

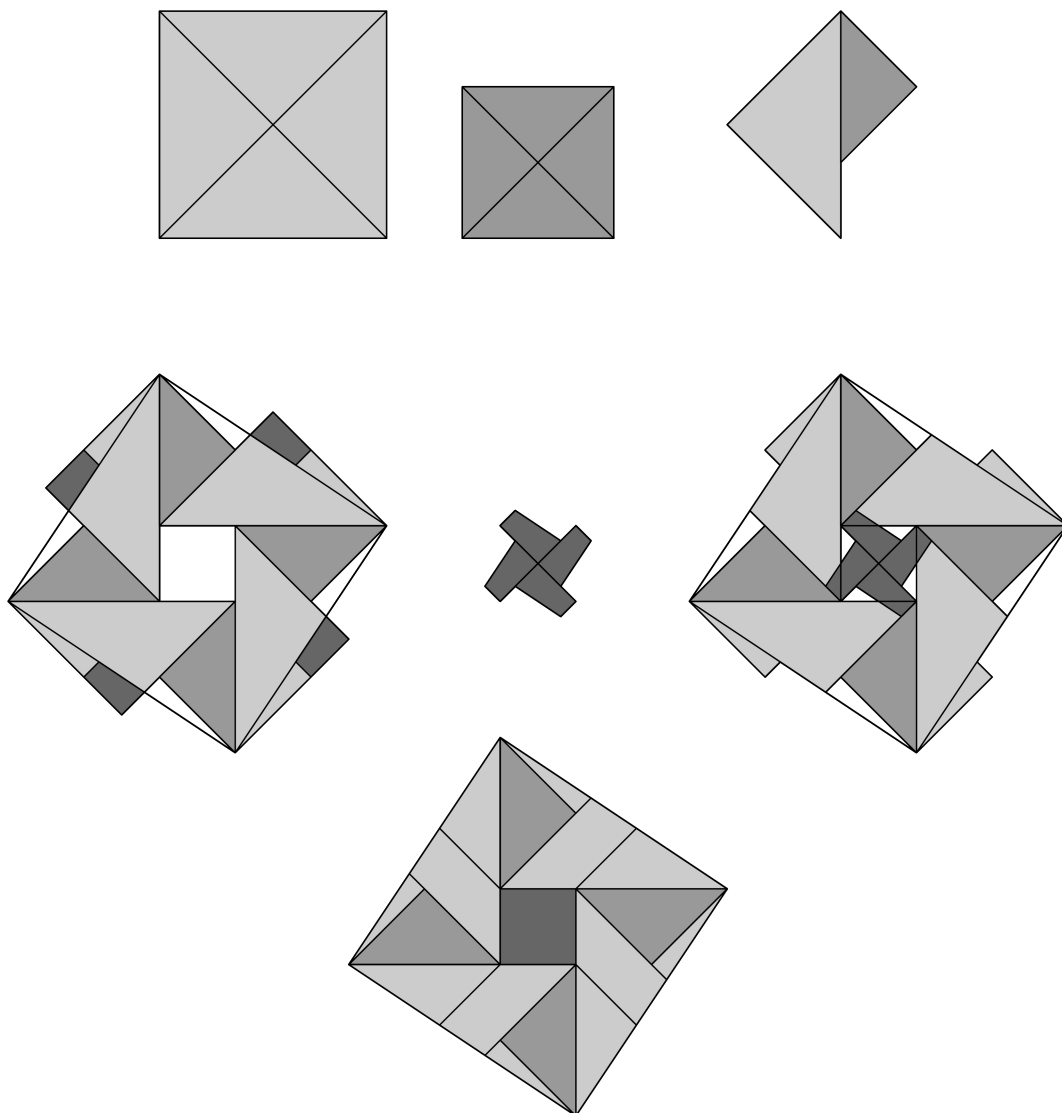
Một số bài tập lập trình với ASY

Tips 03 2023 ...

Bài viết sẽ mô tả việc lập trình để vẽ hình với ASY.

1 Bài tập ASY 1

Vẽ hình dưới đây.



Trước khi vẽ hình cần phân tích và ghi chú lại những điều cần thiết cho việc lập trình.

Ở hình trên, nhận thấy:

- Hình gồm nhiều hình nhỏ. Có thể viết hàm để vẽ từng hình.
- Mỗi hình cần áp vào hệ tọa độ và điểm gốc tọa độ $(0, 0)$. Có thể dùng kiểu dữ liệu **pic** để vẽ từng hình.
- Hình **a**, **b** giống nhau, chỉ khác nhau màu **fill** nên dùng chung 1 hàm, lưu ý có biến kiểu **pen** để đổi màu tô khi cần.

