

fscanf の戻り値

前にほとんどの関数に戻り値がある説明をしました。fscanf (scanf も同じ) も、戻り値をもっています。それは、**入力で、設定できたデータの個数**です。次のプログラムで実験します。プリントに従って、プログラムを作成して実験しましょう。(問題1として ファイル tH01.c)

```
#include <stdio.h>
main()
{
    int a=1;
    int b=2;
    int c=3;
    int d=4;
    int n;
    fprintf(stdout, "2つ入力>");
    n = fscanf(stdin, "%d%d", &a, &b );          /* 1 */
    fprintf(stdout, "%d個入力できました\n", n);
    fprintf(stdout, "%d, %d\n", a, b);
    /* 2 */
    fprintf(stdout, "2つ入力>");
    n = fscanf(stdin, "%d%d",&c , &d );          /* 3 */
    fprintf(stdout, "%d個入力できました\n", n);
    fprintf(stdout, "%d, %d\n", c, d);
}
```

3通りの入力で、実験した結果を示します。

```
2つ入力>10 20
2個入力できました
10, 20
2つ入力>30 40
2個入力できました
30, 40
```

```
2つ入力>10 20 30 40
2個入力できました
10, 20
2つ入力>2個入力できました
30, 40
```

```
2つ入力>10 a 30 40
1個入力できました
10, 2
2つ入力>0個入力できました
3, 4
```

中央の実行例では、最初の入力で 四つのデータを入力しています。標準入力バッファに記憶されたデータから/*1*/の fscanf で取り出すのは2つだけで、残りはバッファに残ります。

よって、/*3*/の fscanf で入力待ちにならずに、残っているバッファから2個取り出して変数に設定しているのが分かります。

右の実行例では、%d の数字入力指定と合わない **a** の入力に変数に設定できず、入力個数が1になっています。次の/*3*/ではバッファにデータが残っているので入力待ちにならないのですが、%d の指示と合わない **a** がバッファに残っているので、読み取りできずに0個の入力になっています。なお、/*2*/ に scanf ("%na"); を追加記述すると、/*3*/ で、30と40の2個が入力できるようになります。

"%d%d" や "%na" は、scanf の**入力用フォーマット**と呼ばれ、標準バッファからの読み取り指示になります。そして%d や%lf は、読み取ったデータを次の引数で指定している記憶域へ設定をする機能が伴う指示になっているのです。この中にある**半角のスペース**や\nが1つ以上ある指示は、『半角スペースや、改行やタブの文字』でないところまでバッファを読み取る指示です。この『半角スペースや、改行やタブの文字』は、**空白文字**と呼ばれます。

つまり、"**%na**"は、バッファの中の半角スペースや、改行やタブを取り除き、バッファに残っている **a** の文字を取り除く指示です。もしこのフォーマットが合わない指示の場合、この scanf は終わって次に進みます。例えばバッファに空白文字がなければ次に進みます。空白文字があれば読み取り除き、それに続く **a** の文字があれば読み取り除くというわけです。

fscanf の戻り値を利用すると、次のような繰り返しで、連続するデータを処理できます。

```
while( fscanf(stdin, "%d" , &d) == 1){
    printf("d=%d ", d);
}
```

実行例

```
10
d=10
20
d=20
30
d=30
40
d=40
abc
```

実行例

```
10 20 30 40 abc
d=10 d=20 d=30 d=40
```

下線部が入力です。

問題2 前述のような繰り返しを使って、次の実行例のように合計の結果を算出せよ。

実行例 (ファイル名 **tH02.c**)

```
入力>10 20 30 40 50 =
10 + 20 + 30 + 40 + 50 = 150
```

最初は、+ の表示を出さないプログラムを作ると、分かりやすいです！

```
入力>10 20 30 40 50 =
10 20 30 40 50 = 150
```

それから + の表示を if で表示させるとよいでしょう。

次の解説 さてここで、再び問題1 (tH01.c) のファイルを変更して実験を続けます。

```
/* 2 */ に追加した fscanf("%na"); を削除して、コメント/*1*/と/*3*/を次のように変更します。
n = fscanf(stdin, "%d + %d", &a, &b ); /* 1 */
n = fscanf(stdin, "%d - %d",&c , &d ); /* 3 */
そして、各変数 a, b, c, d に設定できる入力を行わせてみましょう。
```

