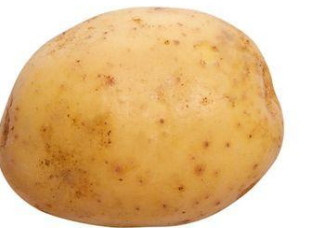


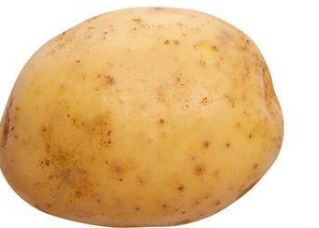
Sports stacking Challenge

포테이토조 김우정 복권근 장세환 최현성



순서

1. 시나리오
2. 챌린지 포인트
3. 시연



시나리오 # 1

1. 홈 포지션 이동
2. 컵 잡기
3. 컵 들기
4. 목표 좌표 이동
5. 힘 제어로 컵 내려놓기
6. 반복

시나리오 # 2

1. 컵 옆으로 이동 후 잡기
2. 목표 좌표 근처로 축 이동
3. 6축 회전을 통한 컵 뒤집기
4. 목표 좌표 위로 이동 후 컵 놓기
5. 컵에 걸리지 않게 위치 이동
6. 홈 포지션 이동

목표 좌표 찾기

목표 위치

target = [600, -120, 190, 154, 180, 175]

홈위치

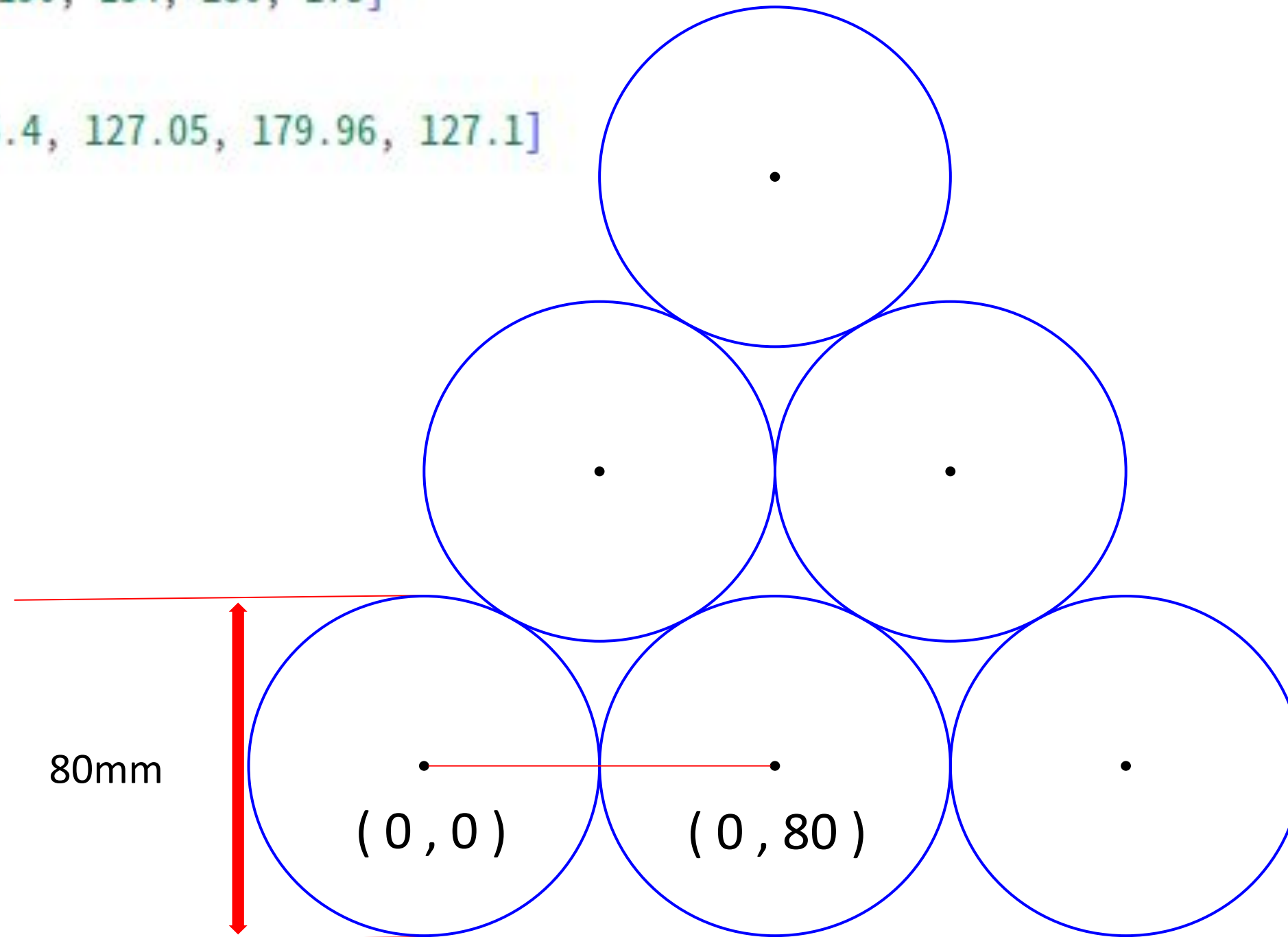
HOME = [340, 100, 195.4, 127.05, 179.96, 127.1]

