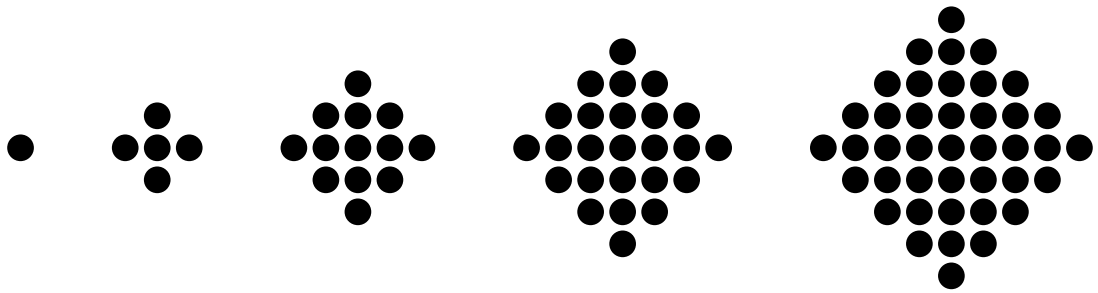
問題は 4 問ある。それとは別におまけ問題が 2 問ある。

問題 1. 次のように並べた玉の数を**中心つき四角数**と呼ぶ。



n 番目の中心つき四角数 C_n は

$$C_n = n^2 + (n-1)^2 \tag{1}$$

と表せることが知られている。

プログラム 1-1 は n 番目の中心つき四角数の計算を C の関数として実装したものである。プログラム 1-2 は最初の n 個の中心つき四角数の計算を C の関数として実装したものである。

(a) 空欄を埋めよ。

プログラム 1-1: n 番目の中心つき四角数を計算する関数

```
int
centered_square_number(int n)
{
        int     n1 = n - 1;
        return （あ）;
}
```

プログラム 1-2: 最初の n 個の中心つき四角数を計算する関数

```
/*      array[n − 1] に Cn が格納される      */
int *
centered_square_number_sequence(int *array, size_t size)
{
        int     n = （い）;
        int     n2_prev = （う）;
        int     *p;
        for (p = （え）; p < （お）; p ++) {
                int     n2 = n * n;
                （か） = （き）;
                n2_prev = n2;
                n ++;
        }
        return array;
}
```

問題 2. 三角形 ABC の頂点と重心 G と内心 I とナーゲル点 Na の座標を $A = (x_A, y_A)$, $B = (x_B, y_B)$, $C = (x_C, y_C)$, $G = (x_G, y_G)$, $I = (x_I, y_I)$, $\mathrm{Na} = (x_{\mathrm{Na}}, y_{\mathrm{Na}})$ とおくと、以下の関係が成り立つことが知られている。

$$(x_G, y_G) = \left(\frac{x_A + x_B + x_C}{3}, \frac{y_A + y_B + y_C}{3} \right) \tag{2a}$$

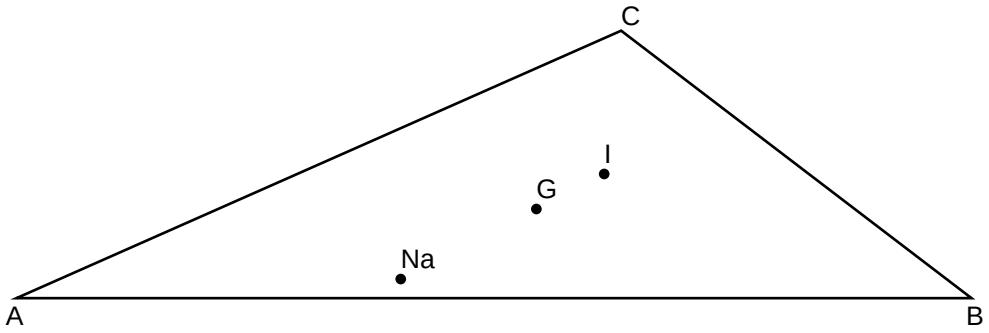
$$(x_I, y_I) = \left(\frac{ax_A + bx_B + cx_C}{a+b+c}, \frac{ay_A + by_B + cy_C}{a+b+c} \right) \tag{2b}$$

