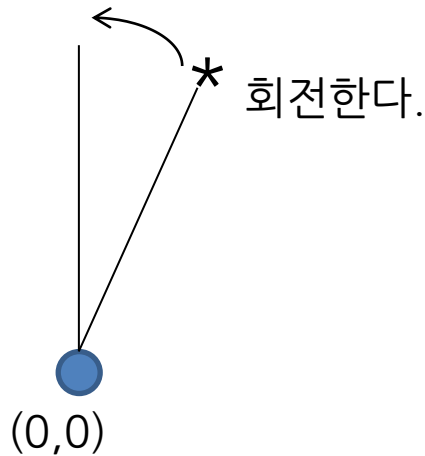


# 오류기 그리기

- ◆ 화면에 오류기를 표시하는 프로그램이다.
- ◆ printf 로 공백의 개수를 직접 넣어 원을 그리거나, 원의 모든 좌표를 직접 입력해서 넣는 것은 안 된다. 계산에 의해 오류기를 그리는 \* 의 좌표를 계산해야 한다.
- ◆ 계산상의 오차로 원이 약간 예쁘지 않을 수 있지만 무관하다.
- ◆ 화면 좌표 제어, 색상 제어는 쓰지 않는다.

## 2) 오류기 그리기

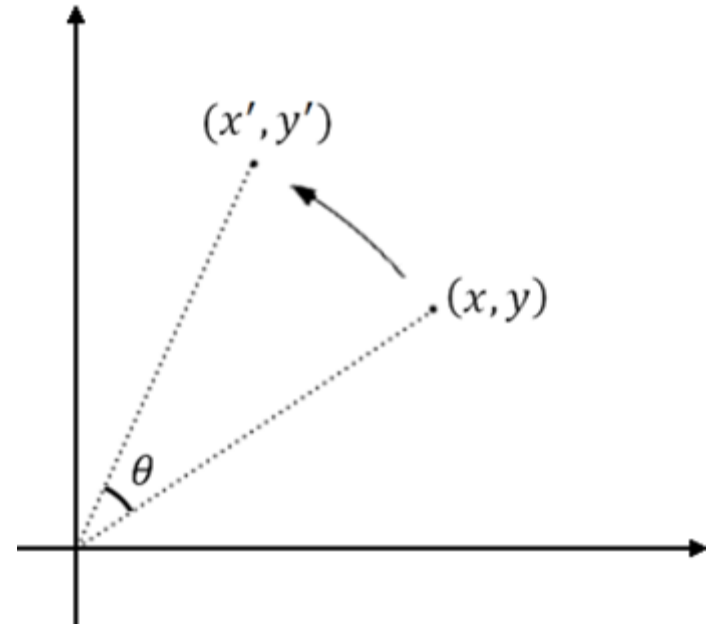
◆ 원은 다음과 같이 그릴 수 있다.



조금씩 회전하면서 그리면 그것이 원이다.

$(x, y)$  좌표를  $\theta$  각도만큼  
반시계 방향으로 회전시킨 좌표

$$\begin{aligned}x' &= x * \cos \theta - y * \sin \theta \\y' &= x * \sin \theta + y * \cos \theta\end{aligned}$$



## 2) 오류기 그리기

- ◆ 원을 구성하는 점의 좌표는 다음과 같이 그릴 수 있다. 중심점이 (0, 0)일 때

```
#include <stdio.h>
#include <math.h>
#define PI 3.1416926535

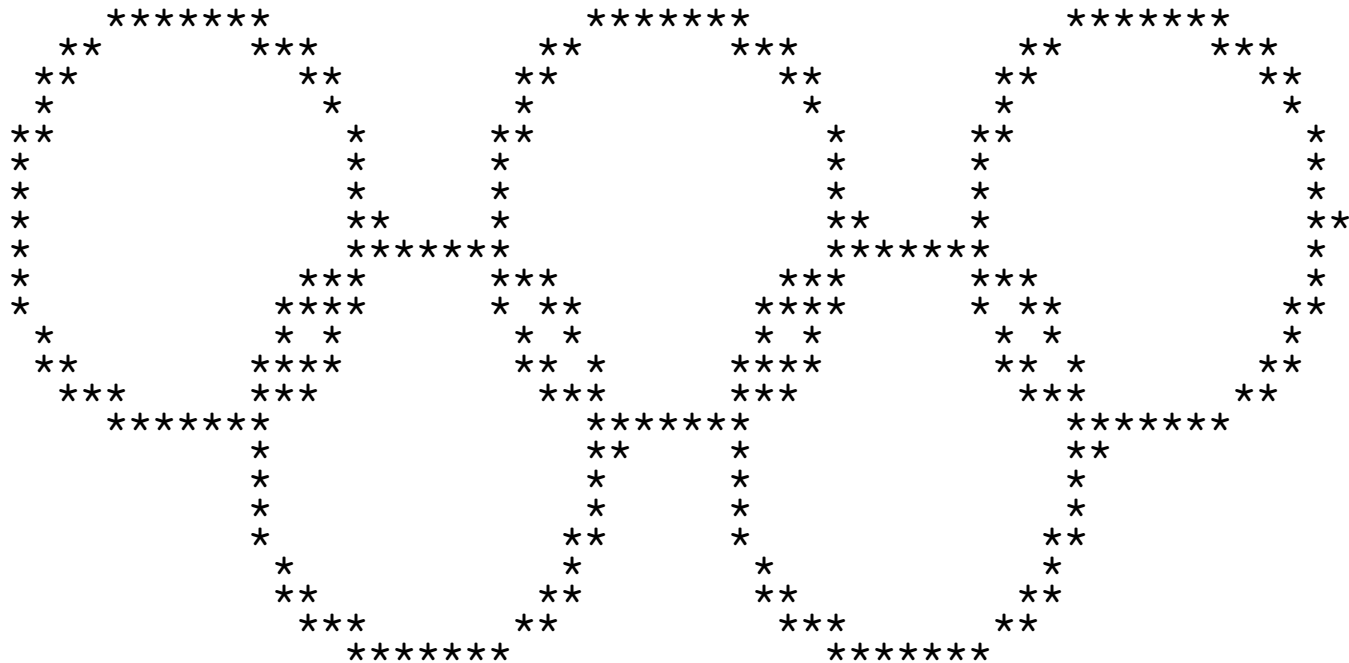
int main() {
    int angle;
    double radian;
    int      newx, newy, oldx, oldy;
    int      radius;

    radius = 10;

    oldx = radius; oldy = 0;

    for (angle = 0; angle < 360; angle += 10) {
        radian = angle / 180.0 * PI;
        newx = oldx * cos(radian) - oldy * sin(radian);
        newy = oldx * sin(radian) + oldy * cos(radian);
        printf("%d %d\n", newx, newy);
    }
}
```

## 2) 오류기 그리기 - 결과



다섯 개의 원의 중심점의 좌표는 각각  $\{15, 8\}$ ,  $\{25, 16\}$ ,  $\{35, 8\}$ ,  $\{45, 16\}$ ,  $\{55, 8\}$  이고 반지름은 8로 그린 결과물이다.

- ◆ 배열에 원을 그리는 함수를 하나 만든 후,
- ◆ 중심점의 좌표를 바꿔가면서 해당 함수를 5회 호출한다.