C 言語プログラミング ＆ アルゴリズム

演習課題 vol.8　繰り返し処理 I

1. 問題は基本問題とチャレンジ問題に分かれています。
   ※印のついている問題はチャレンジ問題です。時間のある人はやりましょう。
   チャレンジ問題を提出した場合、評価に反映します。

2. 提出したソースファイルは教員が回収するのでデータが残りません。必ず GoogleDrive のマイドライブもしくは USB メモリに保存してバックアップして下さい。

3. 演習で作成したソースファイル名は課題番号にしてください。
   例）ex0101.c ならファイル名は、ex0101.c になります。

4. 右の例のように先頭にソースコードの先頭に自分の学籍番号と名前をコメントで入れてからプログラムを書き始めましょう。

```
/*
  00EO0199 電子太郎
*/
#include <stdio.h>
```

5. GoogleDrive の下記の共有フォルダが提出先です。
   そこへ作成したソースファイル (.c) をコピーして提出してください。

   C 言語プログラミング_自分の学籍番号

   例）学籍番号が 23EO0199 なら「C 言語プログラミング_23eo0199」

6. 書き換え問題はソースファイルをコピーして、必ず別々のファイルにしてください。

■ ex0801.c

変数 i の値をカウントアップして連番を作りたい。ソースコードのコメントと実行結果を参考に (A)〜(D) の空欄を穴埋めしてプログラムを完成させなさい。
なお連番は実行結果と同じようにスペースで区切り、制御式の条件はコメントの通りに書くこと。

```
#include <stdio.h>

int main(void) {
  // 整数型の変数 i を宣言
  (A) ;  // ループカウント（0 で初期化）

  while( (B) ) { // i が 12 より小さい間はループ

    // 変数 i の値を表示（ループすることで連番表示）
    (C)

    // インクリメント演算子で i の値に+1 する
    (D)
  }

  // 文末で改行を行う
  putchar('¥n');

  return 0;
}
```

実行結果

```
0  1  2  3  4  5  6  7  8  9  10  11
```

■ ex0802.c

ex0801.c のプログラムを、while 文を用いた処理から for 文を用いた処理に修正しなさい。

実行結果
※ex0801.c と同じ

■ ex0803.c

ex0802.c のプログラムを修正して、連番の数値を実行結果と同じように 2 桁表示にしなさい。なおループは for 文のままにすること。

実行結果
00　01　02　03　04　05　06　07　08　09　10　11

■ ex0804.c

ex0502.c のプログラムを修正して、vy に 0 が入力された場合に再入力出来るようにしたい。vy の再入力処理を do-while 文で作成しプログラムを修正しなさい。
なお、表示するメッセージなども実行結果を参考に修正すること。

実行結果
vx>5 ⏎ vy>0 ⏎ vy には 0 以外の値を代入してください。 vy>0 ⏎ vy には 0 以外の値を代入してください。 vy>2 ⏎ vx / vy = 2.500

■ ex0805.c ※

for 文と rand 関数を使い、10 個の乱数を生成し表示するプログラムを作成しなさい。
なお、表示結果と同じになるように表示処理を行い、以下の変数宣言と条件を守ること。

変数名	型	初期化	意味
i	int	初期化なし	ループ回数カウント
rand_val	double	初期化なし	乱数を一時的に格納

＜条件＞

① 生成する乱数は整数部が最大 3 桁で小数部が最大 2 桁の小数の乱数を生成
② 生成した乱数は表示する前に一度 rand_val に代入して、その rand_val の値を表示（生成した値を再利用するケースを想定した処理）
③ ループは必ず i = 0 から始め、実行結果では 1 から表示されるようにする

＜ヒント＞

▷ 乱数で 5 桁の整数を生成し、それになんらかの演算を行うことで小数の乱数にすること。

実行結果 ※実行するごとに下記の数値は変わります
1 回目 : 294.00 2 回目 : 4.99 3 回目 : 79.55 4 回目 : 173.40 5 回目 : 247.98 6 回目 : 114.02 7 回目 : 151.96 8 回目 : 293.38 9 回目 : 240.45 10 回目 : 168.50

■ ex0806.c ※

for 文で i が 1〜100 までのループ処理を行い、i の値が 3 の倍数の時は Fizz、5 の倍数の時は Buzz、3 と 5 の倍数の時は FizzBuzz と表示するプログラムを作成しなさい。

実行結果

```
1
2
Fizz
4
Buzz
Fizz
7
8
Fizz
Buzz
11
Fizz
13
14
FizzBuzz
16

※ 長いので省略

94
Buzz
Fizz
97
98
Fizz
Buzz
```


■ ex0807.c ※

for 文で 0 から入力したループ回数までの数値を 2 つ飛ばしで表示するプログラムを作成しなさい。ただし、カウントアップ処理には複合代入演算子を使い、for 文内では条件分岐処理（if-else 文や switch 文）は使用しないこと。

またループ回数が 0 以下の場合は再入力させるようにし、表示内容については実行結果と同じようにすること。なお再入力処理のメッセージの出力には if 文を使って良いものとする。

実行結果

```
ループの上限を入力>0 ⏎
入力する数は 1 以上にしましょう。
ループの上限を入力>30 ⏎
00 03 06 09 12 15 18 21 24 27 30
```

■ ex0808.c ※

ex0707.c を修正して、3 勝するか 3 敗するまでじゃんけんを続けるプログラムに修正しなさい。なお何勝何敗したのか実行結果を参考に表示すること。

実行結果 1 ※勝った場合 / 実行するごとにじゃんけんの手は変わります

```
じゃんけん…（0：グー 1：チョキ 2：パー）
あなたの手>2 ⏎
あいての手：チョキ
あなたの負けです。
じゃんけん…（0：グー 1：チョキ 2：パー）
あなたの手>1 ⏎
あいての手：チョキ
あいこです。
じゃんけん…（0：グー 1：チョキ 2：パー）
あなたの手>0 ⏎
あいての手：チョキ
あなたの勝ちです。
じゃんけん…（0：グー 1：チョキ 2：パー）
あなたの手>1 ⏎
あいての手：パー
あなたの勝ちです。
じゃんけん…（0：グー 1：チョキ 2：パー）
あなたの手>1 ⏎
あいての手：パー
あなたの勝ちです。
----------
3 勝 1 敗であなたの勝ちです。
```

実行結果 2 ※負けた場合 / 実行するごとにじゃんけんの手は変わります

```
じゃんけん…（0：グー 1：チョキ 2：パー）
あなたの手>1 ⏎
あいての手：パー
あなたの勝ちです。
じゃんけん…（0：グー 1：チョキ 2：パー）
あなたの手>1 ⏎
あいての手：パー
あなたの勝ちです。
じゃんけん…（0：グー 1：チョキ 2：パー）
あなたの手>1 ⏎
あいての手：グー
あなたの負けです。
じゃんけん…（0：グー 1：チョキ 2：パー）
あなたの手>1 ⏎
あいての手：グー
あなたの負けです。
じゃんけん…（0：グー 1：チョキ 2：パー）
あなたの手>1 ⏎
あいての手：グー
あなたの負けです。
----------
2 勝 3 敗であなたの負けです。
```