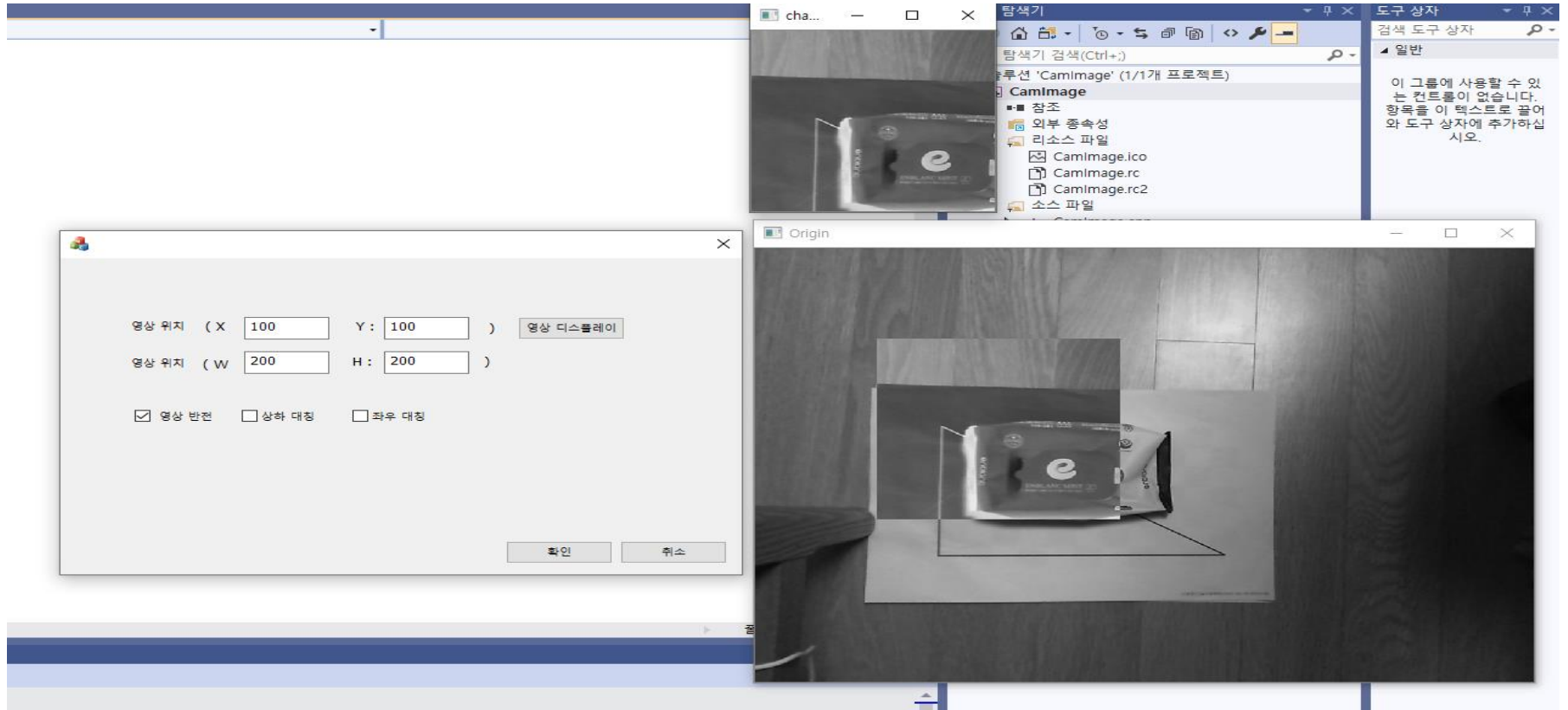


# C프로젝트 Part1 실습 결과물

---

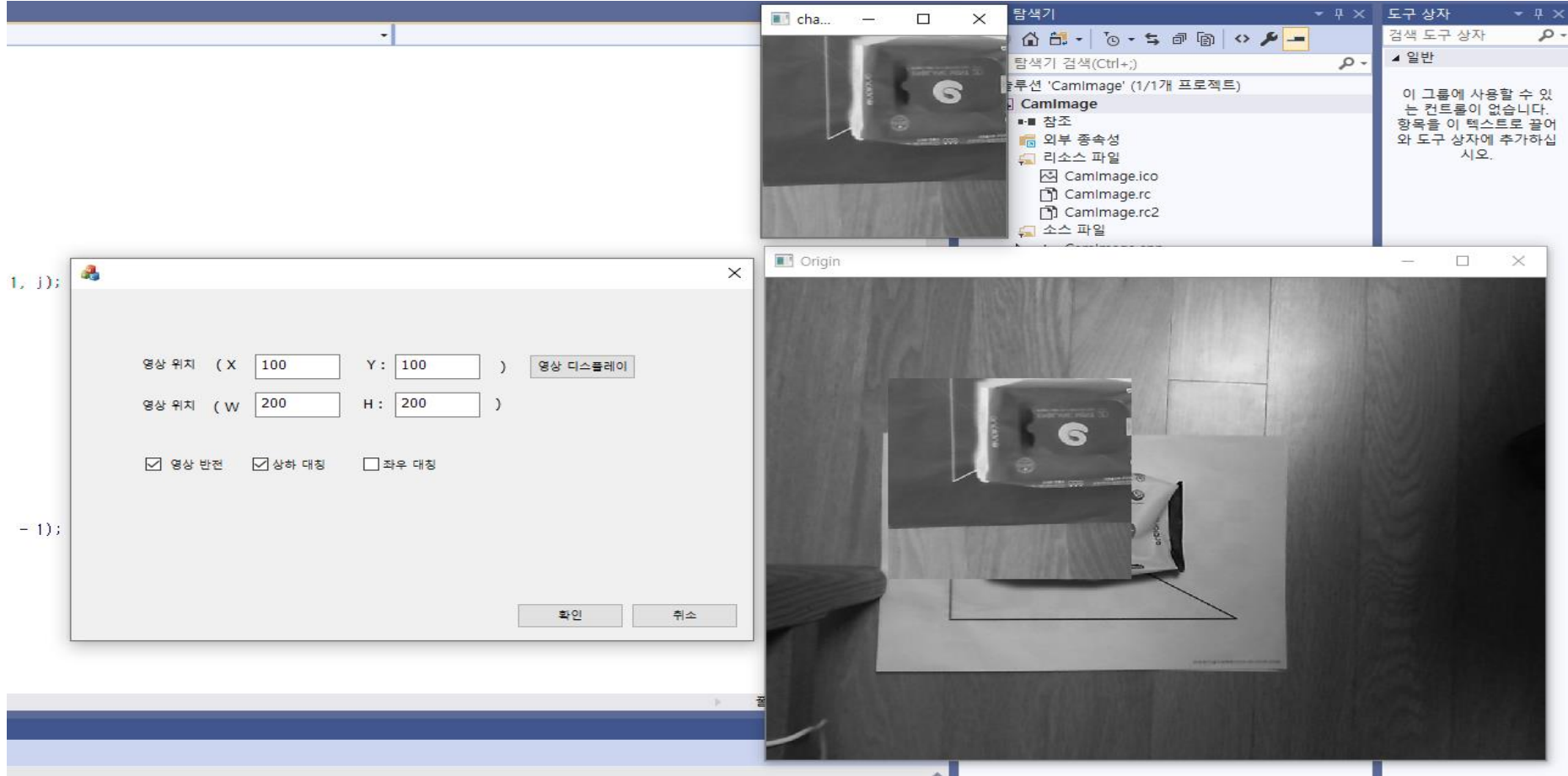
