

サンプル問題

C言語プログラミング能力認定試験

2 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8 |
| 選択方法 | 8問必須 |
| 試験時間 | 90分 |

2. HBの黒鉛筆を使用してください。訂正の場合は、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。
3. 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、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 ™ を明記していません。

次の問1～問8はすべて必須問題です。全問について解答してください。

答えは、解答群の中から一つずつ選び、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言語に関する次の記述の正誤を、解答群の中から選べ。

- (1) printf 関数は正常時には出力した文字数を返す。
- (2) 浮動小数点型は float, double, long double の三つのサイズをもち、ANSI 規格により、それぞれのサイズは規定されている。
- (3) ポインタ変数を const 宣言することはできない。
- (4) typedef はコンパイラで解釈され、#define はプリプロセッサで解釈される。
- (5) キャストは明示的に型変換することであり、ポインタをキャストすることも可能である。
- (6) グローバル変数（広域変数）とローカル変数（局所変数）に同一の名前が使われたとき、両者のスコープが及ぶ箇所では、グローバル変数が優先される。
- (7) 複数の論理演算子を使用した条件式において、真偽の判定は右から左の順に行われる。
- (8) float 型や double 型に対して、ビット演算子を使うことはできない。

解答群

ア 正しい

イ 誤り

問2 列挙型に関する次の記述について に入れる適切な字句を、解答群の中から選べ。また、(11)～(14)で出力される値を、解答群の中から選べ。

列挙型は、データの取り得る値をすべて列挙定数と呼ばれる名前で列挙することにより、定義されるデータ型である。列挙型の定義は、次のとおりである。

(9) 列挙タグ { 列挙定数のリスト } 列挙変数;

列挙定数の内部表現としての値は (10) 型である。

<プログラム>

```
#include <stdio.h>
main()
{
     (9) col {cyan, magenta, yellow = 6, black} color;
    int sum = 0, i;

    color = magenta;
    for (i = cyan; i <= black; i++)
        sum++;
    printf("%d", cyan);           … (11)
    printf("¥n%d", black);       … (12)
    printf("¥n%d", color);       … (13)
    printf("¥n%d", sum);         … (14)
}
```

(9) , (10) の解答群

ア char イ enum ウ extern エ int オ static

(11) , (13) の解答群

ア 0 イ 1 ウ 2 エ 3 オ 4

(12) , (14) の解答群

ア 3 イ 4 ウ 6 エ 7 オ 8

問3 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。また、(19)で出力される値を、解答群の中から選べ。

変数 dat_1 の下位 8 ビットを上位 8 ビットとし、変数 dat_2 の上位 8 ビットを下位 8 ビットとしたビット列を反転した値を表示するプログラムである。

<プログラム>

```
#include <stdio.h>
main()
{
    unsigned short dat_1 = 0xABCD;
    unsigned short dat_2 = 0x1234;

    dat_1  8;
    dat_2  8;
    dat_1  dat_2;
    dat_2 =  dat_1;
    printf("%04hX", dat_2);    ... (19)
}
```

(15) ~ (17) の解答群

ア <<= イ >>= ウ &= エ |= オ *=

(18) の解答群

ア & イ ^ ウ ~ エ ! オ |

(19) の解答群

ア 12CD イ 32ED ウ 4402 エ CD12 オ ED32

問4 構造体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。

次のように日付を格納する構造体を宣言した。

```
 (20) hizuke {  
    int year;      /* 年 */  
    int month;    /* 月 */  
    int day;      /* 日 */  
} date_1, *date_2 = &date_1;
```

この構造体のhizukeは (21) であり、省略することは (22) である。

この構造体の変数date_1に値が格納されているとき、「年」を参照する記述として正しいのは (23) であり、「月」を参照する記述として正しいのは (24) である。また、「日」を参照する記述として正しいのは (25) である。

(20) の解答群

| | | |
|-----------|----------|-----------|
| ア #define | イ struct | ウ typedef |
| エ union | オ void | |

(21) の解答群

| | | |
|----------|---------|----------|
| ア 構造体タグ名 | イ 構造体変数 | ウ アロー演算子 |
| エ ポインタ変数 | オ メンバ名 | |

(22) の解答群

| | |
|------|-------|
| ア 可能 | イ 不可能 |
|------|-------|

(23) の解答群

| | | |
|---------------|----------------|----------------|
| ア hizuke.year | イ date_1.year | ウ date_1->year |
| エ date_2.year | オ *date_2.year | |

