プログラミング１　第８回教材　　担 当　神保 雅人

2013 年 6 月 6 日（木）実施

【前回の補遺】前回の教材で扱った０回以上の繰り返しについて，例題を加える。

例題１（for 文を用いたアンケートの集計）

次のソースプログラムをテキストエディタで入力して，prog8-1.c の名前を付けて保存する。それを翻訳・編集して実行形式のファイルを作成し，実行せよ。

```
/* prog8-1.c */
#include <stdio.h>
#define KOUMOKU 8

int main()
{
    char s[KOUMOKU+1];
    int i,sa[KOUMOKU]={0,0,0,0,0,0,0,0},si[KOUMOKU]={0,0,0,0,0,0,0,0};
    int su[KOUMOKU]={0,0,0,0,0,0,0,0},se[KOUMOKU]={0,0,0,0,0,0,0,0};
    int sm[KOUMOKU]={0,0,0,0,0,0,0,0},sg[KOUMOKU]={0,0,0,0,0,0,0,0};

    printf("半角アルファベット aiue,または半角空白で８文字分ずつ入力してください。¥n");
    printf("________¥r");

    while(fgets(s,KOUMOKU+1,stdin)!=NULL)
    {
        for(i=0;i<KOUMOKU;i++)
        {
            switch(s[i])
            {
                case 'a':
                case 'A': sa[i]++; break;
                case 'i':
                case 'I': si[i]++; break;
                case 'u':
                case 'U': su[i]++; break;
                case 'e':
                case 'E': se[i]++; break;
                case ' ': sm[i]++; break;
                default:  sg[i]++;
            }
        }
        rewind(stdin);
    }

    printf("¥n¥n");

    for(i=0;i<KOUMOKU;i++)
    {
      printf("(%1d) ア %3d 件｜イ %3d 件｜ウ %3d 件｜エ %3d 件｜"
       "無答 %3d 件｜誤答 %3d 件｜合計 %3d 件¥n"
       ,i+1,sa[i],si[i],su[i],se[i],sm[i],sg[i],sa[i]+si[i]+su[i]+se[i]+sm[i]+sg[i]);
    }

    return 0;
}
```

【解説】

1. sa，si，su，se，sm，sg は，８項目(KOUMOKU で指定)から成るアンケートを集計するためのカウンタであり，それぞれ８個の要素を持つ配列である。

2. ¥r は復帰(return)で，カーソルを行頭に戻すための制御コードを表す。

3. fgets は，第３引数で指定した入力ストリームから第２引数で指定したバイト数分の文字列を読み込み，第１引数で指定した配列に格納するライブラリ関数である。

4. stdin は標準入力ストリームを表す。

5. rewind は入力ストリームの位置を先頭に戻すライブラリ関数である。

6. データの入力終了時にはコントールキーを押しながら Z を打ち，Enter キーを打つことで繰り返しから抜ける。

１回以上の繰り返しのプログラム

do 文

do 文の構文は次のようになる。

```
do 文 while ( 継続条件式 );
```

先ず繰り返しの対象となる文を実行し，続いて継続条件式を評価し，真であれば再度繰り返しの対象となる文を実行し，継続条件式を評価する，という繰り返しを行う。継続条件式が真でなくなれば，繰り返しの対象となる文を実行せず，do 文から抜け出す。最初から継続条件式が真でなくとも，先ず繰り返しの対象となる文を実行するので，１回以上の繰り返しと呼ばれる。

do 文は通常，**ある処理を実施し，その処理を特定の状態になるまで繰り返す目的**で用いられる。例えば，数値計算に於いて処理を繰り返すことにより，数値が収束していく場合がその典型的な例である。

例題２（**do 文を用いた特定の処理の決まった回数の繰り返し**）

次のソースプログラムをテキストエディタで入力して，prog8-2.c の名前を付けて保存する。それを翻訳・編集して実行形式のファイルを作成し，実行せよ。

```
/* prog8-2.c */
#include <stdio.h>
#include <string.h>
#define NAGASA 15
#define NINZUU 10

int main(void)
{
```

```
    int i=0,j;
    char namae[NINZUU][NAGASA], denwa[NINZUU][NAGASA];

    do
    {
        printf("%d 番目の人の名前を入力してください：", i+1);
        scanf("%s", namae[i]);

        if (strncmp(namae[i],"quit",4) == 0) break;

        printf("%d 番目の人の電話番号を入力してください：", i+1);
        scanf("%s", denwa[i]);

        i++;

    }while (i < NINZUU);

    if (i > 0) printf("¥n 電話番号簿¥n");

    for (j=0; j<i; j++)
        printf("%2d) 名前：%-14s  電話番号：%-14s¥n", j+1, namae[j], denwa[j]);

    return 0;
}
```

【注意】これまで，scanf の書式指定を単に"%s"としているが，実用的なプログラムを作成する場合には，これでは想定する文字数を超える文字列の入力に対処できないので，書式指定で文字数を指定した上で，更に工夫が必要となる。(対処しない場合にはセキュリティホールを生じる)
或いは，例題１のように fgets を用いる。

例題３（do 文を用いた特定の状態になるまでの処理の繰り返し）

次のプログラムを入力し，翻訳・編集して実行形式のファイルを作成し，実行せよ。ここで，ソースプログラム名は prog8-3.c とする。

```
/* prog8-3.c */
#include <stdio.h>
#define EPSILON 1.0e-9

int main(void)
{
    int n=0;
    double x=1.0;

    do
    {
        x = x / 2;

        n++;

    }while (x >= EPSILON);
```

```
    printf("1/2 の%d 乗は%.1e より小さい。¥n", n, EPSILON);

    return 0;
}
```

【解説】

1. printf 中の変換指定 %.1e は精度（小数点以下の桁数）を 1 桁として指数形式の 10 進表現に変換する。なお，%e では精度は 6 桁に設定されている。

2. 10^3 (k) に近い値として，$2^{10} = 1024$ がコンピュータ関係ではよく用いられる。また，10^6 (M)，10^9 (G) に近い値としてはそれぞれ，$2^{20} = 1024 \times 1024$，$2^{30} = 1024 \times 1024 \times 1024$ がコンピュータ関係ではよく用いられる。

演習 1

do 文を用いて整数 n の階乗 n! = n×(n-1)×・・・×1 を計算し，n の階乗の値が 10000 を上回るような n を求めるプログラムの空欄 1)　　，2)　　，3)　　を埋めてソースプログラムを完成させ，テキストエディタで入力して，ex8-1.c の名前を付けて保存する。それを翻訳・編集して実行形式のファイルを作成し，実行せよ。

［ヒント： 例えば，整数の変数を int n=0; のように宣言時に初期化しておき，階乗を求めるための変数を int f=1; と宣言した上で，f=n*f; を，n の値を 1 ずつ増やしながら繰り返す。
（定義からは 1!=1, 2!=2×1=2, 3!=3×2×1=6, … であるが，2!=2×1!, 3!=3×2!, … を利用)］

```
/* ex8-1.c */
#include <stdio.h>
1)      MAX 10000

int main(void)
{
    int n=0,f=1;

    do
    {
        2)      ;
        f=n*f;
        printf("%d の階乗は%d¥n", n, f);
    }while (f <= 3)      );

    printf("%d の階乗は%d より大きい。¥n", n, MAX);

    return 0;
}
```

提出物：

1) 例題 1，2，3 の出力結果をコピーして貼り付けたテキストファイル res8.txt

2) 演習 1 のソースプログラムのファイル ex8-1.c の完成版