- Hình d, e, f, g có thể dùng chung được nhiều biến. Có thể viết thành 1 hàm.

```
unitsize(1cm);
void draw_pic_ab(pair O=(0,0), real a, pen pcolor) {
    path square=O--(O.x+a, O.y)--(O.x+a, O.y+a)--(O.x, O.y+a)--cycle;
    filldraw(square, pcolor);
    draw(square^^O--(O.x+a, O.y+a)--(O.x+a, O.y)--(O.x, O.y+a));
}

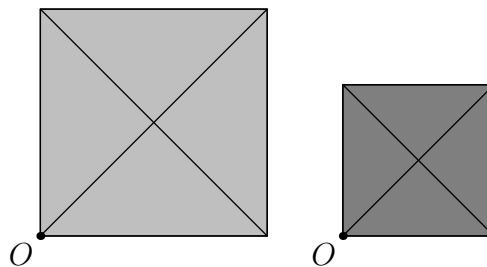
////////////////////////////////////

real a=3, b=2, c=sqrt(a^2-b^2);

draw_pic_ab((0,0), a=a, pcolor=1.5*grey);
draw_pic_ab((a+1,0), a=b, pcolor=grey);

shipout(bbox(1.5mm, invisible));
```

Đoạn code trên, hàm **draw_pic_ab** được sử dụng để vẽ 02 hình ban đầu. Biến *O* được dùng để xác định đỉnh dưới cùng bên trái của hình vuông. Lưu ý nên vẽ các **path** sau khi tô màu. Kết quả như sau:



Để vẽ hình c, ta viết hàm **draw_pic_c**:

```
unitsize(1cm);

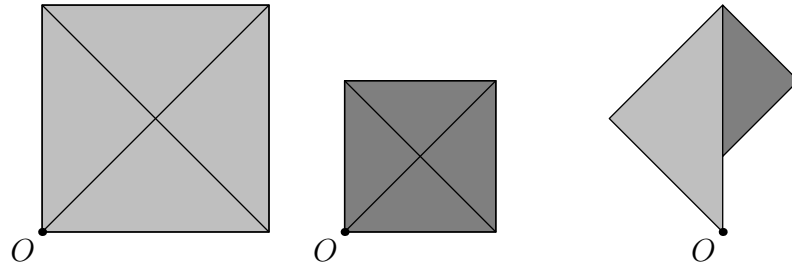
void draw_pic_ab(pair O=(0,0), real a, pen pcolor) {
    path square=O--(O.x+a, O.y)--(O.x+a, O.y+a)--(O.x, O.y+a)--cycle;
    filldraw(square, pcolor);
    draw(square^^O--(O.x+a, O.y+a)--(O.x+a, O.y)--(O.x, O.y+a));
}

void draw_pic_c(pair O=(0,0), real a, real b) {
    path half_sq_a=O--(O.x, O.y+a)--(O.x-a/2, O.y+a/2)--cycle;
    path half_sq_b=(O.x, O.y+a)--(O.x, O.y+a-b)--(O.x+b/2, O.y+a-b/2)--cycle;

    filldraw(half_sq_a, 1.5*grey);
    filldraw(half_sq_b, grey);
}

////////////////////////////////////

real a=3, b=2, c=sqrt(a^2-b^2);
draw_pic_ab((0,0), a=a, pcolor=1.5*grey);
draw_pic_ab((a+1,0), a=b, pcolor=grey);
draw_pic_c((3a,0), a=a, b=b);
```



