

## 修了試験「C 言語基礎 1」(1-A 第 2 回) 解答例

\_\_\_\_\_/100

出席番号 \_\_\_\_\_

社員番号 \_\_\_\_\_

氏名 \_\_\_\_\_

### 【問題 1】 (各 1 点、計 5 点)

以下の文は正しいか、解答欄に正しい場合には○、誤っている場合には×を記入してください。

- (1) #include ディレクティブはセミコロン(;)で止めなければならない。 × (P. 133)
- (2) すべての関数の外側で宣言した変数をローカル変数と呼びます。 × (P. 140)
- (3) 値を返さない関数では戻り値の型を void 型とします。 ○ (P. 135)
- (4) 関数プロトタイプとは、関数を使用される前に関数の定義に先行して関数を宣言するものです。 ○ (P. 132)
- (5) 関数の引数とは関数を呼び出すときに関数へ渡す値のことです。 ○ (P. 130)

### 【問題 2】 (各 1 点、計 4 点) (P. 78)

以下の式の値を 16 進数で答えなさい。ただし a=0x5a とする。

- (1) a & 0xFF 0x5a
- (2) a | 0 0x5a
- (3) a | 0xa5 0xff
- (4) a ^ a 0x0

### 解説

a について、ビット変換する。また、a 以外の値も変換しておく。

0x5a → 01011010

0xff → 11111111

0xa5 → 10100101

0 → 00000000

- (1) & はビット演算子のビット AND…両方(左辺、右辺)のビットが共に「1」の場合に「1」

01011010 左辺

11111111 右辺

**01011010 結果 → 0x5a**

- (2) | はビット演算子のビット OR…両方(左辺、右辺)のビットのどちらかが「1」の場合に「1」

01011010 左辺

00000000 右辺

01011010 結果 -> 0x5a

(3) (2)と同様。

01011010 左辺

10100101 右辺

11111111 結果 -> 0xff

(4) ^ はビット演算子のビット XOR…両方(左辺、右辺)のビットの値が異なる場合に「1」

01011010 左辺

01011010 右辺

00000000 結果 -> 0x0

【問題 3】 (各 1 点、計 8 点) (P. 34)

以下に示すエスケープ文字を答えなさい。

(1) ニル文字	<u>¥0</u>
(2) 改行文字	<u>¥n</u>
(3) 水平タブ文字	<u>¥t</u>
(4) バックスペース	<u>¥b</u>
(5) 2 重引用符	<u>¥"</u>
(6) 引用符	<u>¥'</u>
(7) ベル	<u>¥a</u>
(8) 円マーク文字	<u>¥¥</u>

【問題 4】 (各 1 点、計 8 点) (P. 62)

以下に示すプログラムは、2つの整数値を入力して、それに対して関係演算と論理演算を行い、それぞれの演算結果を表示するプログラムです。以下のプログラムへ穴埋めをしてください。

<プログラム>

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

```
int main(void)
```

```
{
```

```
    int i, j;
```

```
    printf("第 1 の整数を入力してください。: ");
```

```
    scanf("%d", &i);
```

```
    printf("第 2 の整数を入力してください。: ");
```

```
    scanf("%d", &j);
```

```
    /* 関係演算 */
```

```
    if (i ) { printf("i は j より小さいです¥n"); }
```

```
    if (i ) { printf("i は j より小さいか等しいです。¥n"); }
```

※ { } で括ります。  
(MISRA-C)

```
if (i  j) { printf("i は j と等しいです。¥n"); }  
if (i  j) { printf("i は j より大きいです¥n"); }  
if (i  j) { printf("i は j より大きいか等しいです。¥n"); }
```

```
/* 論理演算 */  
printf("i AND j : %d¥n", i  j);  
printf("i OR j : %d¥n", i  j);  
printf("NOT i : %d¥n",  );
```

```
return 0;
```

```
}
```

(1) <u>&lt;</u>	(5) <u>&gt;=</u>
(2) <u>&lt;=</u>	(6) <u>&amp;&amp;</u>
(3) <u>==</u>	(7) <u>  </u>
(4) <u>&gt;</u>	(8) <u>!i</u>

【問題 5】 (各 1 点、計 5 点) (P. 89・92・122・124)

以下に示すプログラムはユーザーの入力する値を集計するプログラムです。

新しい値を集計する前に、ユーザーに整数値を2回入力させて、入力された値が正しいことを確認し、2回目の値が1回目と合致しなければ、もう一度やり直します。以下のプログラムへ穴埋めをしてください。

<プログラム>

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

```
int main(void)
```

```
{
```

```
    int total, i, j;
```

```
    total = 0;
```

```
    do {
```

```
        printf("数を入力してください (0で終了): ");
```

```
        scanf("%d", &i);
```

```
        printf("もう一度、同じ数を入力してください: ");
```

```
        scanf("%d", &j);
```

```
        if (  ) {
```

```
            printf("入力した数が合致しません¥n");
```

```
             ;
```

```
        }
```

```
        total =  ;
```

```
    } while (  );
```

```
printf("合計: %d¥n", (5));  
  
return 0;  
}
```

(1) i != j

(2) continue

(3) total + i

(4) i

(5) total

以降の問題は実際にプログラムを行い、プログラムソース、実行ファイルは指定した場所に保存してください。

【問題 6】 (15 点) (P. 89・92・110)

