

1 関数とは

1.1 学習のポイント

関数の役割と、ユーザー関数の作り方について学びます。

1.2 関数の役割

長いプログラムをよくみると、それにはいろいろな仕事の寄せ集めでできています。そのような個々に分割した単位を機能単位と呼びます。この機能単位をメインプログラムと別な部分に書き、必要に応じて呼び出すようにしたものを一般にサブルーチンといいます。Cではこのサブルーチンを関数というもので構成します。

一般にいうサブルーチンと関数の違いは、サブルーチンはそれ自体が値を持つことです。最も標準的な関数として算術関数があり、次のように使います。

```
y = 10.0 * sin(x);
```

これは、 $\sin(x)$ という関数がコールされると、サインの計算が行なわれ、その結果が $\sin(x)$ の値として返されているため、式の中に書いてその値を用いて計算することができるわけです。関数から返される値のことを戻り値と呼びます。

Cでは $\sin(x)$ のような算術関数以外にも、すべての処理単位が関数として扱われています。たとえば、`getchar()` は、キーボードから1文字入力を行なう関数で、入力された文字が関数の戻り値として返されます。

1.3 関数の定義と引数

`getchar`、`scanf`、`printf` などの関数はCのメーカーがあらかじめ作成し、ライブラリとして提供しているので、私たちはこれらの関数を定義することなく、使用することができます。このようにCのメーカーがライブラリとして提供する関数を標準ライブラリ関数といいます。

これに対し、私たちが独自に作る関数をユーザ関数といい、これは関数を定義してからでないと使えません。関数の呼び出しと、関数の定義は次のように書きます。

```
関数名 (実引数, 仮引数, . . .);    //関数の呼び出し
```

```
型 関数名 (型 実引数, 型 仮引数, . . .)  
{  
    .  
    .  
    .  
  
    return (式);  
}
```

関数は値を返すという性質を持っているので、その関数がどのような型のデータを返すものなのかを関数の型として明確に宣言しなければなりません。return(式) の式の値が関数の戻り値として呼び出し元に返されます。したがって式の値と関数の型は一致していなければなりません。

呼び出し元と関数との間でデータのやりとりを行なうには引数 (ひきすう) というものを用います。呼び出し側で指定する引数を実引数 (じつひきすう)、関数定義側で指定する引数を仮引数 (かりひきすう) と呼びます。仮引数は関数定義部においてその型を宣言しておく必要があります。実引数には、定数、変数、式 (定数や変数を演算子で結んだもの) が書けますが、仮引数ははデータを受け取る器なので変数しか書けません。対応する実引数と仮引数は同じ名前を付ける必要はありませんが型は必ず一致していなければなりません。

```
abc(10, 20);    //関数の呼び出し
```

```
int abc(int a, int b)  
{  
    .  
    .  
    .  
}
```

「C 言語」(河西朝雄著 ナツメ社)100 頁

1.4 例題 21

```
/*
  例題 21 C 言語 102 頁
  2つのデータ a,b のうちの大きい方を返す関数 max() を作りなさい。
  reidai21.c
*/

#include <stdio.h>

int max(int a, int b)
{
    int m;

    if(a > b){
        m = a;
    }else{
        m = b;
    }

    return m;
}

int main()
{
    int x, y;

    printf("data ? ");

    scanf("%d %d", &x, &y);

    printf("max = %d\n", max(x,y));

    return 0;
}
```

「C 言語」(河西朝雄著 ナツメ社)102 頁

2 ポートフォリオのリターンとリスク

2.1 2資産ポートフォリオのリターンとリスク

2つの資産 A、B のリターンがそれぞれ、 r_A 、 r_B と与えられるとき、この2資産へ投資を行なうポートフォリオのリターン r_P は、それぞれへの投資比率を X_A 、 X_B として、

$$r_P = X_A r_A + X_B r_B$$

となる。ここで投資比率に関して、各資産は無限に分割可能で、非負の値を取ると仮定している。また、 $X_A + X_B = 1$ である。

また、ポートフォリオのリスク (分散) σ_P^2 は、それぞれの標準偏差を σ_A 、 σ_B 、相関係数を ρ_{AB} として、

$$\sigma_P^2 = X_A^2 \sigma_A^2 + X_B^2 \sigma_B^2 + 2X_A X_B \rho_{AB} \sigma_A \sigma_B$$

となる。共分散と相関係数の関係を示した式を用いると上式は、

$$\sigma_P^2 = X_A^2 \sigma_A^2 + X_B^2 \sigma_B^2 + 2X_A X_B \sigma_{AB}$$

と表すことができる。ファイナンス入門 ('12) 50 頁

2.2 数値例 ファイナンス入門 52 頁

A、B 社の株価のリターンとリスクが下記で、相関係数が -0.58 の時のポートフォリオを計算します。

会社名	リターン (平均)	リスク (標準偏差)
A 社	3.2%	15.6%
B 社	1.2%	7.1%

```
/*
    portfol1.c
*/

#include <stdio.h>
#include <math.h>
```

```

int main()
{
    double rtn_a = 3.2;
    double rtn_b = 1.2;

    double std_a = 15.6;
    double std_b = 7.2;

    double cov    = -0.58;

    double rat;

    double p_rtn;
    double p_rsk;

    for(rat = 0.0; rat < 1.0; rat = rat + 0.01){
        p_rtn = rat * rtn_a + (1 - rat) * rtn_b;
        //printf("ポートフォリオのリターンは、 %lf\n", p_rtn);
        p_rsk = sqrt(
            pow(std_a, 2) * pow(rat, 2)
            + pow(std_b, 2) * pow((1 - rat), 2)
            + rat * (1 - rat) * 2 * cov * std_a * std_b
        );

        //printf("ポートフォリオのリスクは、 %lf\n", p_rsk);
        printf("%lf %lf\n", p_rsk, p_rtn);
    }

    return 0;
}

```

2.3 使い方

```
./portfol1
```

2.4 gnuplot で作図する

```
./portfolio1 > port1out.txt
```

として画面の表示をファイルに書き込む。
gnuplot を起動する。

3 gnuplot で作図する

```
./portfolio1 > port1out.txt
```

として画面の表示をファイルに書き込む。
gnuplot を起動する。

```
plot "port1out.txt"
```

で 2 資産のポートフォリオのリスクとリターンのグラフが描かれます。