$$(x_{\mathrm{Na}}, y_{\mathrm{Na}}) = (3x_G - 2x_I, 3y_G - 2y_I) \tag{2c}$$

ただし、a, b, c はそれぞれ辺 BC, CA, AB の長さであり、

$$a = \sqrt{(x_B - x_C)^2 + (y_B - y_C)^2}, \quad b = \sqrt{(x_C - x_A)^2 + (y_C - y_A)^2}, \quad c = \sqrt{(x_A - x_B)^2 + (y_A - y_B)^2}$$

と書ける。



プログラム 2-1 と 2-2 と 2-3 は、(2a)(2b)(2c) を C の関数として素直に実装したものである。

- いずれのプログラムでも、平面上の点を表すのに `struct point` を、平面上の三角形を表すのに `struct triange` を使う。`struct point` の定義は三つのプログラムで共通だが、`struct triangle` の定義はプログラム 2-3 のみ他の二つと異なる。それぞれの構造体のメンバの意味はコメントに書かれた通りである。
- 三つのプログラムは、関数のインターフェースが異なる。それぞれの関数の動作は、コメントとして書かれた通りである。

(a) プログラム 2-1 の空欄を埋めよ。2 回出現している（も）には同じものを埋めよ。

(b) プログラム 2-2 の空欄を埋めよ。（も）と（や）と（ゆ）は、それぞれ、2 回ずつ出現しているが、同じ記号の空欄には同じものを埋めよ。

(c) プログラム 2-3 の空欄を埋めよ。（も）と（や）と（ゆ）は、それぞれ、2 回ずつ出現しているが、同じ記号の空欄には同じものを埋めよ。

プログラム 2-1：ナーゲル点の座標を計算するプログラム（1）

```
#include <stdio.h>
#include <math.h>

struct point {
        double  x;      /* x 座標 */
        double  y;      /* y 座標 */
```

```
};

struct triangle {
        struct point    A;      /* 頂点 A */
        struct point    B;      /* 頂点 B */
        struct point    C;      /* 頂点 C */
};

/*      第一引数：元の三角形      返値：重心      */
struct point
centroid(struct triangle ref)
{
        struct point    result;
        （あ） = （い） + （う） + （え） / 3;
        （お） = （か） + （き） + （く） / 3;
        return result;
}

/*      第一引数：元の三角形      返値：内心      */
struct point
incenter(struct triangle ref)
{
        double  a = hypot（け） - （こ）, （さ） - （し）);
        double  b = hypot（す） - （せ）, （そ） - （た）);
        double  c = hypot（ち） - （つ）, （て） - （と）);
        struct point    result;
        （な） = (a * （に） + b * （ぬ） + c * （ね）) / (a + b + c);
        （の） = (a * （は） + b * （ひ） + c * （ふ）) / (a + b + c);
        return result;
}

/*      第一引数：線分の端点 1    第二引数：線分の端点 2    返値：線分を 2:3 に外分する点   */
struct point
externally_dividing_point_2_3(struct point endpoint1, struct point endpoint2)
{
        struct point    result;
        （へ） = （ほ） * 3 - （ま） * 2;
        （み） = （む） * 3 - （め） * 2;
        return result;
}

/*      第一引数：元の三角形      返値：ナーゲル点   */
struct point
nagel_point(struct triangle ref)
{
        return externally_dividing_point_2_3(centroid（も）, incenter（も）);
}

int
main()
{
        struct triangle t;
```

```
        struct point    Na;

        scanf("%lf%lf%lf%lf%lf%lf", &t.A.x, &t.A.y, &t.B.x, &t.B.y, &t.C.x, &t.C.y);
        Na = nagel_point(t);
        printf("%15.10f %15.10f\n", Na.x, Na.y);
        return 0;
}
```