(24) の解答群

| | | |
|-----------------|-------------------|-----------------|
| ア hizuke.month | イ date_1->month | ウ *date_1.month |
| エ *date_2.month | オ (*date_2).month | |

(25) の解答群

| | | |
|---------------|---------------|---------------|
| ア hizuke.day | イ date_1->day | ウ *date_1.day |
| エ date_2->day | オ *date_2.day | |

問5 次のプログラムを実行したとき、(26)～(30)で出力される値を、解答群の中から選べ。

<プログラム>

```
#include <stdio.h>

main()
{
    char mark_1[][5] = {"zyx", "wv", "utsr", "qpo", "nmlk"};
    char *mark_2[5] = {"cba", "ed", "ihgf", "lkj", "ponm"};
    char *pt_1, **pt_2;

    pt_1 = *mark_1;
    printf("%s¥n", mark_1[1] + 1);    ... (26)
    printf("%s¥n", pt_1 + 1);        ... (27)
    pt_2 = mark_2;
    printf("%s¥n", *pt_2 + 2);        ... (28)
    printf("%c¥n", **pt_2 + 2);        ... (29)
    printf("%s¥n", *(pt_2 + 2) + 2); ... (30)
}
```

(26) の解答群

ア zyx イ yx ウ wv エ v オ utsr

(27) の解答群

ア z イ yx ウ wv エ zx オ v

(28) の解答群

ア a イ cba ウ c エ ed オ ihgf

(29) の解答群

ア c イ d ウ e エ f オ ihgf

(30) の解答群

ア e イ gf ウ i エ ihgf オ ponm

問6 次の各関数と同等の処理を行う標準ライブラリ関数を、解答群の中から選べ。

```
(31) int func_a(int ch)
{
    if ( isalpha(ch) || isdigit(ch) )
        return 1;
    else
        return 0;
}
```

解答群

ア isalnum イ isalpha ウ isdigit エ islower オ isupper

```
(32) int func_b(int ch)
{
    ch = tolower(ch);
    if ( isdigit(ch) || ch >= 'a' && ch <= 'f' )
        return 1;
    else
        return 0;
}
```

解答群

ア isalnum イ isalpha ウ isdigit
エ ispunct オ isxdigit

```
(33) char *func_c(char *s1, const char *s2)
{
    char *s3 = s1;
    for ( ; *s2; s2++ )
        *s1++ = *s2;
    *s1 = '\0';
    return s3;
}
```

解答群

ア strcat イ strchr ウ strcmp エ strcpy オ strlen

(34) char *func_d(char *s1, const char *s2)

```
{
    char *s3 = s1;
    for ( ; *s1 != '\0'; s1++ );
    for ( ; *s2 != '\0'; s1++, s2++ )
        *s1 = *s2;
    *s1 = '\0';
    return s3;
}
```

解答群

ア strcat イ strchr ウ strcmp エ strcpy オ strlen

(35) char *func_e(char *s1, const char *s2, size_t n)

```
{
    char *s3 = s1;
    for ( ; n > 0; n-- )
        if ( *s2 == '\0' )
            *s1++ = '\0';
        else
            *s1++ = *s2++;
    return s3;
}
```

解答群

ア strncat イ strncmp ウ strncpy エ strstr オ sscanf

問7 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>

このプログラムは、四字熟語の学習プログラムである。

登録されている四字熟語の1文字を乱数により隠された状態で表示する。その隠された文字をキーボードから入力し、正誤を判定する。

問題数は10問とし、乱数により出題する四字熟語を選択している。なお、同じ問題は出題されないようにする。

<処理手順>

- ① 登録されている四字熟語の中から、乱数により出題する問題を選択する。すでに
出題されている問題であれば、出題されていない問題が選択されるまで処理を繰り返す。
- ② 登録されている四字熟語が破壊されないように選択された問題をコピーする。
- ③ 乱数により選択された文字（隠す文字）を、あらかじめ用意された正答を格納する
配列にコピーする。
- ④ コピーされた問題に対し、乱数により選択された文字を隠す。（'?'を該当文字位
置に格納する。）
- ⑤ キーボードから入力された解答と正答が一致するかどうかを判定する。
- ⑥ 10問出題されるまで①～⑤の処理を繰り返す。