Ta tiếp tục vẽ hình d bằng hàm `draw_pci_d`. Ở hàm này ta xác định các điểm A, B và tâm quay là điểm C .

```
unitsize(1cm);

void draw_pic_ab(pair O=(0,0), real a, pen pcolor) {
    path square=O--(O.x+a, O.y)--(O.x+a, O.y+a)--(O.x, O.y+a)--cycle;
    filldraw(square, pcolor);
    draw(square^^O--(O.x+a, O.y+a)--(O.x+a, O.y)--(O.x, O.y+a));
}

void draw_pic_c(pair O=(0,0), real a, real b) {
    path half_sq_a=O--(O.x, O.y+a)--(O.x-a/2, O.y+a/2)--cycle;
    path half_sq_b=(O.x, O.y+a)--(O.x, O.y+a-b)--(O.x+b/2, O.y+a-b/2)--cycle;

    filldraw(half_sq_a, 1.5*grey);
    filldraw(half_sq_b, grey);
}

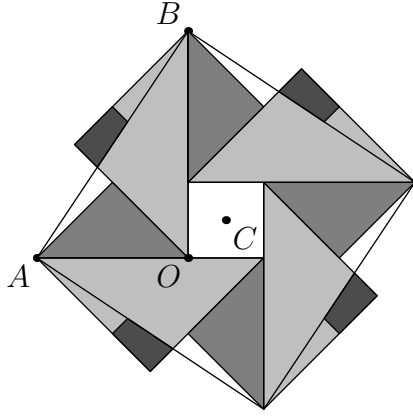
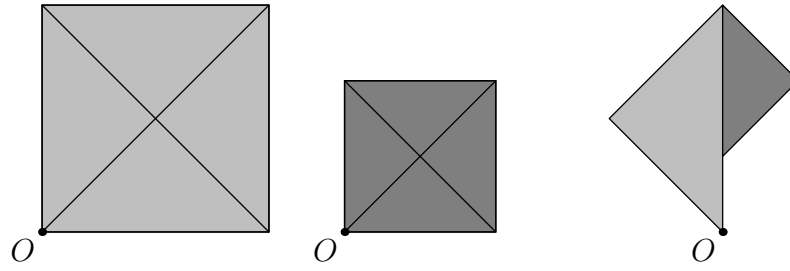
void draw_pic_d(pair O=(0,0), real a, real b) {
    real d=a-b;
    pair A=(O.x-b, O.y), B=(O.x, O.y+a), C=(O.x+d/2, O.y+d/2);
    pair M=(O.x-a/2, O.y+a/2);
    pair N=(O.x-b/2, O.y+a-b/2);
    pair P=extension(A, B, (O.x, O.y+a-b), N);
    pair Q=extension(A, B, O, M);

    path p0=A--O--B--cycle;
    path p1=O--B--M--cycle;
    path p2=O--A--(O.x-b/2, O.y+b/2)--cycle;
    path p3=M--N--P--Q--cycle;

    for (int i=0; i<4; ++i) {
        filldraw(rotate(i*90,C)*p1, 1.5grey);
        filldraw(rotate(i*90,C)*p2, grey);
        filldraw(rotate(i*90,C)*p3, 0.6grey);
        draw(rotate(i*90,C)*p0);
    }
}

////////////////////

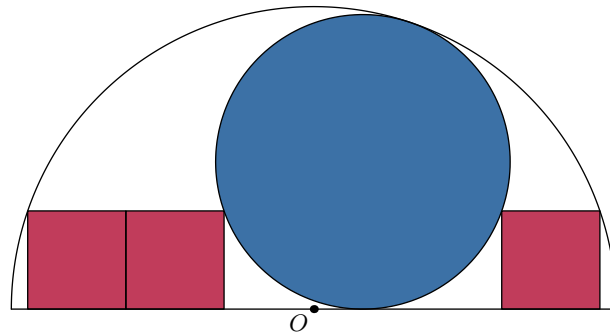
real a=3, b=2, c=sqrt(a^2-b^2);
draw_pic_ab((0,0), a=a, pcolor=1.5*grey);
draw_pic_ab((a+1,0), a=b, pcolor=grey);
draw_pic_c((3a,0), a=a, b=b);
draw_pic_d((0,-1.6a), a=a, b=b);
```



Các hình còn lại để người đọc viết tiếp!

2 Solution

P1. Vẽ hình dưới.



Hình trên là một ví dụ nữa cho việc tính toán khi vẽ hình. Để vẽ đường tròn chính xác, ta dùng gói **geometry**.

Để thuận tiện cho việc tính toán, chọn đường tròn ngoài là đường tròn tâm $O = (0, 0)$ và bán kính **1**. Gọi r , x là bán kính đường tròn nhỏ và cạnh hình vuông, tính toán ta sẽ có $r = \frac{1}{2} - \frac{x^2}{8}$ và $\frac{3x}{2} - \sqrt{1 - x^2} + x^2 - 2 * r * x = 0$. Đến đây nhiều người sẽ đi tính để chỉ còn hàm với biến x nhưng việc này là không cần thiết. Tham khảo hàm **f(x)** ở đoạn code dưới.