なお、コメント1の入力用フォーマット指示は次のようになります。
標準入力バッファが空なら入力待ちになる。バッファが空でない場合は次のように処理します。
『空白文字ならを取り出す。数字が出現するならを取り出して &a に設定する。次に空白文字なら取り出す。+が出現するなら取り出す。次に空白文字なら取り出す。数字が出現するならを取り出して &b に設定する。』この連続処理の過程で、ならの部分が合わないなら fscanf を終わらせます。またバッファが空になると、入力待ちになります。

問題3 以上の scanf の機能をよく知っていると次のように実行できるプログラムが作れます。チャレンジしてみましょう。(ファイル名 **tH03.c**)

```
入力>1 + 2 + 3 - 10 =
1+2+3-10 = -4
```

[余裕があればの問題4]
掛け算 * または / の演算を一つだけ入れた演算を可能にするように変更する。tH04.c
なお、演算子の優先順位を考慮する。
(複数の*や/を扱うのは2年後半か3年のレベルで今は無理です。)

問題2 解答例

```
#include <stdio.h>
main()
{
    int d; /*入力用 */
    int sum = 0; /* 合計記憶 */
    int n = 0;
    fprintf(stdout, "入力>");
    while( fscanf(stdin, "%d" , &d) == 1){
        if(n != 0) {
            fprintf(stdout, " + ");
        }
        n = n + 1;
        sum = sum + d;
        fprintf(stdout, "%d", d);
    }
    printf(" = %d\n", sum);
}
```

問題3 解答例

```
#include <stdio.h>
main()
{
    int d; /*入力用 */
    int sum = 0; /* 合計記憶 */
    int n;
    fprintf(stdout, "入力>");
    fscanf(stdin, "%d" , &sum);
    fprintf(stdout, "%d", sum);
    do {
        n = fscanf(stdin, " + %d" , &d);
        if(n == 1){
            sum = sum + d;
            fprintf(stdout, "+%d", d);
        } else {
            n = fscanf(stdin, " - %d" , &d);
            if( n == 1){
                sum = sum - d;
                fprintf(stdout, "-%d", d);
            }
        }
    }
    } while ( n == 1);
    printf(" = %d\n", sum);
}
```

問題4 解答例

```

#include <stdio.h>
main()
{
    int d; /* 入力用 */
    int d2 = 0; /* ループにおいて、前の入力用 */
    int flag = 1; /* ループにおいて、前の演算が+なら1 前が-なら-1 */
    int sum = 0; /* 合計記憶 */
    int n;
    fprintf(stdout, "入力>");
    fscanf(stdin, "%d", &sum);
    fprintf(stdout, "%d", sum);
    d2 = sum;
    do {
        n = fscanf(stdin, " + %d", &d);
        if(n == 1){
            sum += d;
            flag = 1;
            fprintf(stdout, "+%d", d);
        } else {
            n = fscanf(stdin, " - %d", &d);
            if( n == 1){
                sum -= d;
                flag = -1;
                fprintf(stdout, "-%d", d);
            } else {
                n = fscanf(stdin, " * %d", &d);
                if( n == 1){
                    sum += -flag * d2; //前の演算をなしにする
                    sum = sum + flag * d2 * d;
                    fprintf(stdout, "*%d", d);
                } else {
                    n = fscanf(stdin, " / %d", &d);
                    if( n == 1){
                        sum += -flag * d2; //前の演算をなしにする
                        sum = sum + flag * d2 / d;
                        fprintf(stdout, "/%d", d);
                    }
                }
            }
        }
        d2 = d; // 次のループに、データを d2 へ残す
    } while ( n == 1);
    printf(" = %d\n", sum);
}

/*
入力>3 + 15 / 3 =
3+15/3 = 8

入力>10 / 2 + 3 =
0/2+3 = 8

入力>2 * 4 + 1 =
2*4+1 = 9

入力>2 + 3 * 4 =
2+3*4 = 14

入力>2 - 3 * 4 =
2-3*4 = -10
*/

```