총 컵 개수

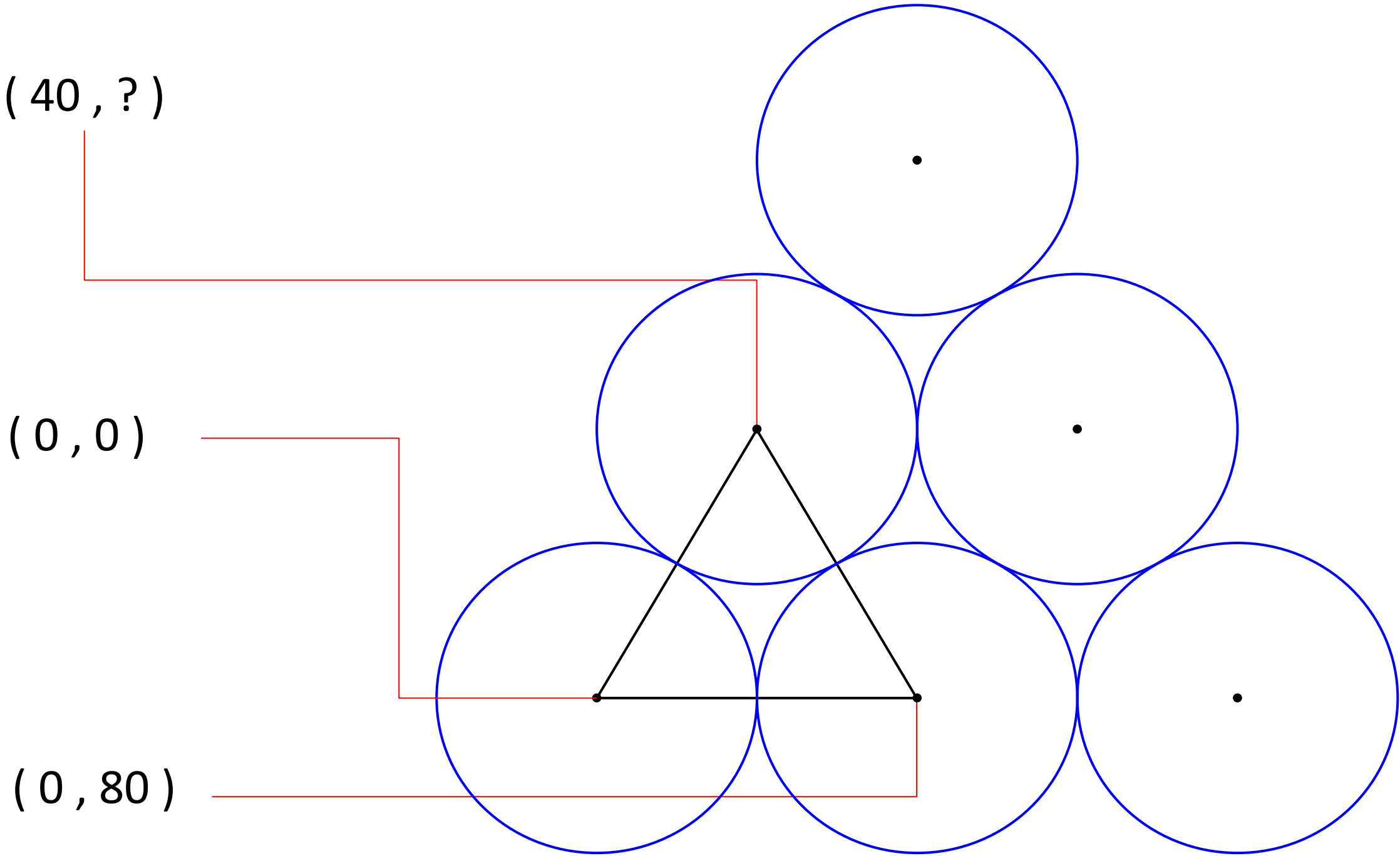
total_cup_num = 11

컵 지름

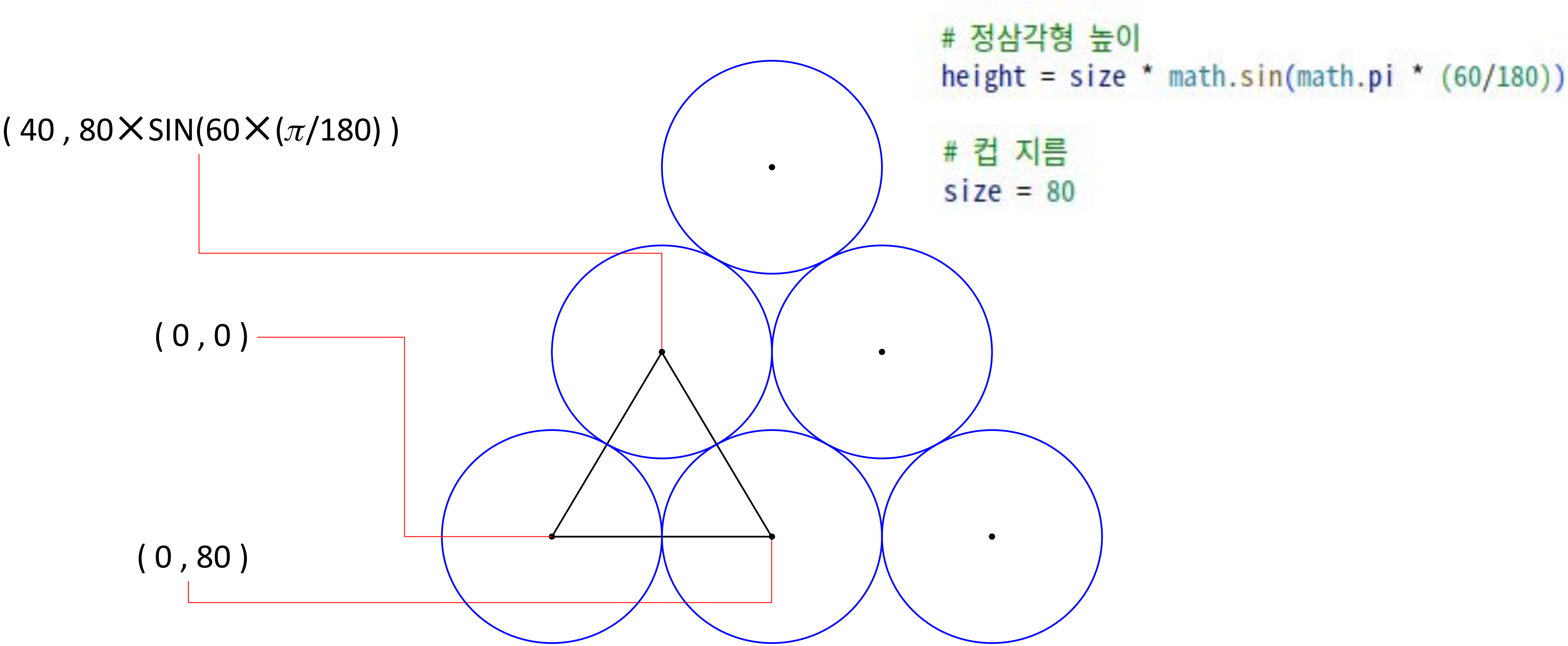
size = 80