By 강호용

# 실습1번 (영상반전, 상하, 좌우 대칭)



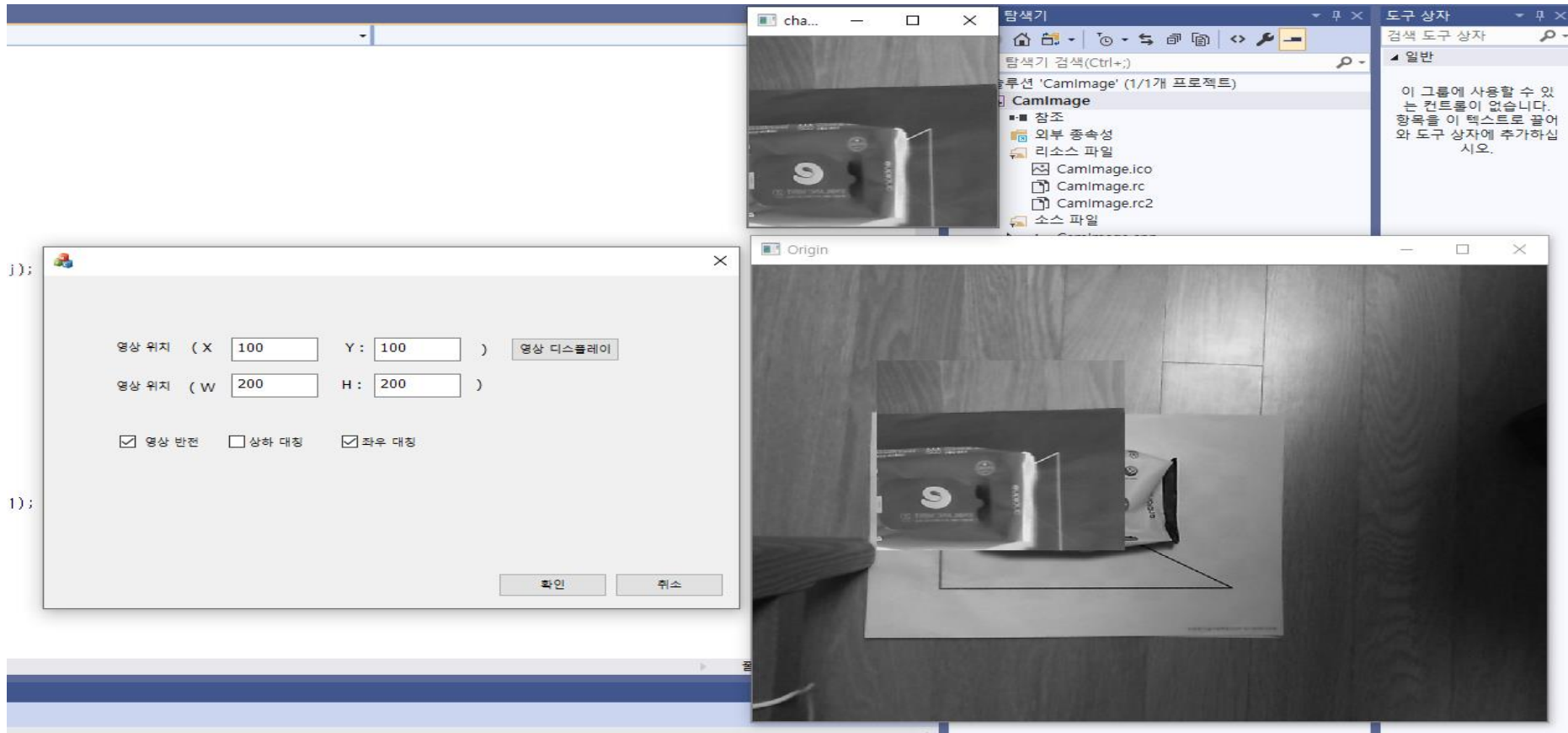
