

サンプル問題

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	60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	

試験問題に記載されている会社名又は製品名は、それぞれ各社の商標又は登録商標です。
なお、試験問題では、® 及び TM を明記していません。

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、Niklaus Wirth により教育用プログラミング言語として開発された Pascal 言語を改良したものである。
- (2) C 言語では、関数名や変数名などの決め方は自由で何も制限がない。
- (3) C 言語では、整数型の定数記述において、8 進数表記、10 進数表記、16 進数表記の表現が可能である。
- (4) C 言語で扱える多次元配列は 2 次元までで、3 次元以降の配列は使用できない。
- (5) C 言語のソースプログラムには、注釈（コメント）を記述することができるが、注釈の量に比例して、作成されたプログラムの実行速度が低下する。
- (6) C 言語には、文字列操作に関する演算子は用意されていない。
- (7) EOF は、ヘッダファイル `stdio.h` で定義されている。
- (8) C 言語は、構造化プログラミングに適した言語仕様となっている。

解答群

ア 正しい

イ 誤り

問2 プリプロセッサ機能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。

マクロの定義を行う `#define` 命令は、コンパイラの前に実行されるプリプロセッサで処理される。

`#define` 命令を使うと、ソースプログラム上の文字の並びを他の文字の並びに置き換えることができる。例えば、<プログラム1>を実行すると、画面には

```
(9) =   
(10) =   
(11) = 
```

と出力される。

<プログラム1>

```
#include <stdio.h>  
  
#define CC      "CCompiler"  
#define ADD1    1 + 3  
#define ADD2    x + y + z  
  
int main(void)  
{  
    int x = 8, y = 2, z = 5;  
    printf("(9) = %s\n", CC);  
    printf("(10) = %d\n", ADD1);  
    printf("(11) = %d\n", ADD2);  
    return 0;  
}
```

ただし、マクロによる文字列の置換は、単純に文字の並びを置き換えるだけなので、置換する文字列が多項式の場合は、期待した結果が得られないことがある。例えば、<プログラム2>を実行すると、画面には

```
(12) = 
```

と出力される。

<プログラム2>

```
#include <stdio.h>
#define ADD2      x + y + z ← α

int main(void)
{
    int x = 8, y = 2, z = 5;
    printf("(12) = %d¥n", ADD2 * 2);
    return 0;
}
```

ADD2 の計算結果 (x と y と z の和) に 2 を乗じた値を求めたい場合は、 α の行を以下のように記述すればよい。

```
#define ADD2 
```

(9) の解答群

ア C	イ CC	ウ CCOMPILER
エ CCompiler	オ ompiler	

(10) の解答群

ア 1	イ 2	ウ 3	エ 4	オ 5
-----	-----	-----	-----	-----

(11) の解答群

ア 2	イ 5	ウ 8	エ 10	オ 15
-----	-----	-----	------	------

(12) の解答群

ア 4	イ 8	ウ 10	エ 20	オ 30
-----	-----	------	------	------

(13) の解答群

ア $x + y + (z)$	イ $x + (y + z)$	ウ $(x + y) + z$
エ $(x + y + z)$	オ $(x) + (y) + (z)$	

問3 次のプログラム1, プログラム2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1>

```
#include <stdio.h>

int main(void)
{
    int i, a, b;
    int data[] = { 2, 2, 1, 2, 1, 1 };

    a = 1;
    b = 1;
    for (i = 0; i < 6; i++) {
        switch (i) {
            case 1:
                a *= 3;
                break;
            case 3:
                a -= 5;
                break;
        }
        switch (i * data[i]) {
            case 2:
            case 6:
                b += 3;
                break;
            case 4:
                b += 2;
                break;
            default:
                b += 4;
                break;
        }
    }
    printf("%d¥n", a);          ... (14)
    printf("%d¥n", b);          ... (15)
    return 0;
}
```