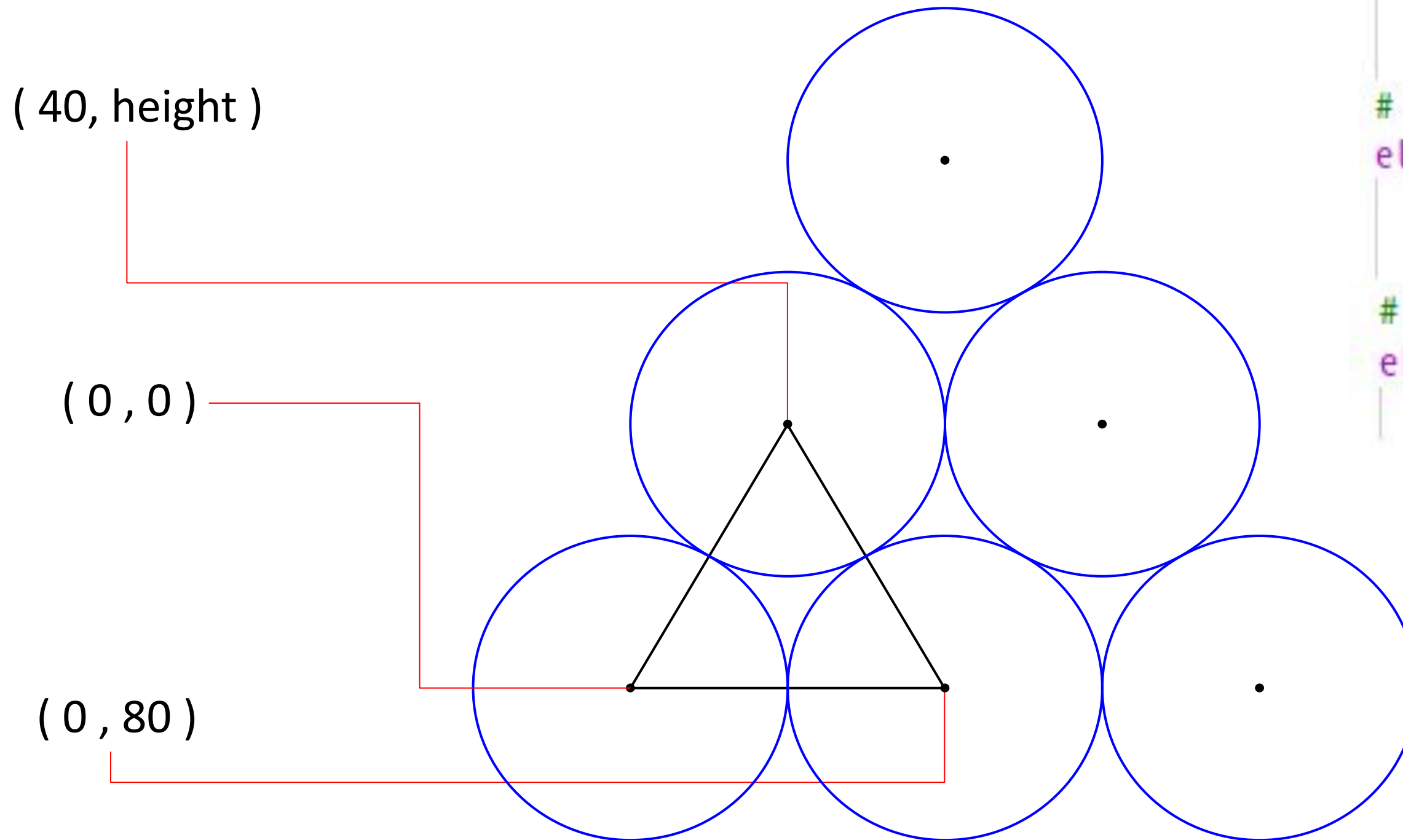
목표 좌표 찾기



목표 좌표 찾기

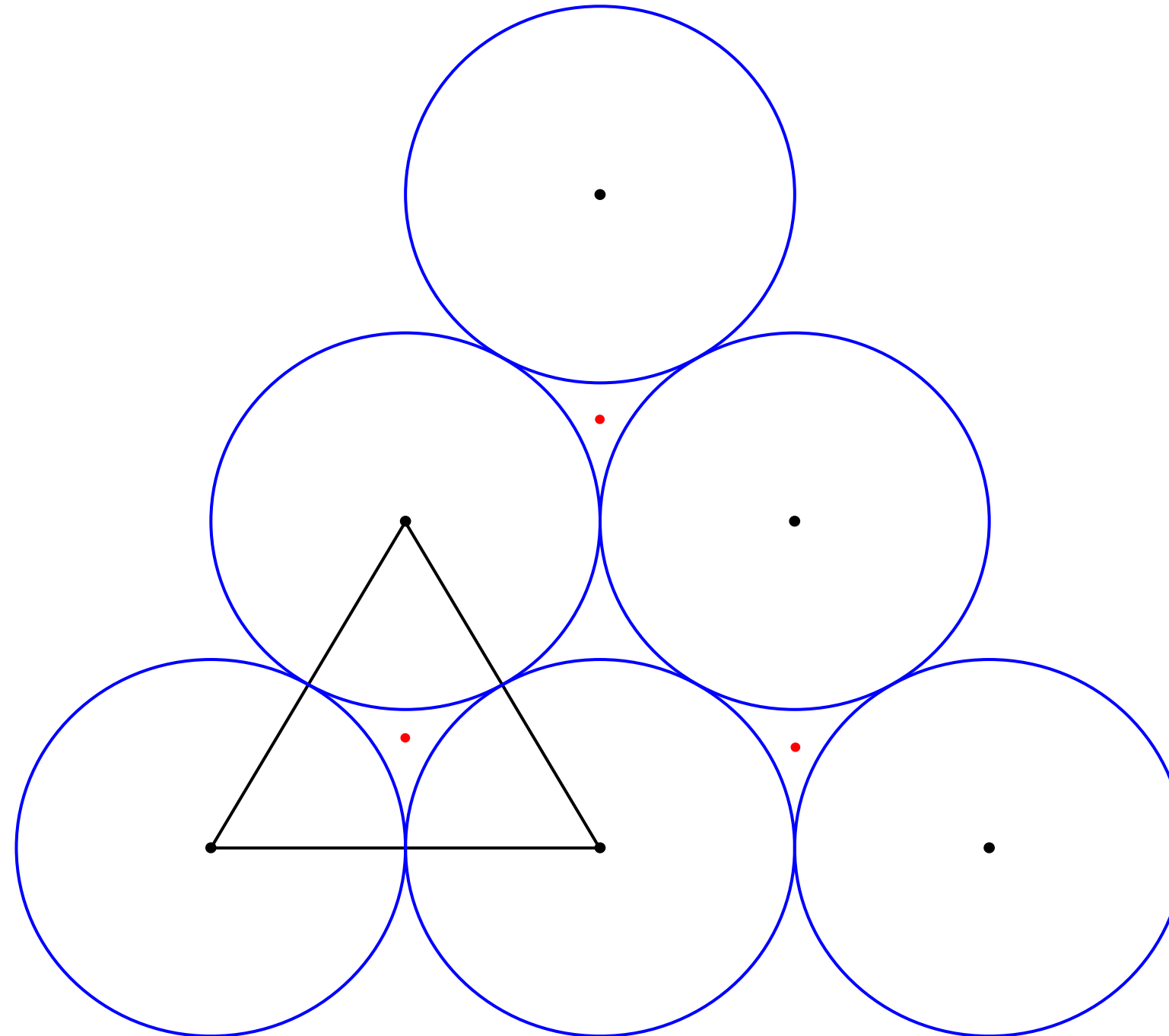


목표 좌표 찾기

