字列 : from%1to%7%%via%5
置換後文字列 : fromTOHOKUtoCHUGOKU%viaTOKAI

上記の入力文字列と置換後文字列における実行結果例の置換の対応を、表 2 に示す。

表 2 実行結果例の置換の対応

入力文字列	置換後文字列
from	from
%1	TOHOKU
to	to
%7	CHUGOKU
%%	%
via	via
%5	TOKAI

《実行結果例》（%に続く文字が数字でない文字列を含む場合）

入力文字列 : from%Sto%2
不正なキー文字列 : %S

<プログラム>

```
#include <stdio.h>
#include <ctype.h>

int main(void)
{
    char rep[10][10] = {
        "HOKKAIDO",    /* %0 */
        "TOHOKU",      /* %1 */
        "HOKURIKU",    /* %2 */
        "KANTO",        /* %3 */
        "CHUBU",        /* %4 */
    }
```

```

    "TOKAI",          /* %5 */
    "KINKI",          /* %6 */
    "CHUGOKU",        /* %7 */
    "SHIKOKU",        /* %8 */
    "KYUSYU"          /* %9 */
};

char inbuf[80], outbuf[1024];
int i, j, k, n, eflg;

printf("入力文字列   : ");
scanf("%s", inbuf);
eflg = 0;
i = 0;
j = 0;
while (inbuf[i] != '\0') {
    if (inbuf[i] == '%') {
        if ((33)) {
            n = inbuf[i + 1] - '0';
            for (k = 0; (34); k++) {
                outbuf[j] = rep[n][k];
                j++;
            }
        } else if ((35)) {
            outbuf[j] = '%';
            j++;
        } else {
            eflg = 1;
            break;
        }
        i += 2;
    } else {
        outbuf[j] = inbuf[i];
        i++;
        j++;
    }
}
if (eflg == 0) {
    (36);
    printf("置換後文字列 : %s\n", outbuf);
} else {
    printf("不正なキ一文字列 : %%%c\n", (37));
}

return 0;
}

```

(33) の解答群

ア `isalnum(inbuf[i])`
ウ `isdigit(inbuf[i])`

イ `isalnum(inbuf[i + 1])`
エ `isdigit(inbuf[i + 1])`

(34) の解答群

ア `rep[n][k] == '¥0'`
ウ `rep[n][k] == '%'`

イ `rep[n][k] != '¥0'`
エ `rep[n][k] != '%'`

(35) の解答群

ア `inbuf[i + 1] == '%'`
ウ `inbuf[i + 1] == outbuf[j]`

イ `inbuf[i + 1] != '%'`
エ `inbuf[i + 1] != outbuf[j]`

(36) の解答群

ア `outbuf[j] = '¥0'`
ウ `outbuf[j - 1] = '¥0'`

イ `outbuf[j] = '%'`
エ `outbuf[j - 1] = '%'`

(37) の解答群

ア `inbuf[i]`
ウ `inbuf[i - 1]`

イ `inbuf[i + 1]`
エ `inbuf[i - 2]`

[メ モ 用 紙]

試験問題は著作権法上の保護を受けています。

試験問題の一部または全部について、サーティファイから文書による許諾を得ずに、いかなる方法においても私的使用の範囲を超えて、無断で複写、複製することを禁じます。

無断複製、転載は損害賠償、著作権法の罰則の対象になることがあります。

©CERTIFY Inc.2019