

1 値による呼び出し (call by value)

1.1 学習のポイント

C の最も一般的な引数渡しである call by value の概念について学びます。

1.2 値による呼び出し (call by value)

call by value (値による呼び出し) は、C における最も一般的な引数渡しの方法で、実引数の実際の値 (value) を仮引数に渡します。

この方法では、仮引数の値を実引数に返すことはできませんから引数によるデータ授受は、呼び出し元 関数への一方通行となります。call by value では、引数を介して関数側から呼び出し元に値を返すことができないという不便さはありますが、呼び出された関数側で不注意に引数の値を変更したとしても、それは呼び出し元に影響を与えませんから、2 つのモジュールの独立性を極めて高くすることができます。

「C 言語」(河西朝雄著 ナツメ社)106 頁

1.3 例題 23

```
/*
  例題 23 C 言語 107 頁
  引数で与えられた数だけスペースを出力する関数 spc() を作りなさい。
  reidai23.c
*/

#include <stdio.h>

void spc(int n)
{
  for(;n > 0; n--){
    printf(" ");
  }
}
```

```
int main()
{
    int i;

    for(i = 0; i < 5; i++){
        spc(i);
        printf("Hello C\n");
    }

    return 0;
}
```

「C 言語」(河西朝雄著 ナツメ社)107 頁

2 AI による大規模データ処理入門

2.1 前処理プログラム

時間とともに変化する時系列データの変動を、上昇あるいは変化なし (1)、および、下降 (0) で判定するためのプログラムです。時系列データとして、たとえば株価データを用いれば、変動の傾向を記号的に把握することが可能です。

AI による大規模データ処理入門 17 頁 小高知宏著 オーム社

2.2 データ前処理プログラム

```
/*
    データ前処理プログラム
    prep.c
    入力された数値の増減を調べる
*/

#include <stdio.h>

#define LIMITL 256
#define TRUE 1
#define FALSE 0
```

```

void eval(double data, double olddata);

int main()
{
    char    linebuf[LIMITL];
    double data;
    double olddata;
    int     firstdata = TRUE;

    while(fgets(linebuf, LIMITL, stdin) != NULL){

        if(sscanf(linebuf, "%lf", &data) <= 0){
            break;
        }

        if(firstdata == TRUE){
            firstdata = FALSE;
        }else{
            eval(data, olddata);
        }

        olddata = data;
    }

    return 0;
}

void eval(double data, double olddata)
{
    if(data >= olddata){
        printf("1\n");
    }else{
        printf("0\n");
    }
}

```

2.3 日経平均株価 nikkei.txt

8802.51
9723.24
10083.56
9520.89
8542.73
9006.78
8695.06
8839.91
8870.16
8928.29
9446.01
10395.18
11138.66
11559.36
12397.91
13860.86
13774.54
13677.32
13668.32
13388.86
14455.80
14327.94
15661.87
16291.31
14914.53
14841.07
14827.83
14304.11
14632.38
15162.10
15620.77
15424.59
16173.52
16413.76
17459.85
17635.14

2.4 使い方

```
./prep < nillei.txt
```