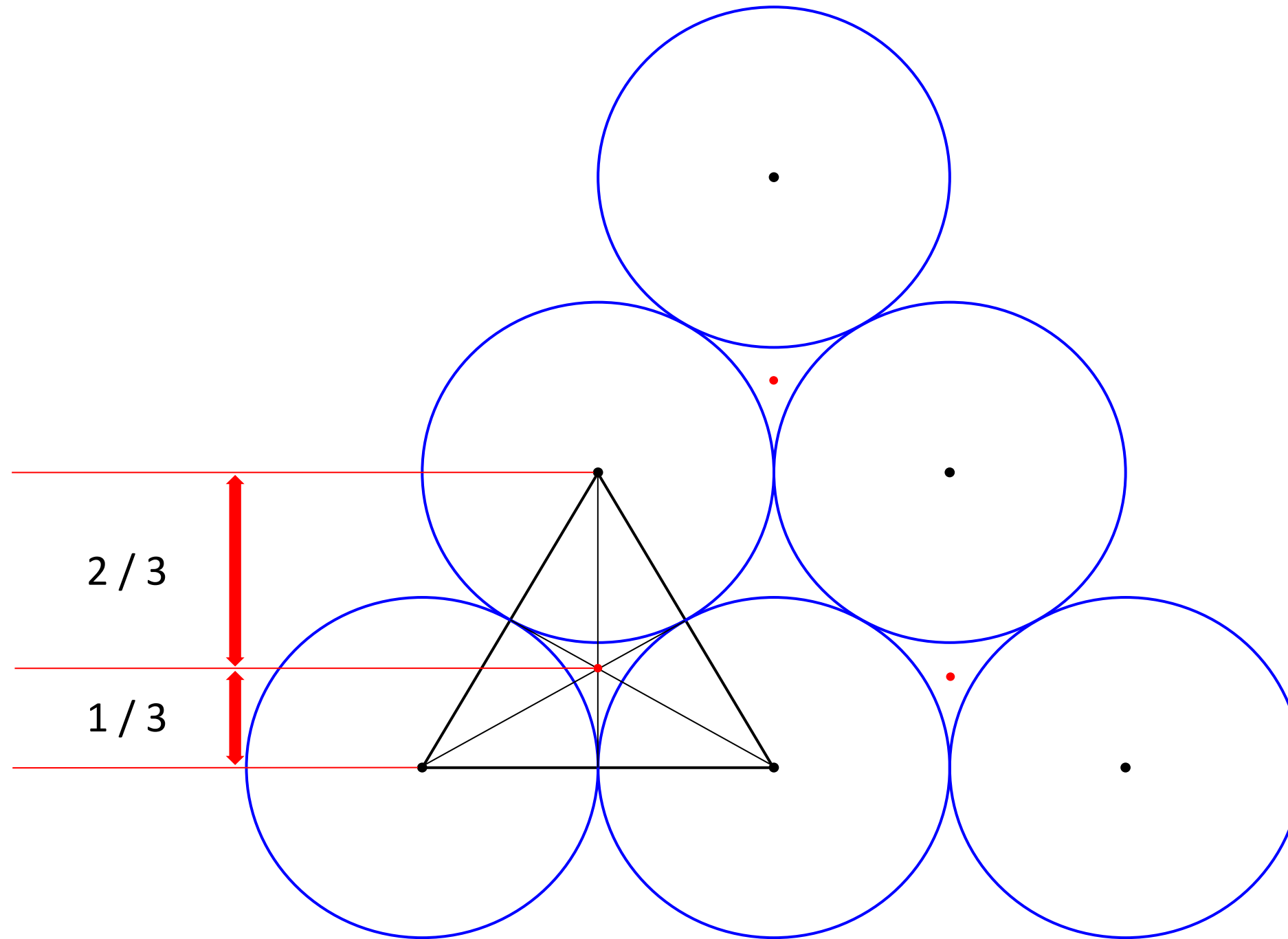


```
# 1층 2개
if cup_num == 2:
    target[0] -= height
    target[1] -= (size//2 + size)
# 1층 1개
elif cup_num == 4:
    target[0] -= height
    target[1] -= (size//2)
# 기본 좌표 수정
else:
    target[1] += size
```