Việc tìm nghiệm của hàm trên, ta viết thêm hàm **SecantMethod** để tìm x gần đúng. Lưu ý đoạn nghiệm $x1, x2$.

```
import geometry;
unitsize(4cm);
defaultpen(fontsize(9pt));
pen mycolor1=RGB(187, 38, 73);
pen mycolor2=RGB(38, 97, 156);

point O=(0,0); dot("$O$", O, SW);
```

```

circle co=circle(0, 1); //draw(co);

real f(real x) {
    real r = 1/2-x^2/8;
    return (3x/2 - sqrt(1-x^2))^2 + x^2 -2r*x;
}

real SecantMethod(real f(real x), real x1=0.3, real x2=0.39, real e=EPS) {
    real x0, c, xm;
    int n=0;
    do {
        x0 = (x1 * f(x2) - x2 * f(x1)) / (f(x2) - f(x1));
        c = f(x1) * f(x0);
        x1 = x2; x2 = x0;
        n=n+1;
        if (c == 0) break;
        xm = (x1 * f(x2) - x2 * f(x1)) / (f(x2) - f(x1));
    } while (abs(xm - x0) >= e);
    return xm;
}

real a=SecantMethod(f, x1=0.2, x2=0.5, e=EPS);
real r=(4-a^2)/8;
point A=(-1,0), B=(1,0);
point C=(-sqrt(1-a^2), a), D=(C.x+a,C.y);
point E=(C.x+2*a, C.y), F=(-C.x, C.y), G=(-D.x, D.y);

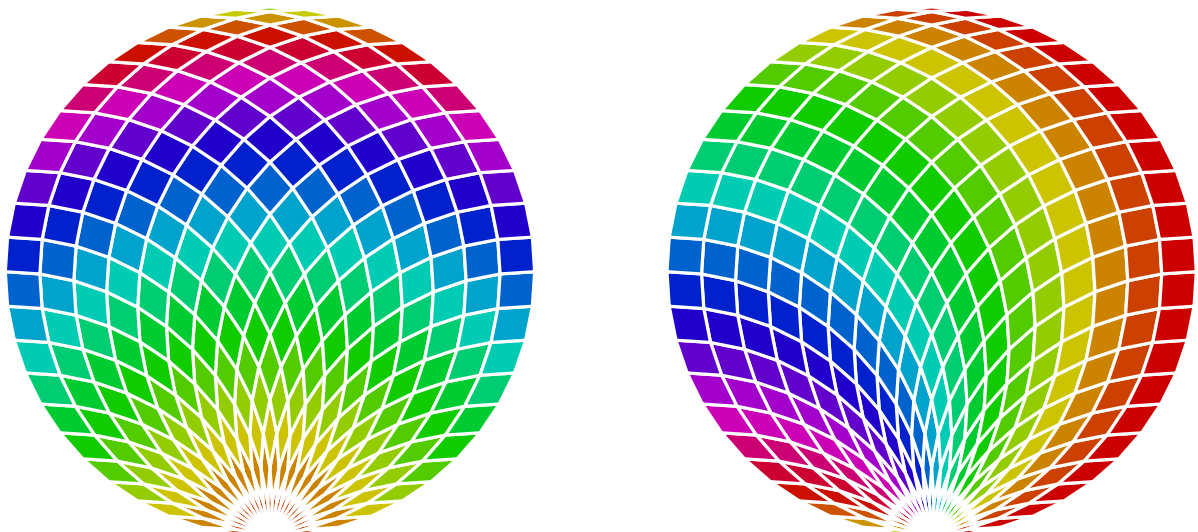
path sq1=C--D--(D.x,0)--(C.x,0)--cycle;
path sq2=D--E--(E.x,0)--(D.x,0)--cycle;
path sq3=F--G--(G.x,0)--(F.x,0)--cycle;
point M=intersectionpoints(circle(E,r), circle(G,r))[1];

filldraw(sq1^^sq2^^sq3, mycolor1+opacity(0.9));
filldraw(circle(M,r), mycolor2+opacity(0.9));
draw(A--B^^arc(co, B, A));

```

3 Problem

P1. Vẽ hình sau:



4 Tài liệu tham khảo

- [1]. Tài liệu Asymptote: the Vector Graphics Language
- [2]. Các ví dụ của Asymptote
- [3]. Tài liệu Asymptote Démarrage rapide
- [4]. Gói geometry.asy và olympiad.asy
- [5]. Tài liệu hướng dẫn gói **geometry.asy**: Euclidean geometry with asymptote
- [6]. Tài liệu hướng dẫn Asymptote 3D: Asymptote 3D, Bruno M. Colombel
- [7]. Tài liệu hướng dẫn ASY (tiếng Việt): ASY for Beginners, Trần Quân