

課題

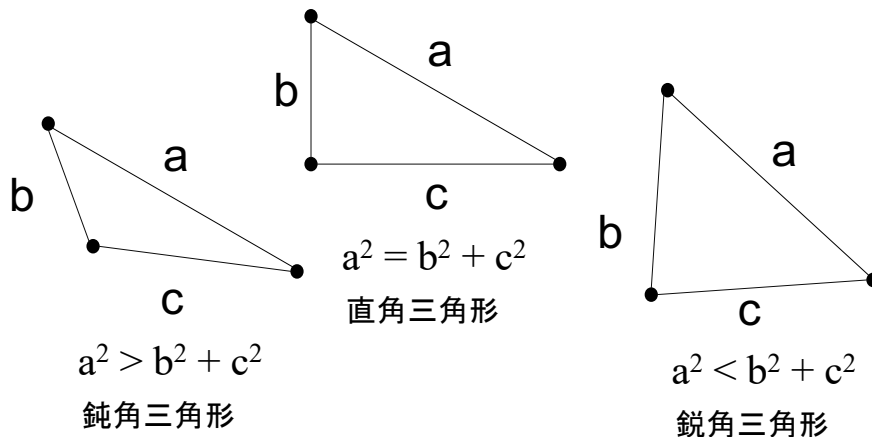
締め切り 7月6日 23:59

課題1

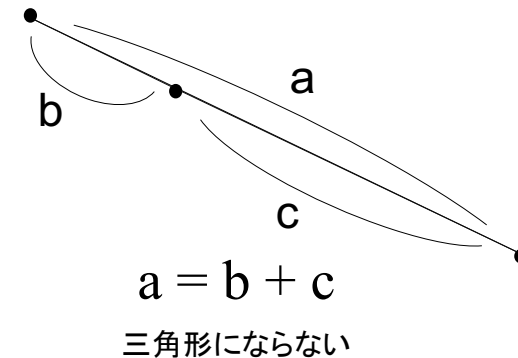
- 問題: 次の要件を満たすプログラムを作成し, ソースコード(kadai1.c)を提出してください.
 - double型の引数x1, y1, x2, y2をとり, 2点(x1, y1)と(x2, y2)間の距離を求めてdouble型で戻す関数distが定義されている
 - dist関数の定義内では標準関数sqrtを使ってよい
 - main関数内では次の処理をする.
 - (1) 3点の座標を標準入力から得る
 - (2) その3点を作る三角形が, 鋭角三角形, 鈍角三角形, 直角三角形のどれか, あるいは三角形を形作らないかを判定する
 - (3) 判定結果を標準出力に出力する

課題1のヒント

- 三角形の種類の判定(最も長い辺の長さがaのとき)



課題1のヒント



課題2

- 問題: 次の要件を満たすプログラムを作成し, ソースコード(kadai2.c)を提出してください.

- 2つの文字列の辞書順を調べる関数

`int strcmp(char *str1, char *str2);`
を定義する. `strcmp`の戻り値は, 辞書順で`str1`が`str2`より先の場合は `-1`, `str1`と`str2`が等しければ`0`, `str1`が`str2`より後の場合は`1`とする.

- 文字列配列を受け取って辞書順(昇順)にソートする関数
`void dictsort (char **, int);`
を定義する. 第1引数が文字列配列の先頭ポインタで, 第2引数が配列の要素数である. 定義の中で上記で作成した `strcmp` 関数を使うこと.

課題(つづき)

- `main`関数では,

```
char *flowers[] = {"cherry", "ume", "tulip", "dandelion",  
"peach", "marigold", "azalea", "orchid", "lily",  
"carnation", "pansy", "hyacinth", "iris"};
```

と初期化された文字列配列 `flowers` の要素を, 上記で作成した `dictsort` を使って辞書順にソートし, その結果を標準出力に表示する.

■ 注意

- 文字列操作の標準関数を使わないこと
(`strcmp`やソートをする関数は標準関数として使えるものがあるが, 今回は使わないこと)

補足

■ 辞書順とは

- (英単語が)辞書に出てくる順番
 - `apple` と `application` では `apple` の方が先
 - `locational` と `location` では `location` の方が先