

【scanf 関数】

scanf 関数は標準入出力（Windows ではコマンドプロンプト）から文字列を取得する関数です。下記は scanf 関数の定義です。

定 義	<code>int scanf(const char* format, ...);</code>
ヘッダファイル	<code>stdio.h</code>
説 明	scanf 関数は標準入出力（コマンドプロンプト）から文字列を取得する関数。第 2 引数は複数指定できる。
戻 り 値	正常に取得できた値の個数。
例 文	<code>int var; scanf("%d", &var); /* 第 2 引数はポインタなので&を付けること */</code>

【書式指定子】

書式指定子を使用して表示形式を指定できます。

書式指定子の記述は「`%[代入抑止][最大入力数][型]`」です。

【型の指定子】

`%[代入抑止][最大入力数][型]`

キーボードから入力する値のデータ型を指定子で指定します。

指定子	入力したい形式
d	符号付き (signed)、10 進数の整数
u	符号無し (unsigned)、10 進数の整数
o (オー)	符号無し、8 進数の整数
x/X	符号無し、16 進数の整数
f	実数
c	文字 (1 文字)
s	文字列 (1 文字以上)

変数のデータ型によっては型の指定子の前に修飾子が付きます。

データ型	修飾子
short 型	h
long 型、double 型	l (エル)

printf 関数と異なり、double 型には「l (エル)」が付くので注意。

【複数の値を取得する場合】

1 度の scanf 関数で複数の値を取得したい場合は、下記のように記述します。

```
int var1;
int var2;
scanf("%d %d", &var1, &var2);    /* A スペース区切り */
scanf("%d,%d", &var1, &var2);    /* B カンマ区切り */
scanf("%d/%d", &var1, &var2);    /* C スラッシュ区切り */
```

A は %d と %d の間を「半角スペース」にしています。

この場合、キーボードから「10 20」と入力すると、変数 var1 に 10、変数 var2 に 20 が代入されます。

B は %d と %d の間を「, (カンマ)」にしています。

この場合、キーボードから「10,20」と入力すると、変数 var1 に 10、変数 var2 に 20 が代入されます。

C は %d と %d の間を「/ (スラッシュ)」にしています。

この場合、キーボードから「10/20」と入力すると、変数 var1 に 10、変数 var2 に 20 が代入されます。

指定子間の区切りは自由に指定でき、指定子と変数を 3 個、4 個と増やすことにより、入力できるデータ数も増やすことができます。

指定子は 2 個なのに 3 個の値を入力した場合、3 個目の値は「バッファ」(P35 参照)に残ります。

【最大入力数】

"%[代入抑止][最大入力数][型]"

取得する文字数を整数で指定します。

```
int var;
scanf("%5d", &var);    /* 最初の 5 文字だけ取得する */
```

キーボードから「123456789」と入力した場合、最初の 5 文字「12345」が変数 var に代入されます。6 文字目以降の文字は「バッファ」に残ります。

【代入抑止】

```
"%[代入抑止][最大入力数][型]"
```

* (アスタリスク) を入れた指定子は「代入抑止」、つまり代入されない指定子になります。

```
int var1;
int var2;
scanf("%d%c%d", &var1, &var2);    /* %c は代入抑止 */
```

キーボードから「10 20」「10-20」「10a20」と入力した場合、変数 var1 に「10」、変数 var2 に「20」が代入されます。10 と 20 の間の 1 文字 (%c は 1 文字) は代入対象とされません。

電話番号や郵便番号の入力等で利用されます。

【scanf 関数の戻り値】

scanf 関数の戻り値は「正常に取得できた値の個数」です。

下記のソースコードの scanf 関数は %d で整数を 2 個入力するように指定しています。

```
int var1;
int var2;
int ret;
ret = scanf("%d %d", &var1, &var2);
```