## 영상 반전 체크박스 선택 시 결과물

# 실습1번 (영상반전, 상하, 좌우 대칭)



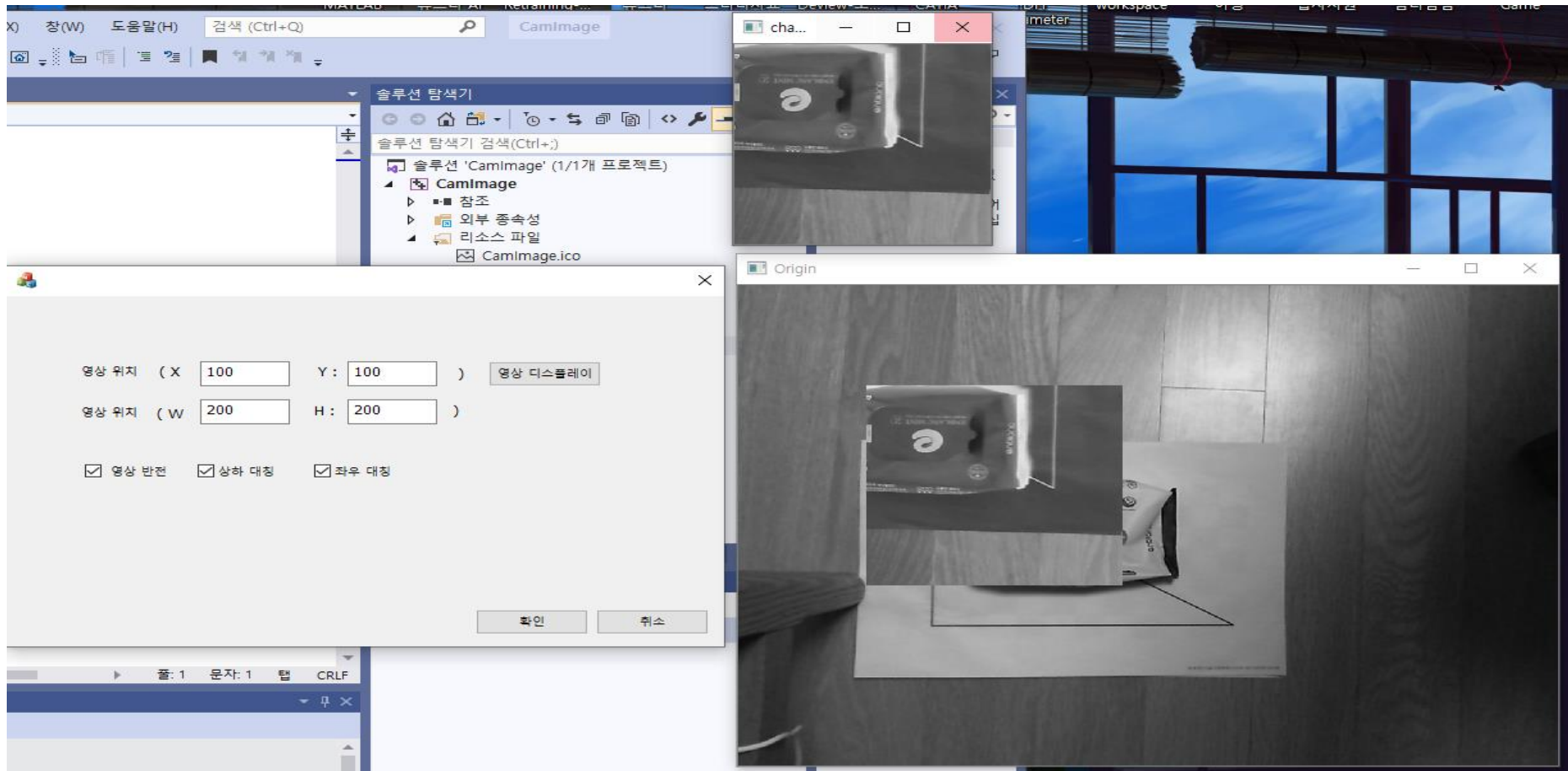
