

型変換とキャスト

これまで学習してきたように、C 言語では変数の宣言時に型を指定する。今、`int i, j, k; float p, q;` のように変数が宣言されていると、`i+j` と `p*q` の結果はそれぞれ整数、小数になる。では `i-q` の結果はどうなるだろう。これは C 言語では「より広い型になる」という原則があり、小数になる。
`p=i+j;` という代入では、右辺の整数の結果が `p` に代入される時に自動的に小数に変換される。`k=p*q;` では、右辺の小数の結果が `k` に代入される時に小数部を切捨てられて整数部のみが代入される。ただ、整数型は小数の型よりも表せる範囲が狭いので、ビルド時に警告が出るかもしれない。一般に、代入において左右の型が異なるのは好ましくない。

C 言語で用意されている標準関数に平方根を求める `sqrt()` がある（この標準関数を使う時は、`#include <math.h>` をプログラムの最初に書かなければならぬ）。この関数は引数として `double` 型を取るため、

```
p = sqrt(10);
```

は

```
p = sqrt(10.0);
```

と書いた方がいいだろう。

また、引数が整数型変数で

```
int i;
scanf("%d", &i);
p = sqrt(i); /* not recommended */
```

のように使いたい時は次のようにする。

```
int i;
scanf("%d", &i);
p = sqrt((double)i);
```

この `(double)` の記法で変数 `i` の値が一時的に `double` 型に変換される（変数 `i` 自身は変わらない）。この変換の記法をキャストといい、`()` 内に `int`、`float` などの型名を書く。さらに、`p` が单精度の時、関数 `sqrt` の返値は倍精度なので、この場合も大きい（精度がよい）値をそれよりも劣る型に代入しようとしているので、

```
p = (float)sqrt((double)i);
```

として、代入の左右の型を揃えた方がよい。

C 言語では、整数同士の割算の結果は整数になってしまふ（たとえその結果を `float` 型の変数に代入しようとしても、小数部が切り捨てられた値になる）。正しく小数の結果を得たい時は、

```
int i,j;
float p;
p = (float)i/j;
```

のようにして、小数を整数で割るようにする。

[練習 1] 年 y 、月 m 、日 d を与えたとき、次の手順 (Zeller の公式) によってその日の曜日を求めることができる。これを使って y 年 m 月 d 日の曜日を求めるプログラムを作りなさい。ただし、 $[x]$ は x を越えない最大の整数を意味し、また $i \bmod j$ は i を j で割った剰余を表す。

- 1) $m \leftarrow m - 2$
- 2) $m \leq 0$ なら $y \leftarrow y - 1$; $m \leftarrow m + 12$
- 3) $w \leftarrow (y + [y/4] - [y/100] + [y/400] + [2.6m - 0.2] + d) \bmod 7$
求まった w が 0 なら日曜、1 なら月曜、…、6 なら土曜である。

[練習 2] 上の式を使って y 年 m 月のカレンダーを作るプログラムを作りなさい。