

第 9 回課題 T09

(関数 I(処理の分解) : 教科書第 12 章, 2008/6/12(木))

基本問題

T09-1:公転周期計算関数

(本提出期限 2008/6/12(木)17:40, 再提出期限 2008/6/26(木)14:30)

提出物 : Makefile, ソースファイル:kepler.c, 入力ファイル:kepler.in, 出力ファイル:kepler.out

ケプラーの第 3 法則により, ある恒星系の惑星の軌道半径 (radius of orbit) からその惑星の公転周期 (period of revolution) を求める関数 (calc_period) を作成せよ. この関数 (calc_period) を利用して, その恒星系内のすべての惑星に対して, 公転周期を求めよ. ただし, 下の囲みおよび実行例を参考にして, 後ろの要求仕様を満たすこと. なお, 入力ファイルの一例を /home/student/submit/T09/1/kepler.in に用意したので, コピーして利用できる.

ケプラーの第 3 法則は, ある恒星系において惑星の軌道半径の 3 乗がその惑星の公転周期の 2 乗に比例するという法則である. すなわち, 軌道半径を r_o , 公転周期を p_r と置くと,

$$p_r^2 = kr_o^3$$

が成り立つ. ここで, k は恒星系によって決まる比例定数である. 各惑星の軌道半径を第 1 惑星の軌道半径の倍率で表わし, 各惑星の公転周期を第 1 惑星の公転周期の倍率で表せば, 比例定数 k は $k = 1$ となり,

$$p_r^2 = r_o^3$$

を満たす.

実行例

```
b09b0xx@tyy:~/T09/1$ ./kepler < kepler.in  
ケプラーの法則を基に，恒星系の公転周期を計算します．
```

惑星数を入力して下さい．

惑星毎に，軌道半径を入力して下さい．

第 1 惑星の軌道半径は？

第 2 惑星の軌道半径は？

第 3 惑星の軌道半径は？

第 4 惑星の軌道半径は？

第 5 惑星の軌道半径は？

第 6 惑星の軌道半径は？

第 7 惑星の軌道半径は？

第 8 惑星の軌道半径は？

公転周期を計算します．

第 i 惑星	軌道半径	公転周期
第 1 惑星	1.0000	1.0000
第 2 惑星	1.8685	2.5541
第 3 惑星	2.5833	4.1520
第 4 惑星	3.9362	7.8094
第 5 惑星	13.4399	49.2713
第 6 惑星	24.6833	122.6323
第 7 惑星	49.6471	349.8169
第 8 惑星	77.7846	686.0258

```
b09b0xx@tyy:~/T09/1$
```

要求仕様 (T09-1)

全体的な仕様:

- 軌道半径から公転周期を求める関数 `calc_period` を作成して利用する。関数 `calc_period` の仕様は後を参照。
- 各惑星の軌道半径、公転周期はそれぞれ `main` 関数内で 1 次元配列に蓄える。
- `for` 文を用いた繰り返しを `main` 関数内で (一度は) 用いる。
- 各惑星の軌道半径は第 1 惑星 (軌道半径の最小の惑星、すなわち恒星に最も近い惑星) の軌道半径に対する倍率で表わされるとする。軌道周期は 1.0 以上の正の実数である。
- 各惑星の公転周期は第 1 惑星の公転周期に対する倍率で表わされるとする。公転周期は 1.0 以上の正の実数である。
- 入力には以下を順に、標準入力から行なう。
 - 恒星系内の惑星数 n が入力される。ただし、恒星系内の惑星数の上限は 100 とする。すなわち、入力される n は 100 以下の自然数とする。
 - 惑星数の入力後、第 1 惑星から順に軌道半径が入力される。入力される軌道半径は 1.0 以上の正の実数とする。
- 入力値に対するチェックを行なう。正当な入力値であった場合には、`main` 関数の戻り値として 0 を返す。不正な入力値であった場合には、`main` 関数の戻り値として -1 を返す。
- 出力は以下の項目全てを整形して、標準出力へ行なう。
 - 各惑星に対して、軌道半径、公転周期を一行内に順に出力する。
 - 第 1 惑星から、軌道半径の小さい順にすべての惑星を出力する。
 - 軌道半径、公転周期は小数点以下 4 桁まで求める。
 - 出力値に対するチェックは行なわなくて良い。