영상 반전 및 상하 대칭 체크박스 선택 시 결과물

# 실습1번 (영상반전, 상하, 좌우 대칭)



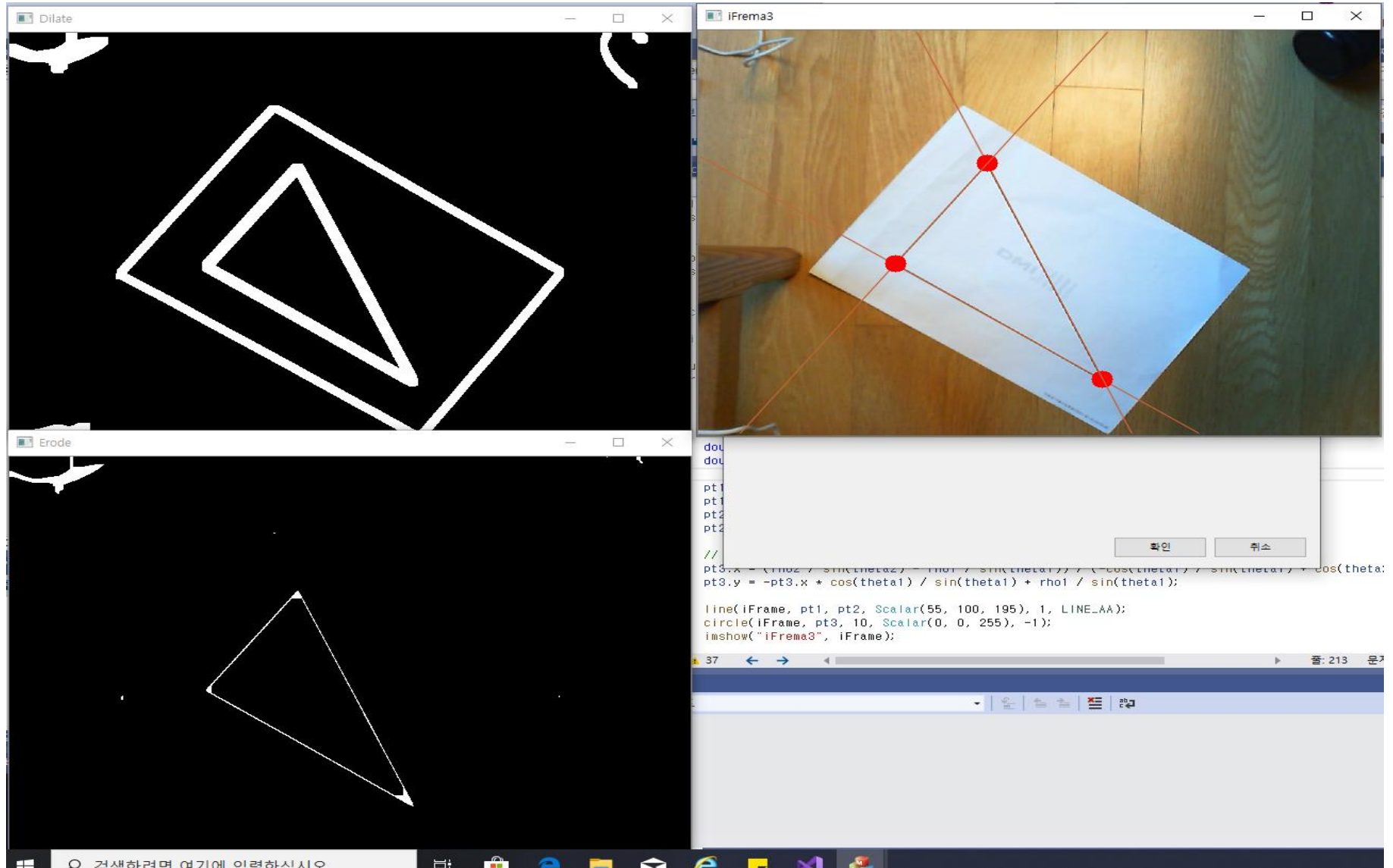
