

関数とポインタ

1. 関数

- 関数の定義：関数の入力と出力を宣言します。例えば `double func(double f, int n)` は、「double型1つと int型1つの引数（入力）を持ち、double型を戻値（出力）とする関数func」を表します。
- 関数の中身（処理内容）はその後に `{}` でくくって記述します。
- プロトタイプ宣言：プログラム中で、関数の実際の処理を記述する部分より前に関数を利用するために宣言（`double func(double f, int n);`）のみプログラムの最初の方に書きます。これによってコンパイラは関数の入力と出力を知ることができ、関数を呼び出している部分を適切に処理できるようになります。
- 関数の内部で宣言した変数は、その関数の外部には影響を与えません。複数の関数で同じ変数名を別の目的で使用できます。**関数を使用するには入力と出力だけを知っていれば良い** こととなります。このように関数同士の相互作用がないことは、大規模なプログラムを作成する際重要になります。（関数から呼び出し側に結果を戻すには、関数内部で定義した変数は使えません。入力として受け取る引数もコピーが渡されることとなります。）
- 関数から呼び出し側に結果を戻すには、a) 戻値、b) 引数のアドレス（ポインタ）渡し、c) 外部(global) 変数の使用、の3つの方法があります。
 - 1) 戻値：`return(変数)` で記述します。あらかじめ関数定義で出力として戻す値の型を宣言しておきます。1つの値しか返せないことに注意が必要です。
 - 2) アドレス渡し：結果を保存すべき変数のメモリ上のアドレスを引数として受け取ります。
 - 3) 外部 (global) 変数の使用：(mainを含む) 関数の外で定義した変数はglobal変数と呼ばれ、プログラムはどこからでもアクセスできます。しかし、global変数はプログラムのどこで書き換えられるかわからないので、多用すると関数同士の相互作用が大きくなり、プログラムのデバッグが困難になります。

2. ポインタ変数によるアドレス渡しの記述

戻値を使って結果を戻す例

```
double neg(double f);
int main(void)
{
    double a;
    a = 1.0;
    a = neg(a);
    printf("%f\n", a);
}

double neg(double f)
{
    return(-f);
}
```

アドレス渡しによって結果を戻す例

```
void neg(double* f);
int main(void)
{
    double a;
    a = 1.0;
    neg(&a);
    printf("%f\n", a);
}

void neg(double *f)
{
    *f = -(*f);
}
```

- ポインタとは「アドレスを使ってデータの格納場所を特定する仕組み」です。
- 引数の定義では、受け取ったアドレスを格納するポインタ変数に `*` をつけます。
- 関数を呼び出す側では、アドレス渡しする変数の実体を用意（宣言）します（`double a`）。これはポインタ変数ではないので `*` はつけません。
- 関数を呼び出す側では、宣言した変数は `*` をつけずに使います。
- 関数の呼び出し（`neg(&a)`）では、変数のアドレスを渡すため、変数に `&` をつけます。
- 呼び出される関数では、アドレスを受け取るポインタ変数が引数の定義で宣言されているので、ベットの宣言の必要はありません。
- 呼び出される関数では、アドレスはポインタ変数に格納されています。それが指し示す実体にアクセスするためにポインタ変数名に `*` をつけて使います（`*f = -(*f)`）。

- * は掛け算を表す演算子、ポインタ変数の宣言、ポインタ変数が指すデータにアクセスする間接演算子という3つの目的に使われるので注意が必要です。

課題

- (1) テキスト5ページの図4 fahrenheit.cを「華氏を引数として与えると摂氏を返す関数」として書き直してください。
- (2) キーボードから整数nを入力すると、その階乗を計算して結果を表示するプログラムを関数を使って記述してください。
- (3) 以下の多項式近似を利用して、xと近似の次数nを引数として与えるとsin(x)の近似計算をするプログラムを作ってください。

$$p(x) = p(0) + p(0)^{(1)}x + \frac{1}{2!}p(0)^{(2)}x^2 + \frac{1}{3!}p(0)^{(3)}x^3 + \dots$$

5次の多項式で近似計算した結果と標準ライブラリを利用して求めた結果を比較してください。Excelで両方のグラフを並べて表示してください。

- (4) アドレス渡しの関数を使って、2つの整数a, bを引数として与えるとaをbで割った商と余りを返すプログラムを作ってください。
- (5) quad2.cを改造して、以下のようなプログラムquad3.c にしてください。
関数quadの仕様を次のようにする：
引数：int型3つ (a, b, c)、float型ポインタ2つ (r1, r2)
戻り値：int型
 - a) 実数解が存在し、
重解でなければ、r1, r2が指すアドレスに解を代入し2を返す。
重解ならば、r1, r2が指すアドレスに解（同じ値）を代入し1を返す。
 - b) 実数解が存在しない時、複素数の解を求め、実部をr1が指すアドレスに、虚部の絶対値をr2が指すアドレス代入し、0を返す。
 - c) aがゼロの場合、何もせず -1を返す。
- (6) swap2.cを改造し、3変数の交換：xをyに、yをzに、zをxに交換するプログラムを作ってください。
- (7) 複素数の除算をするプログラムを作ってください。