목표 좌표 찾기



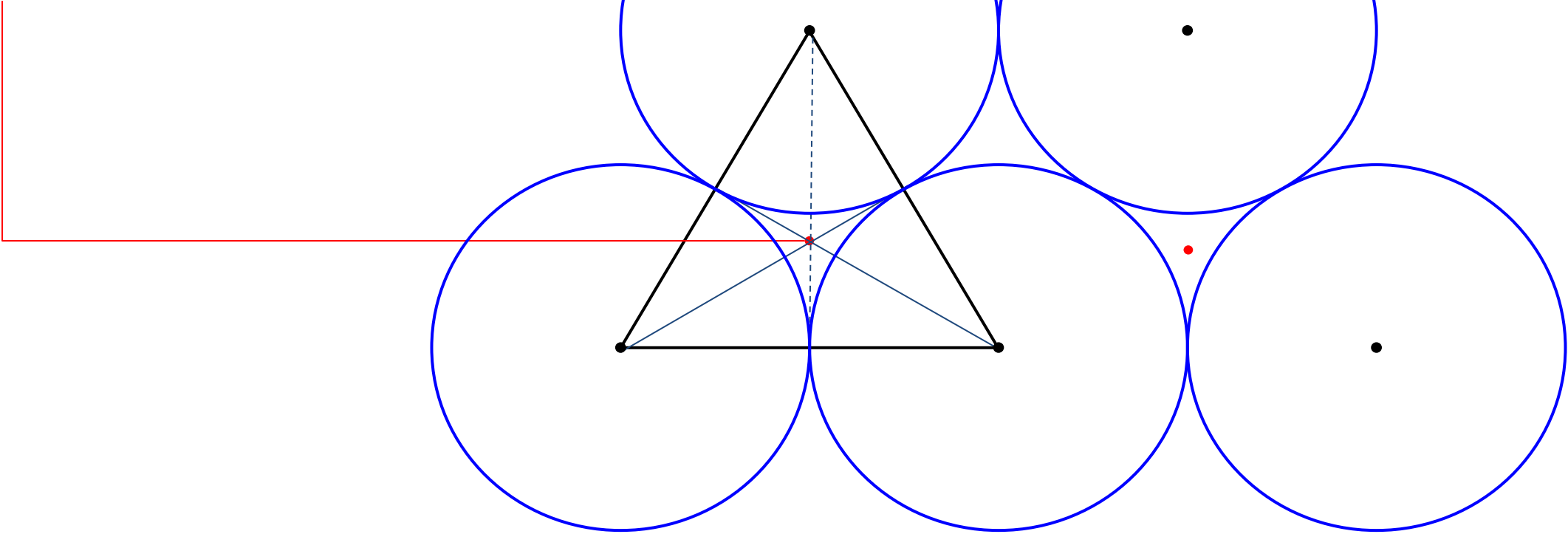
목표 좌표 찾기



```
# 2층 이동
elif cup_num == 5:
    target[0] += (height + height*(2/3))
    target[1] -= (size//2)
    target[2] += 30
# 2층 2개
elif cup_num == 7:
    target[0] -= height
    target[1] -= (size//2)
# 2층 1개
elif cup_num == 8:
    target[0] += (height*(2/3))
    target[2] += 70
# 기본 좌표 수정
else:
    target[1] += size
```

