```c
#include<stdio.h>

typedef struct Point{
    int x,y;
}Point;


int parallel(Point a1, Point a2, Point a3, Point a4){
    int slopeDiff;
    slopeDiff = (a2.x - a1.x) * (a4.y - a3.y) - (a4.x - a3.x) * (a2.y - a1.y);
    // Are the slopes the same?
    if(slopeDiff == 0)
        return 1;
    else
        return 0;
}

int perpendicular(Point a1, Point a2, Point a3){
    int check;
    check = (a2.x - a1.x) * (a3.x - a2.x) + (a2.y - a1.y) * (a3.y - a2.y);
    // Slopes must be negative reciprocals
    if(check == 0)
        return 1;
    else
        return 0;
}

int dist(Point a, Point b){
    return (a.x-b.x)*(a.x-b.x) + (a.y-b.y)*(a.y-b.y);
}

int main(){
    int x, y;
    Point p[4];
    for(int i=0; i<4; i++){
        scanf("%d %d", &x, &y);
        p[i].x = x;
        p[i].y = y;
    }

    int flag1 = 0, flag2 = 0, flag3 = 0;
    int s1 = 0, s2 = 0, s3 = 0, s4 = 0;
    flag1 = parallel(p[0], p[1], p[2], p[3]);
    flag2 = parallel(p[1], p[2], p[3], p[0]);
    printf("%d\n", flag1 + flag2);

    flag3 = perpendicular(p[0], p[1], p[2]) || perpendicular(p[1], p[2], p[3]) ||
perpendicular(p[2], p[3], p[0]) || perpendicular(p[3], p[0], p[1]);
    if(flag3)
        printf("YES\n");
    else
        printf("NO\n");

    s1 = dist(p[0], p[1]);
    s2 = dist(p[1], p[2]);
    s3 = dist(p[2], p[3]);
    s4 = dist(p[3], p[0]);

    if(flag1 && flag2){ // Both opposite sides are parallel
        if(s1 == s2 && flag3) // Perpendicular corner and equal sides
            printf("SQUARE");
        else if(flag3) // Perpendicular corner but unequal sides
            printf("RECTANGLE");
        else // No right angled corners
            printf("PARALLELOGRAM");
    }
    else
        printf("TRAPEZIUM");
```

```
    return 0;
}
```