プログラム 2-2: ナーゲル点の座標を計算するプログラム (2)

```
#include <stdio.h>
#include <math.h>

struct point {
        double  x;      /* x 座標 */
        double  y;      /* y 座標 */
};

struct triangle {
        struct point    A;      /* 頂点 A */
        struct point    B;      /* 頂点 B */
        struct point    C;      /* 頂点 C */
};

/*      第一引数：元の三角形を指すポインタ      */
/*      第二引数：重心を格納するところを指すポインタ      */
void
centroid(const struct triangle *ref, struct point *result)
{
        （あ） = ( （い） + （う） + （え） ) / 3;
        （お） = ( （か） + （き） + （く） ) / 3;
}

/*      第一引数：元の三角形を指すポインタ      */
/*      第二引数：内心を格納するところを指すポインタ      */
void
incenter(const struct triangle *ref, struct point *result)
{
        double  a = hypot( （け） - （こ）, （さ） - （し） );
        double  b = hypot( （す） - （せ）, （そ） - （た） );
        double  c = hypot( （ち） - （つ）, （て） - （と） );
        （な） = (a * （に） + b * （ぬ） + c * （ね） ) / (a + b + c);
        （の） = (a * （は） + b * （ひ） + c * （ふ） ) / (a + b + c);
}

/*      第一引数：線分の端点 1 を指すポインタ      */
/*      第二引数：線分の端点 2 を指すポインタ      */
/*      第三引数：線分を 2:3 に外分する点を格納するところを指すポインタ   */
void
externally_dividing_point_2_3(
        const struct point *endpoint1, const struct point *endpoint2,
        struct point *result)
{
```

```
        （へ） = （ほ） * 3 - （ま） * 2;
        （み） = （む） * 3 - （め） * 2;
}

/*      第一引数：元の三角形を指すポインタ     */
/*      第二引数：ナーゲル点を格納するところを指すポインタ     */
void
nagel_point(const struct triangle *ref, struct point *result)
{
        struct point    G;
        struct point    I;
        centroid(（も）, （や）);
        incenter(（も）, （ゆ）);
        externally_dividing_point_2_3(（や）, （ゆ）, （よ）);
}

int
main()
{
        struct triangle t;
        struct point    Na;

        scanf("%lf%lf%lf%lf%lf%lf", &t.A.x, &t.A.y, &t.B.x, &t.B.y, &t.C.x, &t.C.y);
        nagel_point(&t, &Na);
        printf("%15.10f %15.10f\n", Na.x, Na.y);
        return 0;
}
```