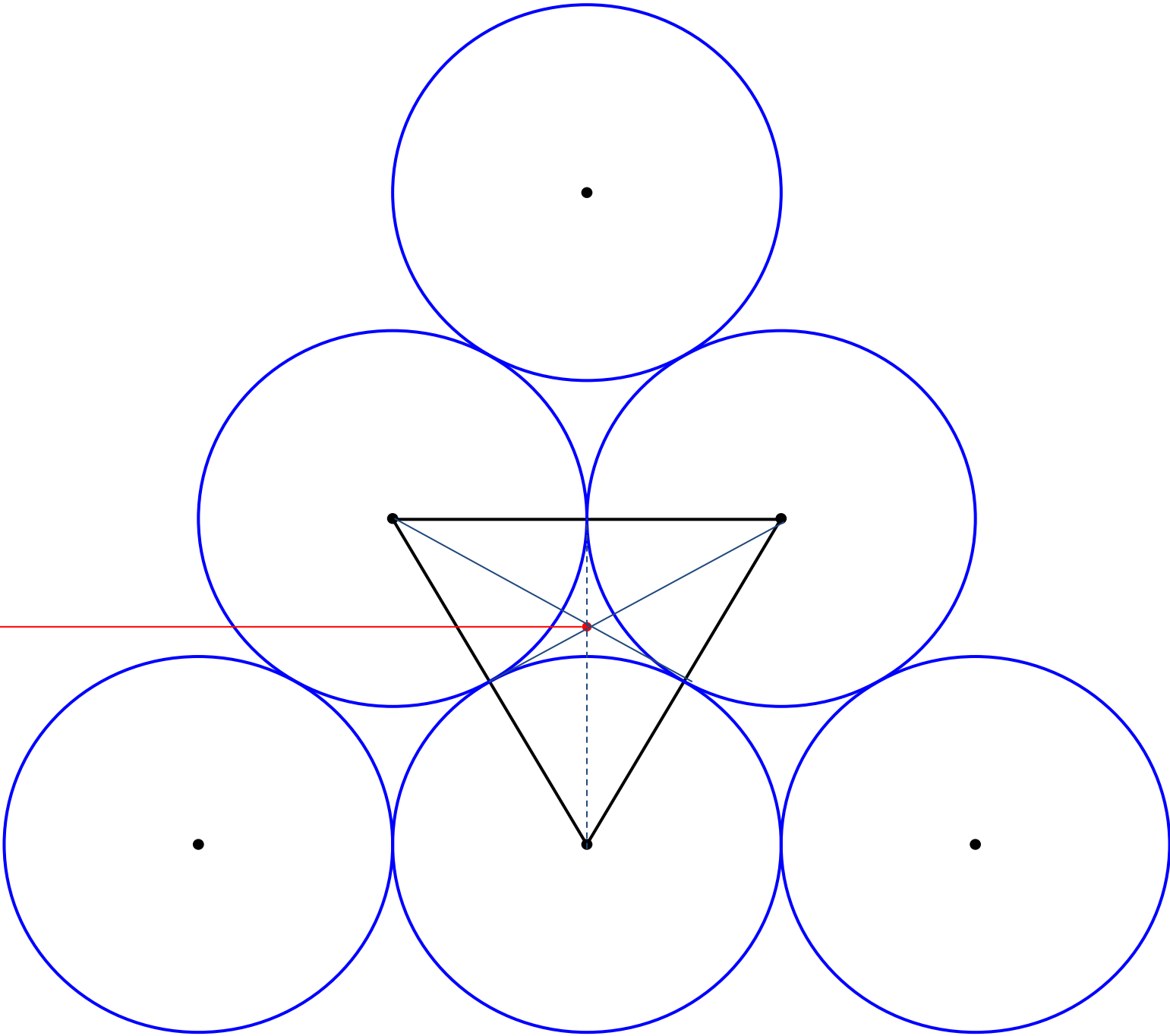
목표 좌표 찾기

(40 , H × ⅓)



목표 좌표 찾기

(40 , H × ⅔)



