

2006年11月30日(木)実施

前回 演習2 解答例

```
/* ex6-2.c */
#include <stdio.h>
#include <string.h>
#define BUF_LEN 15
#define MAX 10

int main(void)
{
    int i=0, j;
    char name[MAX][BUF_LEN], phone[MAX][BUF_LEN];

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

    while (strncmp(name[i], "quit", 4) != 0)
    {
        printf("%d 番目の人の電話番号を入力してください：", i+1);
        scanf("%s", phone[i]);

        i++;
        if (i == MAX) break;

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

    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, name[j], phone[j]);

    return 0;
}
```

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 );
```

先ず文を実行し、続いて継続条件式を評価し、真であれば再度文を実行し、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、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 1

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 prog7-1.c とする。

```
/* prog7-1.c */
#include <stdio.h>
#include <string.h>
#define BUF_LEN 15
#define MAX 10

int main(void)
{
    int i=0, j;
    char name[MAX][BUF_LEN], phone[MAX][BUF_LEN];

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

        if (strncmp(name[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", phone[i]);

        i++;
    }while (i < MAX);

    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, name[j], phone[j]);

    return 0;
}
```

例題 2

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 prog7-2.c とする。

```
/* prog7-2.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。 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

整数 n の階乗 $n! = n \times (n-1) \times \dots \times 1$ 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を、do 文を用いて作成せよ。ここで、ソースプログラム名は `ex7-1.c` とする。

[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+1)*f;` を、 n の値を 1 ずつ増やしながら繰り返す。]