<プログラム2>

```
#include <stdio.h>

int main(void)
{
```

```

int i, a, b, work[] = { 0, 0, 0, 0 };
char str[][4] = { "JAN", "FEB", "MAR", "APL", "MAY" };

a = 0;
b = 0;
for (i = 0; i < 5; i++) {
    switch (str[i][0]) {
        case 'J':
            a += 2;
            switch (str[i][1]) {
                case 'U':
                    b += 7;
                    break;
                case 'A':
                    b += 9;
                    work[i % 4]++;
                    break;
            }
            break;
        case 'M':
            a += 3;
            switch (str[i][1]) {
                case 'U':
                    b -= 7;
                    break;
                case 'A':
                    b -= 9;
                    work[i % 4] += 2;
                    break;
            }
            break;
        default:
            a += 5;
            switch (str[i][2]) {
                case 'L':
                    b += 7;
                    break;
                case 'N':
                    b -= 9;
                    break;
                default:
                    work[i % 2]++;
                    break;
            }
            break;
    }
}
}

```

```

printf("%d\n", a);          ... (16)
printf("%d\n", b);          ... (17)
printf("%d\n", work[0]);    ... (18)
printf("%d\n", work[1]);    ... (19)
printf("%d\n", work[2]);    ... (20)
printf("%d\n", work[3]);    ... (21)
return 0;
}

```

(14) の解答群

ア -3 イ -2 ウ -1 エ 0 オ 1

(15) の解答群

ア 5 イ 8 ウ 16 エ 20 オ 32

(16) の解答群

ア 17 イ 18 ウ 19 エ 20 オ 21

(17) の解答群

ア -3 イ -2 ウ -1 エ 0 オ 1

(18) の解答群

ア 0 イ 1 ウ 2 エ 3 オ 4

(19) の解答群

ア 0 イ 1 ウ 2 エ 3 オ 4

(20) の解答群

ア 0 イ 1 ウ 2 エ 3 オ 4

(21) の解答群

ア 0 イ 1 ウ 2 エ 3 オ 4

問4 scanf 関数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。

scanf 関数は書式付きの入力関数であり、標準入力からの入力を、書式に従って変数へ読み込む。ここで、書式指定文字列に含まれる変換指定子には、以下のものがある。

- | | |
|------|----------------------|
| (22) | … 1文字を入力する。 |
| (23) | … 8進数（整数）を入力する。 |
| (24) | … 10進数（符号付き整数）を入力する。 |
| (25) | … 16進数（整数）を入力する。 |
| (26) | … 文字列を入力する。 |

なお、scanf 関数を利用するには、ヘッダファイル (27) .h をインクルードする必要がある。

(22) ~ (26) の解答群

ア %c イ %d ウ %o エ %s オ %x

(27) の解答群

ア ctype イ define ウ EOF
エ include オ stdio

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、解答群の△は空白文字を表す。

<プログラム>

```
#include <stdio.h>

int main(void)
{
    printf("[%5o]¥n", 210);           … (28)
    printf("[%05d]¥n", 0x234);       … (29)
    printf("[%3s]¥n", "ASIA");       … (30)
    printf("[%5.3s]¥n", "AMERICA");  … (31)
    printf("[%−8.8s]¥n", "EUROPE");  … (32)
    return 0;
}
```

(28) の解答群

ア [△△322]	イ [210]	ウ [210△△]
エ [322]	オ [322△△]	

(29) の解答群

ア [△△234]	イ [△△564]	ウ [00234]
エ [00564]	オ [56400]	

(30) の解答群

ア [△△△ASIA]	イ [ASI]	ウ [ASIA]
エ [ASIA△△△]	オ [SIA]	

(31) の解答群

ア [△△AME]	イ [AME]	ウ [AME△△]
エ [AMERI]	オ [AMERICA]	

(32) の解答群

ア [−△△EUROPE]	イ [−EUROPE△△]	ウ [△△EUROPE]
エ [EUROPE]	オ [EUROPE△△]	

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>

入力文字列（1～127文字）の中から、「'@' 数字('0'～'6')」の文字列を探し出し、見つかった場合は、以下のように数字に対応する文字列（曜日を表す英単語の省略形）に置換した文字列（以下、置換後文字列という）を画面に出力する。

```
"@0" → "SUN"    "@1" → "MON"    "@2" → "TUE"    "@3" → "WED"
"@4" → "THU"    "@5" → "FRI"    "@6" → "SAT"
```