キーボードから入力した値と scanf 関数の戻り値

1 個目の 入力値	2 個目の 入力値	戻り値	解説
10	20	2	10 も 20 も整数なため、正しく取得される。
10	a	1	a は文字のため正しく取得できない。
a	b	0	2 個とも正しく取得できない。
a	10	0	1 個目のエラーで関数を終了するため、2 個目の値が取得できない。

scanf 関数の戻り値を見ることで、正常に値を取得できたかが分かります。

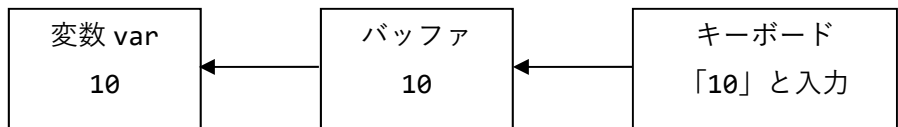
【バッファ】

キーボードから入力されたデータはメモリ内の「バッファ」なる領域に記録され、そこから変数に代入される仕組みになっています。

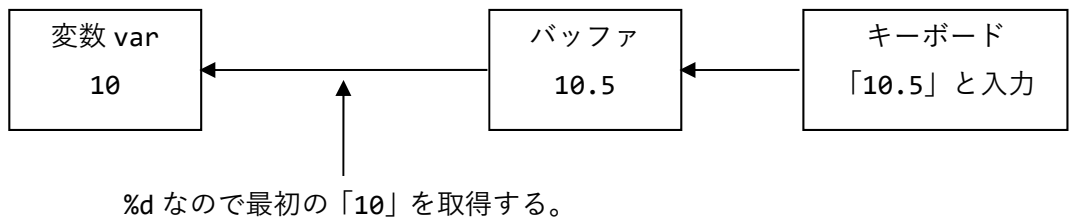
下記のソースコードを色々なパターンで実行してみます。

```
int var;
scanf("%d", &var);
printf("%d", var);
```

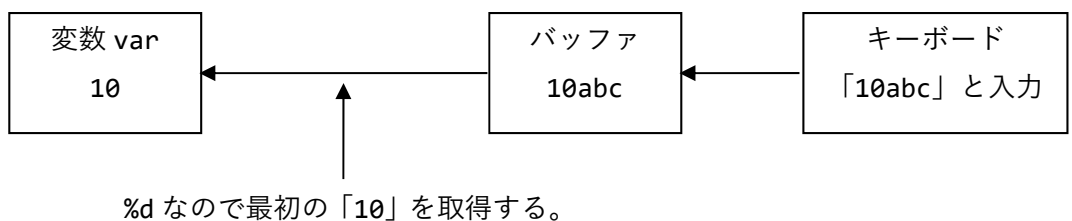
キーボードから「10」と入力した場合「10」と出力され、scanf 関数の戻り値は 1（正常）になります。



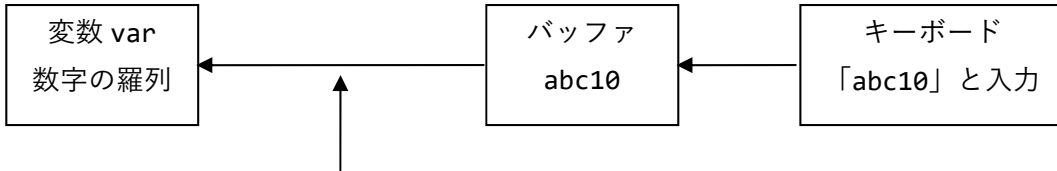
キーボードから「10.5」と入力した場合「10」と出力され、scanf 関数の戻り値は 1（正常）になります。



キーボードから「10abc」と入力した場合「10」と出力され、scanf 関数の戻り値は 1（正常）になります。



キーボードから「abc10」と入力した場合、不定値が出力され、scanf 関数の戻り値は 0（異常）になります。



最初の文字「a」を%d（10進整数）で取得するためおかしくなる。

【バッファフラッシュ】

下記のソースコードを実行してみましょう。

```

char var1;
char var2;

printf("変数 var1 を入力¥n");
scanf("%c", &var1);
printf("変数 var1 は%c です¥n", var1);

printf("変数 var2 を入力¥n");
scanf("%c", &var2);
printf("変数 var2 は%c です¥n", var2);
  
