

## do-while 文

これまでの問題で、`scanf` が while ループの外と中の両方にして出てきて、見栄えがあまりよくなかったことがあった。それを do-while 文を使ってすっきり書いてみよう。

while 文では条件をみて、条件が成り立つ時だけ本体を実行する。一方、do-while 文では、本体を実行してから条件をチェックする。

構文としては以下のとおり。

```
do {  
    ...  
    本体;  
    ...  
} while (条件);
```

例 1 : 0 が入力されるまで整数を入力し、総和を求める。

```
#include <stdio.h>  
main()  
{  
    int n, sum=0;  
  
    do {  
  
        scanf("%d", &n);  
  
        sum = sum + n;  
  
    } while (n != 0);  
  
    printf("%d\n", sum);  
}
```

例 2 (例 1 と同じような問題だが、総和を求めるのではなく、全入力を配列に格納する)

```
1 #include <stdio.h>
2 main()
3 {
4     int i, n, a[1000];
5
6     i = 0;
7     do {
8         printf("a[%d] = ", i);
9         scanf("%d",&a[i]);
10    } while ( a[i++] != 0 );
11
12   for ( i--, i--; i>=0; i-- ) {
13       printf("a[%d] = %d\n", i, a[i]);
14   }
15 }
```

(注意) `a[i++]` はその時点の変数 `i` の値を元に `a[i]` を決定してから、`i` の値を 1 増やす。一方、`a[++i]` は変数 `i` の値を 1 増やしてから、`a[i]` を決定する。