

プログラミング演習 II

プログラミング課題 1 — ファイル入出力・復習 —

課題概要

この課題では、プログラミング言語 I とプログラミング演習 I で学習した内容の復習と、ファイル入出力を扱う問題に取り組む。課題の評価ポイントは、以下のプログラミング技法のとおりである。

プログラミング技法（復習）

- Level 1
 - プログラムにインデントを入れて、読みやすく書いている。
 - 意味のわかるような変数名を付けている。
- Level 2
 - 括弧の対応{ }がわかりやすく書いている。
 - 条件式が冗長ではない。（無駄な条件判定をしない）
 - 負の平方根や 0 の割り算など、数学的に誤りのある計算をしていない。
- Level 3
 - do 文、while 文、for 文の使い分けができる。
 - 適切な実行例を作ることができる。
- Level 4
 - 複合代入演算子を正しく扱える。
 - インクリメント++やデクリメント--を正しく扱える。
 - printf や scanf の様々な書式化文字列を扱える。

プログラミング技法

- Level 5
 - 適切な変数の型を使用することができる。
 - プログラムの適切な説明をコメント/* */で挿入することができる。
 - ファイルの open と close を適切に行うことができる。

要点

ファイル入出力のプログラム (部分)

```
#include <stdio.h>
FILE *fp; /*ファイルポインタ*/
if ( (fp=fopen("input", "r")) == NULL ) {
    printf("ファイルを開けません \n");
} else {
    while( !feof(fp) ) {
        fscanf(fp, "%d\n", &x); /*\n が重要*/
        ...
    }
    fclose(fp);
}
その他の関数： feof(), fgetc(), fputc(), fread(), fwrite() など
```

```
#include <stdio.h>
FILE *fp; /*ファイルポインタ*/
if ( (fp=fopen("output", "w")) == NULL ) {
    printf("ファイルを開けません \n");
} else {
    for(...) { /*ループ文*/
        ...
        fprintf(fp, "%d\n", x);
    }
    fclose(fp);
}
```

feof(fp) による判定ではなく、fscanf() の返却値で判定することもできる。

```
while( fscanf(fp, ... ) != EOF ) {
```

ファイル入出力のポイント

- ファイルに書き込む場合や、ファイルから読み込む場合には、そのファイルを open する (開く) 必要がある。また、開いたファイルは必ず close する (閉じる) 必要がある。
- ファイルへのアクセスは、ファイル名ではなくファイルポインタで行う。
- 何らかの原因で、ファイルを開けない場合は、ファイルポインタが NULL になっている。この場合は、エラーとして処理をする必要がある。また、close してはいけない。

課題 1：次のプログラムを作りなさい。

全ての問題で、適切な実行例を示すこと。

課題 1 では、まだ、**配列は使用しない**こと。

1. いくつかの商品の単価 y [円] と個数 n 個から金額を求めるプログラムを作りたい。ただし、単価と個数のデータは、ファイルに保存されており、ファイルから値を読み出して、金額を計算する。ファイルの中には複数の商品のデータが入っている。第 1 列を単価、第 2 列を個数とする。また、計算した結果 (単価、個数、金額) は別のファイルに保存する。また、すべての商品の合計金額を求め、合計金額と消費税を加えた代金の両方を出力しなさい。合計金額と代金は標準出力に出力すること。

単価と個数のファイル名を input.data とし、金額を計算したファイルの出力ファイル名を total.data とする。実行結果には、ファイルの内容を cat コマンドで表示した内容を入れること。input.data の内容は下記の 3 行の例を含め、その他の例を考えること。このファイルはエディタで作成してよい。

```
$ cat input.data
198    10
368     3
39    128
...
```

2. (This question will be provided from Adrien sen-sei.)

3. 整数 n を入力し、その数がある整数の二乗 x^2 になっていたら、「 x の平方数です」と表示するプログラムを作成しなさい。例えば、 $n = 4$ のときは、「4 は 2 の平方数です」と表示し、 $n = 3$ のときは、「3 は平方数ではありません」と表示する。

【ヒント】 \sqrt{n} までの整数を二乗して、 n に一致するかを調べればよい。

4. 数当てゲームを作ろう。正解となる整数は 4 桁とし、乱数を用いて決定する。正解はプレイヤーに秘密である。プレイヤーは整数を入力し、もし正解よりも大きな数を入力した場合は「正解はもっと小さな数です」と表示する。また、入力が正解よりも小さな数の場合は「正解はもっと大きな数です」と表示する。入力が正解に一致するまで、この操作を繰り返し、一致したら、「正解です」と表示して終了する。このとき、正解までに何回入力したかも表示しなさい。

正解は次のようにして作成する。実行毎に異なる正解を必要とするので、擬似乱数を用いる。擬似乱数は C 言語の関数 `drand48()` を用いれば良い。`drand48()` は 0 以上 1 未満の実数をランダムに返す関数である。ただし、この関数を使う前に、関数 `srand48()` でシード (乱数の種) を与える必要がある。

```
#include <stdio.h>
#include <stdlib.h>
#include <time.h>
int main( void ) {
    time_t tp;
    int ans;    /* 正解用の変数 */
    time(&tp); srand48(tp);          /* 乱数の初期化 */
    ans = (int)(drand48()*10000); /* 10000 倍して整数化すると 4 桁の整数になる */
    ...
}
```