

1.3 変数を使ったプログラム

リスト 1.7 キーボードから二つの整数を入力して和を求める

```
/* Sum of 2 variables */
#include <stdio.h>

int main()
{
    int integer1, integer2, sum;    /* 宣言 */

    printf("Enter first integer\n"); /* プロンプト */
    scanf("%d", &integer1);         /* 1つ目の整数を読み込む */
    printf("Enter second integer\n"); /* プロンプト */
    scanf("%d", &integer2);         /* 2つ目の整数を読み込む */
    sum = integer1 + integer2;      /* 和を代入する */
    printf("Sum is %d\n", sum);     /* 和をプリントする */

    return 0;    /* プログラムが正常終了したことを示す */
}
```

実行結果

```
Enter first integer
15
Enter second integer
3
Sum is 18
```

- 変数 (variable)
 - 値を入れておくための名前の付いた入れ物
 - * 実際はコンピュータのメモリ上の特定の場所
 - 変数に入れる値の種類 (型: type) は予め決めておく
 - * 当面は整数のための変数だけを扱う
 - 変数に値を入れる (代入する) と以前の値は無くなる
 - 変数から値を読んでも (参照しても) 変数の値は変わらない
- `int integer1, integer2, sum;`
 - 変数宣言
 - * 変数を使えるようにするための文
 - * 変数を使う前に必ず必要
 - `int` は変数が整数を保持できる (`int` 型変数である) ことを宣言するもの
 - 変数名 (識別子)

- * `integer1, integer2, sum`
 - * 識別子: 英字, 数字, 下線 (`_`) の並び (数字から始まることはできない)
 - ・ 大文字と小文字は区別される
 - ・ ANSI C の規格では少なくとも 31 文字まで有効
 - * 変数名は小文字で始めるのが慣習
 - * 用途がわかりやすい変数名にする
 - 宣言文は実行文より前に置く
 - * 宣言せずに変数を使ったり, ブロック内で実行文より後に変数宣言をすると, 構文エラー (syntax error) というコンパイル時エラーが起きる [ことがある]
- `scanf("%d", &integer1);`
 - `scanf` は標準入力 (通常はキーボード) からの入力を受け取り, 変数に入れるために使われる。
 - ここでの `scanf` 文は二つの引数を持っている
 - * `%d`: 入力が 10 進整数であるべき
 - * `&integer1`: 入力を格納する変数 (`integer1`) のメモリ内の場所
 - * 当面は, `scanf` 文の変数の前には `&` を付けなければならない, と考えることにする
 - `scanf` 文が実行されるとプログラムは入力待ちの状態になる。
ユーザーが整数をタイプしてリターン (エンター) キーを押すと, その整数が変数 (`integer1`) に入り, プログラムの実行が継続される
 - = 「代入演算子」
 - 変数に値を代入する
 - 二項演算子である (二つの被演算子 (operand) を持つ)


```
sum = integer1 + integer2;
```

 「sum に `integer1` と `integer2` の和を代入する」と読める
 - 左辺の変数に右辺の値が入る
 - * 右辺の `integer1` と `integer2` は, 各変数に格納された値 (変数値) を意味する
 - `printf("Sum is %d\n", sum);`
 - 二つの引数を持つ `printf` 文で変数の内容 (変数値) を出力 (表示)
 - * `%d`: 10 進整数として表示する指定
 - * `sum`: 表示する値 (ここでは変数値)
 - 前節のように `printf` 文の中で計算を行うことも可能


```
printf("Sum is %d\n", integer1 + integer2);
```
 - 変数は, 宣言しただけでは中身は不定 (何が入っているかわからない) なので, 必ず値を入れてから使う (参照する)⁵

⁵正確には変数の種類によるが, 当面はこう考えておくのがよい。

1.4 キーワード (予約語)

- C 言語において特別の意味を持つ語

auto, break, case, char, const, continue, default, do, double, else, enum, extern, float, for, goto, if, int, long, register, return, short, signed, sizeof, static, struct, switch, typedef, union, unsigned, void, volatile, while

- 変数名などの識別子として利用できない