

第 10 回課題 T10

(関数 II(関数の利用と変数のスコープ) : 教科書第 12 章, 2008/6/19(木))

基本問題

T10-1:数学関数

(本提出期限 2008/6/19(木)17:40, 再提出期限 2008/7/3(木)14:30)

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

平方根を求めるプログラムを作成せよ。ただし、後ろの要求仕様を満たすこと。

実行例 1

```
b09b0xx@tyy:~/T10/1$ ./mymath
x の平方根を求めます。
x=? 2.0
x= 2.00 の平方根は, y=mysqrt(x)= 1.41421356 です。
```

実行例 2

```
b09b0xx@tyy:~/T10/1$ ./mymath
x の平方根を求めます。
x=? -3.0
x= -3.00 は負なので, 平方根はありません。
```

要求仕様 (T10-1)

全体的な仕様:

- 数学関数のライブラリを用いないこと。(ヘッダファイル `math.h` を `include` しないこと。)
- 与えられた実数値の絶対値を求める関数 `myfabs` を作成して利用する。関数 `myfabs` の仕様は後ろ参照。
- 与えられた実数値の平方根を求める関数 `mysqrt` を作成して利用する。関数 `mysqrt` の仕様は後ろ参照。
- `main` 関数以外では、標準入出力は用いないこと。
- プログラム全体の入力は、`main` 関数内で標準入力から読み込む。
- 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。
- プログラム全体の入力は、`main` 関数内で標準出力へ行なう。

絶対値を求める関数 myfabs の仕様

- 関数 myfabs は、数学ライブラリ中の関数 fabs 中と同様の動作をすること。
- 次のプロトタイプ宣言を持つ。

```
/*  
説明：  
与えられた絶対値を求める関数  
  
引数：  
x: 被演算項 (任意の実数)  
  
戻り値：  
引数 x の絶対値 |x| (非負の実数)  
*/  
double myfabs(double x);
```

- 引数は絶対値を求める演算の被演算項であり、任意の実数 (double 値) とする。(関数 myfabs 内でチェックする必要はない)
- 戻り値は引数の絶対値であり、非負の実数 (double 値) である。(関数 myfabs 内でチェックする必要はない)
- 関数 myfabs 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。

平方根を求める関数 mysqrt の仕様

- 関数 mysqrt は、数学ライブラリ中の関数 sqrt と同様の動作をすること。
- 次のプロトタイプ宣言を持つ。

```
/*  
説明：  
与えられた実数値の平方根を求める関数  
  
引数：  
x: 被演算項 (0 以上の実数)  
  
戻り値：  
引数 x の平方根 (0 以上の実数)  
*/  
double mysqrt(double x);
```

- 引数は平方根を求める演算の被演算項であり、0 以上の実数 (double 値) とする。(関数 mysqrt 内でチェックする必要はない。関数 mysqrt が呼び出された段階で、引数 x には 0 以上の実数値が保持されているとして良い。もし、実引数として 0 未満の値が渡された場合には、どのような動作をしてもかまわない。)
- 戻り値は引数の平方根であり、0 以上の実数 (double 値) である。(関数 myfabs 内でチェックする必要はない)
- 関数 myfabs 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。
- 次を満たすように、ニュートン法を利用すること。
 - 繰り返し回数は 1000 回を上限として、階差の絶対値が 10^{-5} より小さくなるまで行なう。ただし、これらの値 (繰り返し回数の上限、収束条件で用いる微小数) は、マクロ定義を利用して設定できるようにせよ。
 - 階差の絶対値を求める際には、関数 myfabs を利用せよ。(数学関数 fabs を用いてはならない。)
 - 繰り返し回数の上限 (1000 回) を越えた場合には、戻り値に大きな誤差が含まれていてもかまわない。

応用問題

T10-2:三角形面積計算関数

(本提出期限 2008/6/26(木)14:30 , 再提出期限 2008/7/3(木)14:30)

提出物:Makefile, ソースファイル:area_func.c, 入力ファイル:area_func.in,
出力ファイル:area_func.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@txx:~/T10/2/$ ./area_func < area_func.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@txx:~/T10/2/$
```

要求仕様 (T10-2)

全体的な仕様:

- 数学ライブラリを用いないこと。(ヘッダファイル `math.h` を `include` しないこと.)
- 絶対値を求める関数 `myfabs` を作成して利用する。関数 `myfabs` の仕様は基本問題を参照。
- 平方根を求める関数 `mysqrt` を作成して利用する。関数 `mysqrt` の仕様は基本問題を参照。
- 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。
- 三角形の面積を求める関数 `triangle_area` を作成して利用する。関数 `triangle_area` の仕様は後ろを参照。
- `main` 関数以外では、標準入出力は用いないこと。
- プログラム全体の入力は、`main` 関数内で標準入力から読み込む。
- 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);
```

- 平方根を求める際には関数 `mysqrt` を利用せよ . (数学関数 `sqrt` を用いてはならない .)
- 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 .

三角形の面積を求める関数 `triangle_area` の仕様:

- 次のプロトタイプ宣言を持つ .

```
/*
説明 :
三角形の面積を求める関数

引数:
x_a:点 A の x 座標 (実数値)
y_a:点 A の y 座標 (実数値)
x_b:点 B の x 座標 (実数値)
y_b:点 B の y 座標 (実数値)
x_c:点 C の x 座標 (実数値)
y_c:点 C の y 座標 (実数値)

戻り値:
点 A(x_a,y_a), 点 B(x_b,y_b), 点 C(x_c,y_c) を頂点と
する
三角形の面積
*/
double triangle_area(double x_a,double y_a,
                     double x_b,double y_b,
                     double x_c,double y_c);
```

- 平方根を求める際には関数 `mysqrt` を利用せよ . (数学関数 `sqrt` を用いてはならない .)
- 関数 `distance` を利用せよ .
- 引数 6 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 .
- 戻り値として , 引数で与えられた 3 点を頂点とする三角形の面積を返す . 戻り値は , 0.0 以上の実数値 (`double` 型の値) である . 戻り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 .