プログラム 2-3：ナーゲル点の座標を計算するプログラム（3）

```
#include <stdio.h>
#include <math.h>

struct point {
        double  x;      /* x 座標 */
        double  y;      /* y 座標 */
};

struct triangle {
        struct point    *A;     /* 頂点 A を指すポインタ */
        struct point    *B;     /* 頂点 B を指すポインタ */
        struct point    *C;     /* 頂点 C を指すポインタ */
};

/*      第一引数：元の三角形を指すポインタ     */
/*      第二引数：重心を格納するところを指すポインタ     */
void
centroid(const struct triangle *ref, struct point *result)
{
        （あ） = (（い） + （う） + （え）) / 3;
        （お） = (（か） + （き） + （く）) / 3;
}

/*      第一引数：元の三角形を指すポインタ     */
```

```
/*      第二引数：内心を格納するところを指すポインタ     */
void
incenter(const struct triangle *ref, struct point *result)
{
        double  a = hypot( (け) - (こ), (さ) - (し) );
        double  b = hypot( (す) - (せ), (そ) - (た) );
        double  c = hypot( (ち) - (つ), (て) - (と) );
        (な) = (a * (に) + b * (ぬ) + c * (ね) ) / (a + b + c);
        (の) = (a * (は) + b * (ひ) + c * (ふ) ) / (a + b + c);
}

/*      第一引数：線分の端点 1 を指すポインタ     */
/*      第二引数：線分の端点 2 を指すポインタ     */
/*      第三引数：線分を 2:3 に外分する点を格納するところを指すポインタ   */
void
externally_dividing_point_2_3(
        const struct point *endpoint1, const struct point *endpoint2,
        struct point *result)
{
        (へ) = (ほ) * 3 - (ま) * 2;
        (み) = (む) * 3 - (め) * 2;
}

/*      第一引数：元の三角形を指すポインタ     */
/*      第二引数：ナーゲル点を格納するところを指すポインタ     */
void
nagel_point(const struct triangle *ref, struct point *result)
{
        struct point    G;
        struct point    I;
        centroid( (も), (や) );
        incenter( (も), (ゆ) );
        externally_dividing_point_2_3( (や), (ゆ), (よ) );
}

int
main()
{
        struct triangle t;
        struct point    v[3];
        struct point    Na;

        scanf("%lf%lf%lf%lf%lf%lf",
              &v[0].x, &v[0].y, &v[1].x, &v[1].y, &v[2].x, &v[2].y);
        t.A = &v[0]; t.B = &v[1]; t.C = &v[2];
        nagel_point(&t, &Na);
        printf("%15.10f %15.10f\n", Na.x, Na.y);
        return 0;
}
```

問題 3. プログラム 3-1 は、文字列の中のアルファベットがいくつあるかを数える C の関数を作ったつもりだが、欠陥のあるものである。どのような欠陥であるかを指摘し、それによってどのような不具合が生じる可能性があるかを説明し、修正案を提案せよ。

プログラム 3-1：文字列の中のアルファベットがいくつあるかを数える C の関数を実装したつもりで欠陥あり

```
#include <ctype.h>

int
count_alpha(const char *str)
{
        int     counter = 0;
        const char      *p;
        for (p = str; *p != '\0'; p ++) {
                if (isalpha(*p)) {
                        counter ++;
                }
        }
        return counter;
}
```

問題 4. n 番目の正八面体数は $n(2n^2+1)/3$ であるが、それを計算する C の関数を池原雄介はプログラム 4-1 のように書いた。それを見た鈴木勇は、プログラム 4-2 のように修正した。

池原雄介版よりも鈴木勇版が優れている点を説明せよ。

プログラム 4-1：（池原雄介 版）

```
unsigned
octahedral_number(unsigned n)
{
        return n * (2 * n * n + 1) / 3;
}
```

プログラム 4-2：（鈴木勇 版）

```
unsigned
octahedral_number(unsigned n)
{
        unsigned        p, q;
        if (n % 3 == 0) {
                p = n / 3; q = 2 * n * n + 1;
        } else {
                p = n; q = (2 * n * n + 1 )/ 3;
        }
        return p * q;
}
```

おまけ問題 1 (自信のない人のみ)**.** 三角形 ABC の辺 BC の中点を M_A、辺 CA の中点を M_B、辺 AB の中点を M_C とおくと、三直線 AM_A と BM_B と CM_C は一点で交わる。その交点が重心 G である。

三角形 ABC の内接円の中心が内心 G である。

三角形 ABC の $\angle A$ 内の傍接円が辺 BC と接する点を T_A と、$\angle B$ 内の傍接円が辺 CA と接する点を T_B と、$\angle C$ 内の傍接円が辺 AB と接する点を T_C とおくと、三直線 AT_A と BT_B と CT_C は一点で交わる。その交点がナーゲル点 Na である。

問題 2 の (2a) と (2b) と (2c) を証明せよ。

おまけ問題 2 (自信のない人のみ)**.** 自分でプログラミング言語を作るとしたらどんな名前をつけたいか。つけたい名前を記し、その名前の由来も説明せよ。