<実行結果>

問1：無？夢中
解答＝我
結果…○

問2：弱肉？食
解答＝強
結果…○

問3：絶？絶命
解答＝体
結果…×

⋮

問10：不言？行
解答＝実
結果…○

<プログラム>

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#define TOI 12      /* 登録されている四字熟語の数 */

main()
{
    char *word[TOI] = {"大同小異", "四面楚歌", "針小棒大",
                      "五里霧中", "弱肉強食", "異口同音",
                      "絶対絶命", "起承転結", "大胆不敵",
                      "不言実行", "半信半疑", "無我夢中"};

    char mon[9];          /* 出題される四字熟語を格納 */
    char kai[128];       /* 入力された解答を格納 */
    char sei[2];         /* 正答を格納 */
    char (36);
    int out[TOI];        /* 出題された問題の状況を登録 */
    int r1, r2, i, q = 1;

    for (i = 0; i < TOI; i++)
        out[i] = 0;

    while (q <= 10) {          /* 出題数は10問とする */
        r1 = rand() % 4;
        do {
            r2 = rand() % TOI;
        } while ( (37) );
        out[r2] = 1;
        (38);                  /* 問題をコピー */
        strncpy(sei, m_t + r1 * 2, 2); /* 正答の作成 */
        (39);                  /* 穴抜きの作成 */
        printf("\n問%d: %s\n", q, mon);
        printf("解答=");
        scanf("%s", kai);
        printf("結果…");
        if ( (40) )            /* 判定 */
            printf("○\n");
        else
            printf("×\n");
        q = q + 1;
    }
}
```

(36) の解答群

ア `m_t = mon`
ウ `*m_t = mon`

イ `m_t = *mon`
エ `*m_t = *mon`

(37) の解答群

ア `out[r2] != 0`
ウ `out[r2] == 0`

イ `out[r2] != 1`
エ `out[r2] == 2`

(38) の解答群

ア `strcpy(mon, word[r2])`
イ `strcpy(word[r2], mon)`
ウ `strncpy(mon, word[r2], 8)`
エ `strncpy(word[r2], mon, 8)`

(39) の解答群

ア `strcpy(m_t + r1, "?")`
イ `strcpy(m_t + r1 * 2, "?")`
ウ `strncpy(m_t + r1 * 2, "?", 2)`
エ `strncpy((m_t + r1) * 2, "?", 2)`

(40) の解答群

ア `strcmp(kai, sei) != 0`
イ `strcmp(kai, sei) == 0`
ウ `strncmp(kai, sei, 2) != 0`
エ `strncmp(kai, sei, 2) == 0`

問8 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>

このプログラムの目的は、テキストファイル (word.dat) に格納されているタブ記号 ('@') を、タブ間隔が 4 文字分となるように、適切な数の間隔文字 (空白文字) に置き換えて表示することである。

なお、テキストファイルは 30 行未満であり、1 行の文字列は 255 字以内であるものとする。

<文書ファイル (word.dat) の例>

```
a@bc@def@ghij@k
lmn@op@qr
@st@u@vwxyz
ABC@DE@FGI
HIJ@KLMNOP@QR
STUVWXYZ
```

<文書ファイルの例で表示された実行結果>

