

2006年11月16日(木)実施

0回以上の繰り返しのプログラム

for 文

C 言語で 0 回以上の繰り返しのプログラムを実現するための文として、while 文と for 文が用意されている。for 文の構文は次のようになる。

```
for ( 初期設定式; 継続条件式; 再設定式 ) 文
```

まず、初期設定式を実行し、続いて継続条件式を評価する。ここで、継続条件式が真であれば、文を実行する。次に再設定式を実行した上で、再度、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ければ、文を実行せず、for 文から抜け出す。最初から継続条件式が真でなければ、全く文を実行しないので、0回以上の繰り返しと呼ばれる。

for 文は通常、特定の処理を繰り返す目的で用いられる。

while 文

while 文の構文は次のようになる。

```
while ( 継続条件式 ) 文
```

まず、継続条件式を評価し、真であれば文を実行し、再度、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ければ、文を実行せず、while 文から抜け出す。最初から継続条件式が真でなければ、全く文を実行しないので、0回以上の繰り返しと呼ばれる。

while 文は通常、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、ファイルを終わりまで読む場合や、キーの入力待ちを行う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 prog6-1.c とする。

```
/* prog6-1.c */
#include <stdio.h>
#define NUM 5

int main(void)
{
    int i;
    double dat[NUM], sum=0.0, ave;

    for (i=0; i<NUM; i++)
    {
        printf("%d 番目のデータを入力してください: ", i+1);
        scanf("%lf", &dat[i]);
        sum=sum+dat[i];
    }
}
```

```
ave = sum/NUM;

for (i=0; i<NUM; i++)
    printf("%d 番目のデータ : %f\n", i+1, dat[i]);

printf("データの合計 : %f\n", sum);
printf("データの平均 : %f\n", ave);

return 0;
}
```

【解説】

1. #define はマクロ定義を行うためのプリプロセッサ命令である。このプログラムで書かれている #define NUM 5 は、プログラムが翻訳される際に、プログラム中の NUM を全て 5 で置き換えることを要求している。
2. ++ は増分演算子と呼ばれ、i++ は i の値を 1 だけ増加させる。

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 prog6-2.c とする。

```
/* prog6-2.c */
#include <stdio.h>
#include <string.h>
#define BUF_LEN 80
#define MAX 5

int main(void)
{
    int i, j;
    char buf[BUF_LEN], buf2[BUF_LEN];

    printf("半角のアルファベットまたは数字で\n 文字列を入力してください： ");
    scanf("%s", buf);

    while (strncmp(buf, "quit", 4) != 0)
    {
        for (j=0; j<MAX; j++)
        {
            i=0;
            while (buf[i+1]!='\0')
            {
                buf2[i]=buf[i+1];
                i++;
            }

            buf2[i]=buf[0];
            buf2[i+1]='\0';
            printf("buf2 : %s\n", buf2);

            strncpy(buf, buf2, i+1);
        }
    }
}
```

```
        printf("次の文字列を入力してください: ");
        scanf("%s", buf);
    }

    return 0;
}
```

【解説】

1. string.h は文字列操作のためのライブラリ関数 strcmp 及び strcpy を利用するために組み込んでおく必要がある。
2. strcmp(文字列 1, 文字列 2, 比較文字数) は文字列 1 と文字列 2 とを先頭からの比較文字数分だけ比較するライブラリ関数である。strcmp(buf, "quit", 4) であれば, buf に quit から始まる文字列が格納されているとき, 0 が返される。
3. strcpy(複写先配列, 複写元文字列, 複写文字数) は複写元文字列から複写文字数文だけの文字列を複写先配列に複写するライブラリ関数である。

演習 1

10 人分の成績判定を行うプログラムを, for 文を用いて作成せよ。ここで, ソースプログラム名は ex6-1.c とする。なお, 成績判定の仕方は prog5-1.c に準ずるものとする。

演習 2

ある人数分の名前と電話番号とを入力し, 一覧を表示するプログラムを, while 文を用いて作成せよ。なお, 名前と電話番号とはそれぞれ別の char 型の 2 次元配列 ([人数]×[文字数+1]) に格納するものとし, 名前の入力の際に quit とすると終了するものとする。ただし, 人数は最大限 10 名までとし, #define MAX 10 で定義して, 配列の要素数には MAX を用いる。while 文の中では配列の添え字を i とし, i が MAX に達した場合には, break 文を実行して, while 文から抜出すものとする。ここで, ソースプログラム名は ex6-2.c とする。