各部屋の寸法から家屋の床面積を計算するプログラムを作成してください。  
最初に部屋の数を入力させ、その後で各部屋の寸法を順次入力させるものとします。

<実行例>

部屋の数を入力してください : 2

長さを入力してください : 2

幅を入力してください : 3

長さを入力してください : 4

幅を入力してください : 5

総床面積 : 26

```
#include <stdio.h>

int main(void)
{
    int rooms, len, width, total;
    int i;

    printf("部屋の数を入力してください : ");
    scanf("%d", &rooms);

    total = 0;
    for (i = rooms; i > 0; i--) {
        printf("\n 長さを入力してください : ");
        scanf("%d", &len);

        printf(" 幅を入力してください : ");
        scanf("%d", &width);

        total = total + len * width;
    }
    printf("総床面積 : %d", total);

    return 0;
}
```

【問題 7】 (15 点) (P. 89・92・110)

正の整数値を入力し、入力した値が素数かどうか判定して、その結果を表示するプログラムを作成してください。0 で終了するものとする。

<実行例>

正の整数を入力してください : 13

その数は素数です。

正の整数を入力してください : 15

その数は素数ではありません。

正の整数を入力してください :

```
#include <stdio.h>

int main(void)
{
    int i, j, prime;

    do {
        printf("正の整数を入力してください : ");
        scanf("%d", &i);

        prime = 1;
        for (j = 2; j <= i / 2; j++)
            if (!(i % j))
                prime = 0;
        if (prime)
            printf("%d は素数です\n", i);
        else
            printf("%d は素数ではありません\n", i);
    } while (i);

    return 0;
}
```

【問題 8】 (15 点) (P. 89-92-110)

次のメニューを表示して選択された操作を実行し、ユーザーが終了を選ぶまでそれを繰り返すプログラムを作成してください。 ※ 1メートルは3.28フィート、1ポンドは16オンスです。

変換

1. フィートからメートルへ
2. メートルからフィートへ
3. オンスからポンドへ
4. ポンドからオンスへ
5. 終了

番号を選んで入力してください：

```
/* apdC-052.c */
#include <stdio.h>

int main(void)
{
    int i;
    float feet, meters, ounces, pounds;

    do {
        printf("変換\n\n");
        printf("1. フィートからメートルへ\n");
        printf("2. メートルからフィートへ\n");
        printf("3. オンスからポンドへ\n");
        printf("4. ポンドからオンスへ\n");
        printf("5. 終了\n\n");
        do {
            printf("番号を選んで入力してください： ");
            scanf("%d", &i);
        } while (i < 0 || i > 5);

        switch (i) {
            case 1:
                printf("フィート数を入力してください： ");
                scanf("%f", &feet);
                printf("メートル： %f\n", feet / 3.28);
                break;
            case 2:
                printf("メートル数を入力してください： ");
                scanf("%f", &meters);
                printf("フィート： %f\n", meters * 3.28);
                break;
```

```
        case 3:
            printf("オンス数を入力してください: ");
            scanf("%f", &ounces);
            printf("ポンド数: %f\n", ounces / 16);
            break;
        case 4:
            printf("ポンド数を入力してください: ");
            scanf("%f", &pounds);
            printf("オンス: %f\n", pounds * 16);
            break;
    }
} while (i != 5);

return 0;
}
```

**【問題 9】** (15 点) (P. 89・92・110)

数当てゲームを作成してください。プログラムの仕様は、以下のとおりです。プレイヤーは謎の数を当てるチャンスが10回あります。プレイヤーの入力した数と一致すれば「当たり！」と表示してプログラムを終了します。当たらなかった場合は入力した数が謎の数より大きい小さいかを知らせ、別の数を入力するように促します。このプロセスはプレイヤーが数を当てるか、10回のチャンスを使い切るまで続きます。

```
#include <stdio.h>

int main(void)
{
    int magic; /* マジックナンバー */
    int guess; /* ユーザーが当てる */
    int i;

    magic = 1325;
    guess = 0;

    for (i = 0; i < 10 && guess != magic; i++) {
        printf("考えた数字を入力してください: ");
        scanf("%d", &guess);

        if (guess == magic) {
            printf("正解!");
            printf(" %d はマジックナンバーです。%n", magic);
            break;
        }
    }
}
```



```
        else {
            printf("残念、違います。");
            if (guess > magic)
                printf("さらに小さい数字です。¥n");
            else
                printf("さらに大きい数字です。¥n");
        }
    }

    return 0;
}
```

**【問題 10】 (10 点) (P. 89-144)**

再帰を使って階乗を計算するプログラムを作成してください。ある数の階乗を求めるには、ある数にその数より小さい数から 1 まですべての整数を掛け合わせます。i n t 型引数の階乗を再帰を使って計算する f a c t ( ) という関数を作成し、この関数をプログラムの中で実際に使ってください。

```
/* apdC-093.c */
#include <stdio.h>

int fact(int i);

int main(void)
{
    printf("5 の階乗は %d です", fact(5));
    return 0;
}

int fact(int i)
{
    if (i == 1)
        return 1;
    else
        return i * fact(i - 1);
}
```