公転周期を計算する関数 calc_period の仕様

- 次のプロトタイプ宣言を持つ .

```
/*
説明：
ケプラーの第 3 法則により，
軌道半径から公転周期を求める関数

引数：
radius 惑星の軌道半径 (1.0 以上の実数値)

戻り値：
引数 radius を軌道半径とする惑星の公転周期 (1.0 以上の
実数値)
*/
double calc_pereod(double radius);
```

- 引数に関して，以下を満たす .
 - 引数 radius はある恒星系における惑星の軌道半径が与えられる .
 - 軌道半径は，第 1 惑星の軌道半径に対する倍率で与えられる .
 - 軌道半径は，1.0 以上の実数値とする .
 - 関数 calc_period が呼び出された段階で，引数 radius には，1.0 以上の実数が蓄えられているとして良い . (関数 calc_period 内で引数の値をチェックするコードを書く必要は無い)
- 戻り値に関して，以下を満たす .
 - 引数 radius を軌道半径とする惑星の公転周期を戻り値とする .
 - 公転周期は，第 1 惑星の公転周期に対する倍率として示す .
 - 公転周期は，1.0 以上の実数値とする . (ただし，プログラムの中でチェックする必要は無い .)
- 関数 calc_period 中では，標準入出力 (printf 文による表示や，scanf 文による読み込み) は行なってはならない .

応用問題

T09-2:距離関数を用いた三角形面積計算

(本提出期限 2008/6/19(木)14:30, 再提出期限 2008/6/26(木)14:30)

提出物:Makefile, ソースファイル:triangle_area.c, 入力ファイル:triangle_area.in,
出力ファイル:triangle_area.out

3点 $A(x_a, y_a), B(x_b, y_b), C(x_c, y_c)$ から, その3点を頂点とする三角形の面積を求めるプログラムを作れ.

ただし, 下の実行例を参考に, 後ろに示す要求仕様を満足するように作成すること. なお, 座標データの一例を /home/student/submit/T09/2/triangle_area.in というファイルに用意したので, コピーして利用して良い.

実行例:

```
b09b0yy@ttx:~/T09/2/$ ./triangle_area < triangle_area.in
x_a?
y_a?
x_b?
y_b?
x_c?
y_c?
( 1.50, 4.80), (-3.40, 7.30), ( 4.60, -2.10)
を頂点とする三角形の面積は,
13.03
です.
b09b0yy@ttx:~/T09/2/$
```

要求仕様 (T09-2)

全体的な仕様:

- 2点の座標 $(x_1, y_1), (x_2, y_2)$ から, 2点間の距離を求める関数 `distance` を作成して利用する. 関数 `distance` の仕様は後ろを参照.
- 入力は, 各座標値を標準入力から $x_a, y_a, x_b, y_b, x_c, y_c$ の順に受け取る. 各座標値は任意の実数 (double 値) とする.
- 入力値のチェックは行なわなくて良い.
- 出力は, 三角形の面積を小数点2桁まで標準出力に行なう. 出力値のチェックは行なわなくて良い.

距離を求める関数 `distance` の仕様:

- 次のプロトタイプ宣言を持つ.

```
/*
説明:
2点間の距離を求める関数

引数:
x1:1点目のx座標(実数値)
y1:1点目のy座標(実数値)
x2:2点目のx座標(実数値)
y2:2点目のy座標(実数値)

戻り値:
点(x1, y1)と点(x2, y2)間の距離(0.0以上の実数値)
*/
double distance(double x1, double y1,
                double x2, double y2);
```

- 引数4つは全て実数値 (double 型の値) とする. 入力値のチェックは行なわなくて良い.
- 戻り値は引数で与えられた2点間の距離を返す. 戻り値は, 0.0以上の実数値 (double 型の値) である. 出力値のチェックは行なわなくて良い.
- 関数 `distance` 中では, 標準入出力 (`printf` 文による表示や, `scanf` 文による読み込み) は行なってはならない.