```
a   bc  def  ghij   k
lmn op  qr
      st  u   vwxyz
ABC DE      FGI
HIJ KLMNOP  QR
STUVWXYZ
```

<タブ記号の変換>

- ① 文字列を先頭から順に調べ、タブ記号を間隔文字に置換する。タブ記号以外の文字は、そのまま出力する。
- ② タブ記号が出現した場合、最も近い 4 の倍数+1 が次の文字位置になるように、タブ記号を一つ以上の間隔文字に置換する。

| | | | | | | | | | | | | | | | | | | | | | |
|------|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|----|----|----|----|----|----|
| 文字位置 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 |
| 展開前 | a | @ | b | c | @ | d | e | f | @ | g | h | i | j | @ | k | | | | | | |
| | | | | | | | | | ↓ | | | | | | | | | | | | |
| 展開後 | a | △ | △ | △ | b | c | △ | △ | d | e | f | △ | g | h | i | j | △ | △ | △ | △ | k |

※△は半角の空白文字を表す。

<処理手順>

- ① 文書ファイルを入力し、2次元配列に格納する。
- ② 配列に格納された文書をタブ編集する。
 - ・文書ファイルの内容が格納された2次元配列を、行単位に検索する。
 - ・先頭から文字単位で検索し、文書を編集後の配列に格納する。
 - ・タブ記号の場合は、最も近い4の倍数の文字位置まで空白を格納し、タブ記号以外の文字はそのまま編集後の配列に転送する。
 - ・編集後の配列を、文書ファイルが格納された2次元配列の該当行に上書きコピーする。
- ③ タブ編集された文書を表示する。

<プログラム>

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

int file_read(void);          /* ファイル入力 */
void henkan(int cnt);        /* タブ編集 */

char data[30][255 * 4 + 1];   /* 文書格納用配列 */

main()
{
    int cnt;                  /* 行数 */
    int i;

    cnt = file_read();        /* ファイル入力 */
    henkan(cnt);              /* タブ編集 */
    /* 画面表示 */
    for (i = 0; i < cnt; i++)
        printf("%s", data[i]);
}

/* ファイル入力 */
int file_read(void)
{
    FILE *fp;
    int cnt = 0;

    if ((fp = fopen("word.dat", "r")) == NULL) {
```

```

        printf("入力ファイルがオープンできません\n");
        exit(1);
    }
    while (fgets( (41) , 256, fp) != NULL)
        ;
    fclose(fp);
    return cnt;
}

/* タブ編集 */
void henkan(int cnt)
{
    char buf[255 * 4 + 1];
    int i, j, k, tb;

    for (i = 0; i < cnt; i++) {
        for (j = 0, k = 0; (42); j++) {
            if (data[i][j] == '@') {
                tb = (43);
                for ( ; k < tb; k++)
                    buf[k] = ' ';
            }
            else
                (44);
        }
        (45);
        strcpy(data[i], buf);
    }
}

```

(41) の解答群

| | | | |
|---|--------------------------|---|--------------------------|
| ア | <code>data[--cnt]</code> | イ | <code>data[cnt]</code> |
| ウ | <code>data[++cnt]</code> | エ | <code>data[cnt++]</code> |

(42) の解答群

| | |
|---|---------------------------------|
| ア | <code>data[i][j] != '¥0'</code> |
| イ | <code>data[i][j] == '¥0'</code> |
| ウ | <code>data[j][i] != '¥0'</code> |
| エ | <code>data[j][i] == '¥0'</code> |

(43) の解答群

| | | | |
|---|----------------------------|---|----------------------------------|
| ア | <code>k / 4 * 4</code> | イ | <code>(k / 4 + 1) * 4</code> |
| ウ | <code>k / 4 * 4 + 1</code> | エ | <code>(k / 4 + 1) * 4 + 1</code> |

(44) の解答群

| | | | |
|---|----------------------------------|---|------------------------------------|
| ア | <code>buf[k] = data[i][j]</code> | イ | <code>buf[k++] = data[i][j]</code> |
| ウ | <code>data[i][j] = buf[k]</code> | エ | <code>data[i][j] = buf[k++]</code> |

(45) の解答群

| | | | |
|---|----------------------------|---|-----------------------------|
| ア | <code>buf[i] = '¥0'</code> | イ | <code>buf[j] = '¥0'</code> |
| ウ | <code>buf[k] = '¥0'</code> | エ | <code>buf[tb] = '¥0'</code> |

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.2009 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

2 級

<正答>

問1

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

問2

| | | | | | |
|-----|------|------|------|------|------|
| (9) | (10) | (11) | (12) | (13) | (14) |
| イ | エ | ア | エ | イ | オ |

問3

| | | | | |
|------|------|------|------|------|
| (15) | (16) | (17) | (18) | (19) |
| ア | イ | エ | ウ | イ |

問4

| | | | | | |
|------|------|------|------|------|------|
| (20) | (21) | (22) | (23) | (24) | (25) |
| イ | ア | ア | イ | オ | エ |

問5

| | | | | |
|------|------|------|------|------|
| (26) | (27) | (28) | (29) | (30) |
| エ | イ | ア | ウ | イ |

問6

| | | | | |
|------|------|------|------|------|
| (31) | (32) | (33) | (34) | (35) |
| ア | オ | エ | ア | ウ |

問7

| | | | | |
|------|------|------|------|------|
| (36) | (37) | (38) | (39) | (40) |
| ウ | ア | ア | ウ | エ |

問8

| | | | | |
|------|------|------|------|------|
| (41) | (42) | (43) | (44) | (45) |
| エ | ア | イ | イ | ウ |