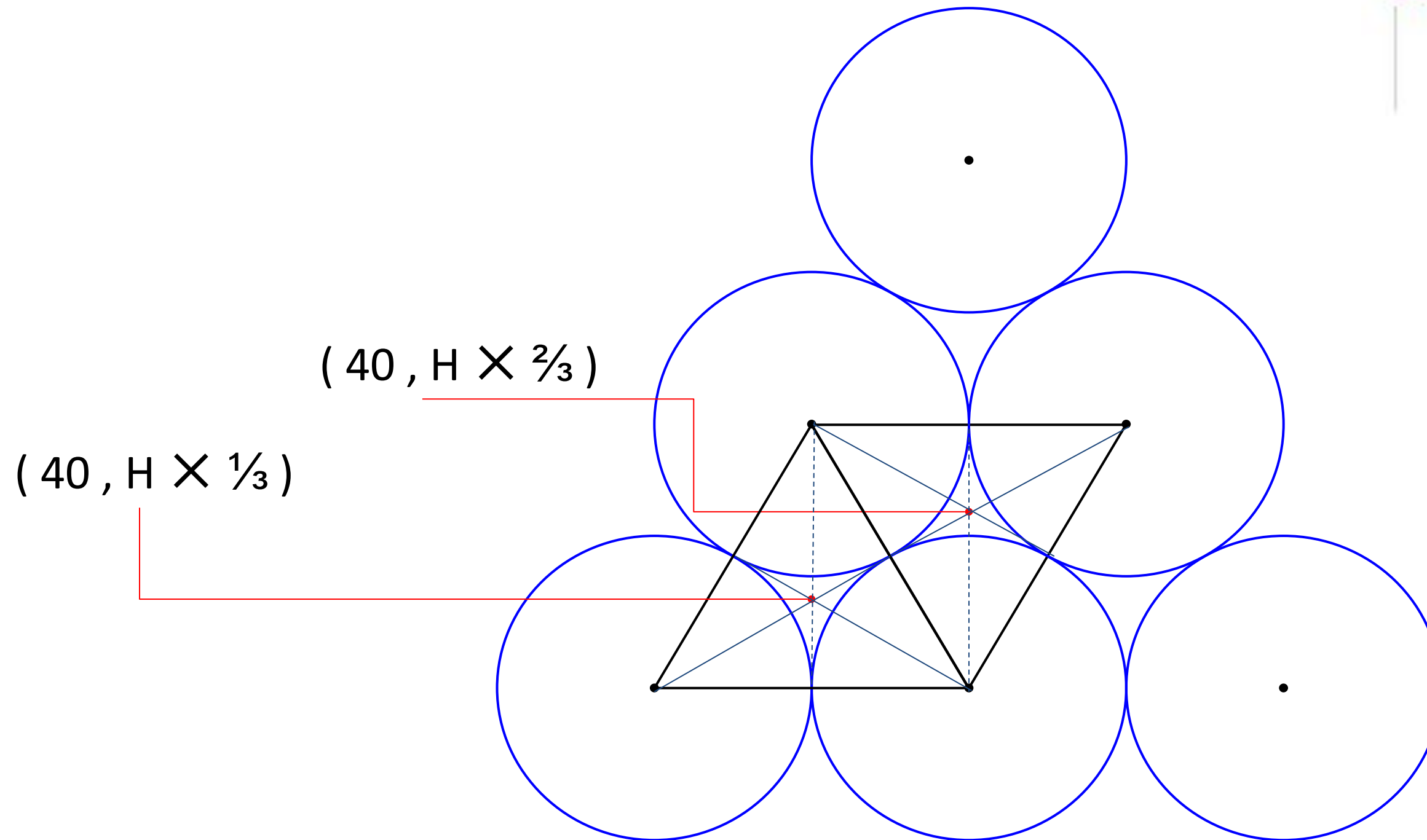
목표 좌표 찾기

3층 이동

```
elif cup_num == 8:
```

```
    target[0] += (height*(2/3))
```

```
    target[2] += 70
```



목표 좌표 찾기

```
while rclpy.ok():  
    for cup_num in range(0, total_cup_num):  
        # 목표 좌표 출력  
        print(target)
```

```
# 1층 2개  
if cup_num == 2:  
    target[0] -= height  
    target[1] -= (size//2 + size)  
# 1층 1개  
elif cup_num == 4:  
    target[0] -= height  
    target[1] -= (size//2)  
# 2층 이동  
elif cup_num == 5:  
    target[0] += (height + height*(2/3))  
    target[1] -= (size//2)  
    target[2] += 30  
# 2층 2개  
elif cup_num == 7:  
    target[0] -= height  
    target[1] -= (size//2)  
# 2층 1개  
elif cup_num == 8:  
    target[0] += (height*(2/3))  
    target[2] += 70  
# 3층 1개  
elif cup_num == 9:  
    pass  
# 기본 좌표 수정  
else:  
    target[1] += size
```

목표 좌표 찾기

```
# 컵 뒤집기
if cup_num == 10:

    # 컵 옆으로 이동
    movej(filp_robot, vel = 10, acc = 10)
    movel([0,0,-10,0,0,0], vel = 10, acc = 10, mod = DR_FC_MOD_REL)

    # 그림 닫기
    grip()

    # 수직으로 위로 이동
    movel([0,0,210,0,0,0], vel = VELOCITY, acc = ACC, mod = DR_FC_MOD_REL)

    # 컵 뒤집기
    movej([0,0,0,0,0, -180], vel = 100, acc = 100, mod = DR_FC_MOD_REL)

    # 쌓은 것 옆으로 이동
    movel([target[0]-HOME[0], -(HOME[1]-target[1]),0,0,0,0], vel = 100, acc = 100, mod = DR_FC_MOD_REL)

    # 그림 열기
    release()

    # 옆으로 빠지기
    movel([0,70,0,0,0,0], vel = 100, acc = 100, mod = DR_FC_MOD_REL)
    movel([-100,0,0,0,0,0], vel = 100, acc = 100, mod = DR_FC_MOD_REL)

    # 홈으로 이동
    movej([0,0,90,0,90,0], vel = 30, acc = 30)

return
```


목표 좌표 찾기

```
# 홈위치로 이동
move1(trans(HOME, [0,0,50,0,0,0], DR_BASE, DR_BASE), vel = VELOCITY, acc = ACC)

# 그립 열기
release()

# 아래로 이동
move1(trans(HOME, [0,0,-3-11*cup_num,0,0,0], DR_BASE, DR_BASE), vel = VELOCITY, acc = ACC)

# 그립 닫기
grip()

# 수직으로 위로 이동
move1(trans(HOME, [0,0,130,0,0,0], DR_BASE, DR_BASE), vel = VELOCITY, acc = ACC)

# 목표 좌표 이동
move1(target, vel = VELOCITY, acc = ACC)

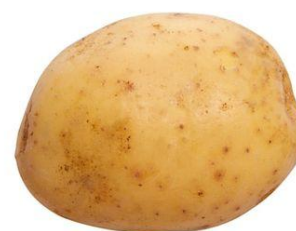
# 힘제어 키기
task_compliance_ctrl()
set_desired_force([0,0,-60,0,0,0], [0,0,1,0,0,0], mod = DR_FC_MOD_REL)

# 바닥에 닿을 때까지 아래로
while not check_force_condition(DR_AXIS_Z, max = 10):
    pass

# 힘제어 끄기
release_compliance_ctrl()
wait(1)

# 그립 열기
release()

# 수직으로 위로 이동
move1(trans(target, [0,0,31,0,0,0], DR_BASE, DR_BASE), vel = VELOCITY, acc = ACC)
```

Thank You

포테이토조