```

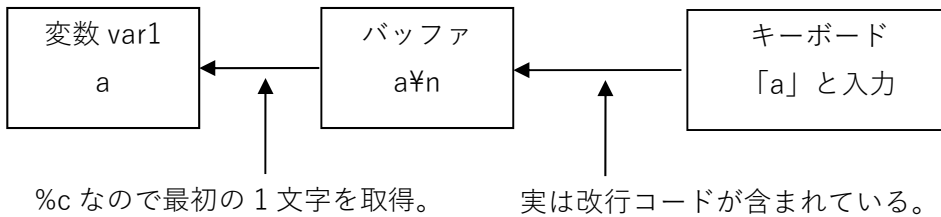
2 回目の変数 var2 に対する入力が勝手にスキップされませんでしたか？

これは「バッファフラッシュ」と称する現象で、「scanf 関数の仕様」と「改行コード」が原因で発生します。

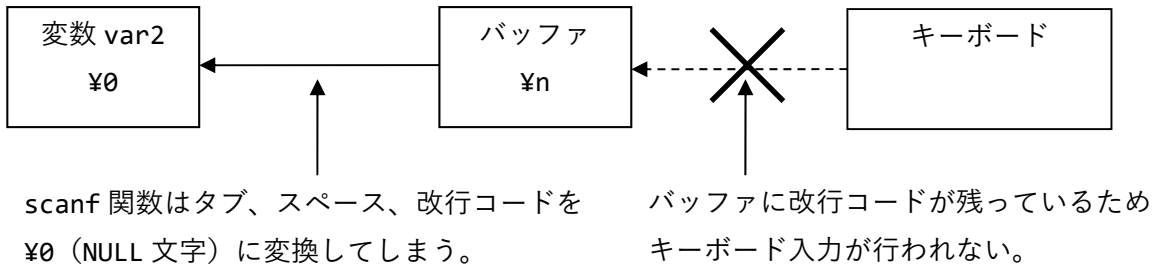
scanf 関数は「バッファを読み込む」関数で、バッファが空の場合はキーボードからの入力を促しますが、バッファ内にデータが残っている場合はキーボードからの入力が行われない仕様になっています。

C 言語 006 scanf 関数

1 回目の入力時



2 回目の入力時



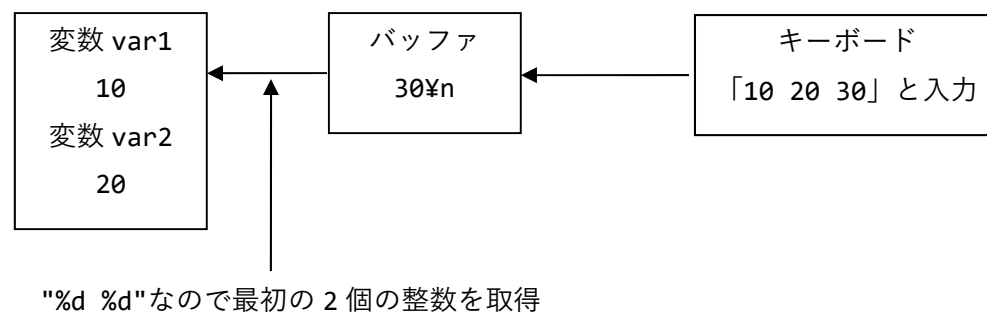
バッファフラッシュの対処法としては「常にバッファを空にしておくこと」です。

その方法は色々ありますが、書式指定子に%s (文字列) を使用してバッファ内の全データを文字列で取得するといいです。

<バッファに注意>

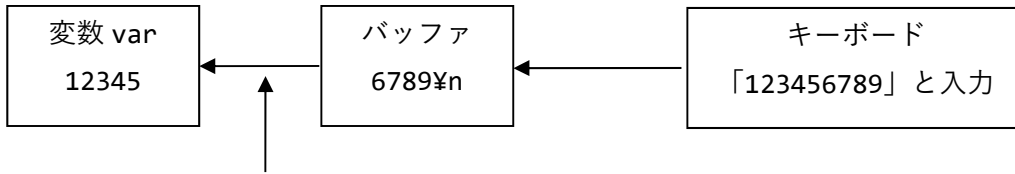
```
int var1
int var2;
scanf("%d %d", &var1, &var2);
```

キーボードから「10 20 30」と入力した場合、30 はバッファに残る。



```
int var;  
scanf("%5d", &var);
```

キーボードから「123456789」と入力した場合、6789 はバッファに残る。



%5d なので最初の 5 文字を取得。