ただし、「@」の次の文字も '@' であった場合、「@」の1文字だけを画面に出力する。

```
"@@ " → "@ "
```

'@'の次の文字が、'0'～'6'、'@'文字以外のときは、その部分（'@'と'@'に続く1文字）は何も画面に出力しない。上記以外の場合は、入力文字をそのまま画面に出力する。

例えば、入力文字列が

```
"@1-@2-@@-@9-@3-@s!"
```

のとき、置換後の文字列は

```
"MON-TUE-@--WED-!"
```

となる。

なお、入力文字列には、「scanf 関数における入力項目を区切る文字（空白文字やタブ文字など）」は含まれないものとし、置換後文字列の文字数は127文字を超えないものとする。

<処理の流れ>

- ① 入力文字列 (is) を入力する。
- ② 入力された入力文字列について先頭から順番に文字を走査し、「@」であれば③～⑤の処理を、「@」でなければ⑥の処理を実行する。入力文字列の末尾の文字まで処理し終えたら、⑦へ移行する。
- ③ 次の文字を走査し、「@」であれば、「@」の1文字に置換して次の文字を走査する(②へ戻る)。
- ④ 次の文字を走査し、「0'～'6」であれば数字に対応した文字列（曜日を表す英単語の省略形）に置換して次の文字を走査する(②へ戻る)。
- ⑤ 次の文字を走査し、「@', '0'～'6」以外のときは読み捨てて、次の文字を走査する(②へ戻る)。
- ⑥ 入力文字を置換せずにそのまま出力し、次の文字を走査する(②へ戻る)。
- ⑦ 置換後文字列の末尾に'¥0'を格納した後、置換後文字列を画面に出力する。

《実行例》

入力文字列 : 2-@0,3-@0,4-@3

置換後文字列 : 2-SUN,3-SUN,4-WED

<プログラム>

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

int main(void)
{
    char week[][4] = { "SUN", "MON", "TUE", "WED", "THU", "FRI", "SAT" };
    char is[128], os[128];
    int i, j, k, n;

    printf("入力文字列 : ");
    scanf("%s", is);

    i = 0;
    j = 0;
    while ( (33) ) {
        if (is[i] == '@') {
            if ( (34) ) {
                os[j++] = '@';
            } else if ( (35) ) {
                n = is[i + 1] - '0';
                if (n < 7) {
                    k = 0;
                    while (week[n][k] != '\0')
                        (36);
                }
            }
            i += 2;
        } else
            (37);
    }
    os[j] = '\0';
    printf("置換後文字列 : %s", os);

    return 0;
}
```

(33) の解答群

ア `is[i] == '¥0'`

イ `is[i] != '¥0'`

ウ `is[j] == '¥0'`

エ `is[j] != '¥0'`

(34) の解答群

ア `is[i + 1] == '@'`

イ `is[i + 1] != '@'`

ウ `is[j] == '@'`

エ `is[j] != '@'`

(35) の解答群

ア `isalpha(is[i + 1])`

イ `isdigit(is[i + 1])`

ウ `isspace(is[i + 1])`

エ `isupper(is[i + 1])`

(36) の解答群

ア `os[j++] = week[k][n]`

イ `os[j++] = week[k++][n]`

ウ `os[j++] = week[n][k]`

エ `os[j++] = week[n][k++]`

(37) の解答群

ア `os[j] = is[i]`

イ `os[j] = is[i++]`

ウ `os[j++] = is[i]`

エ `os[j++] = is[i++]`

試験問題内容に関して、他人にこれを伝え、漏洩することを禁じます。
©CERTIFY Inc.2015 禁無断転載複写

サンプル問題
C言語プログラミング能力認定試験
3級

<正答>

問1

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

問2

(9)	(10)	(11)	(12)	(13)
エ	エ	オ	エ	エ

問3

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

問4

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

問5

(28)	(29)	(30)	(31)	(32)
オ	エ	ウ	ア	オ

問6

(33)	(34)	(35)	(36)	(37)
イ	ア	イ	エ	エ