영상 반전 및 좌우 대칭 체크박스 선택 시 결과물

# 실습1번 (영상반전, 상하, 좌우 대칭)

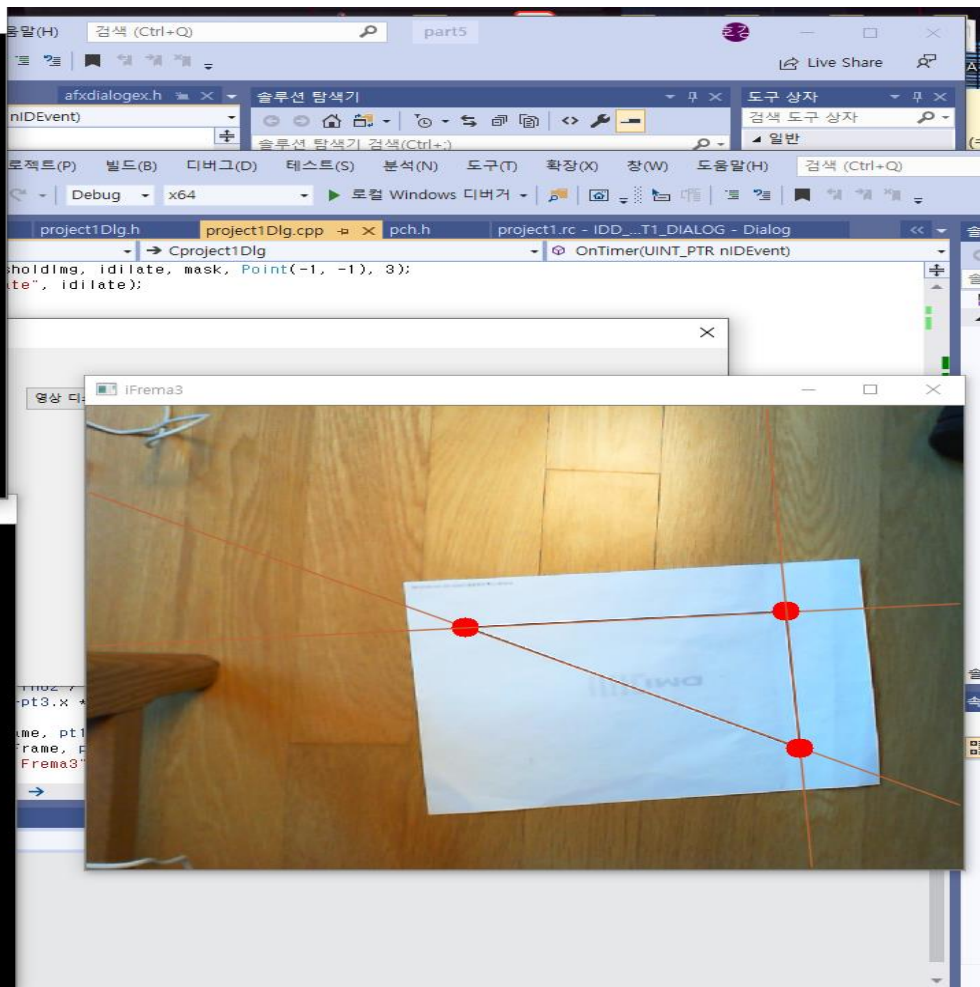
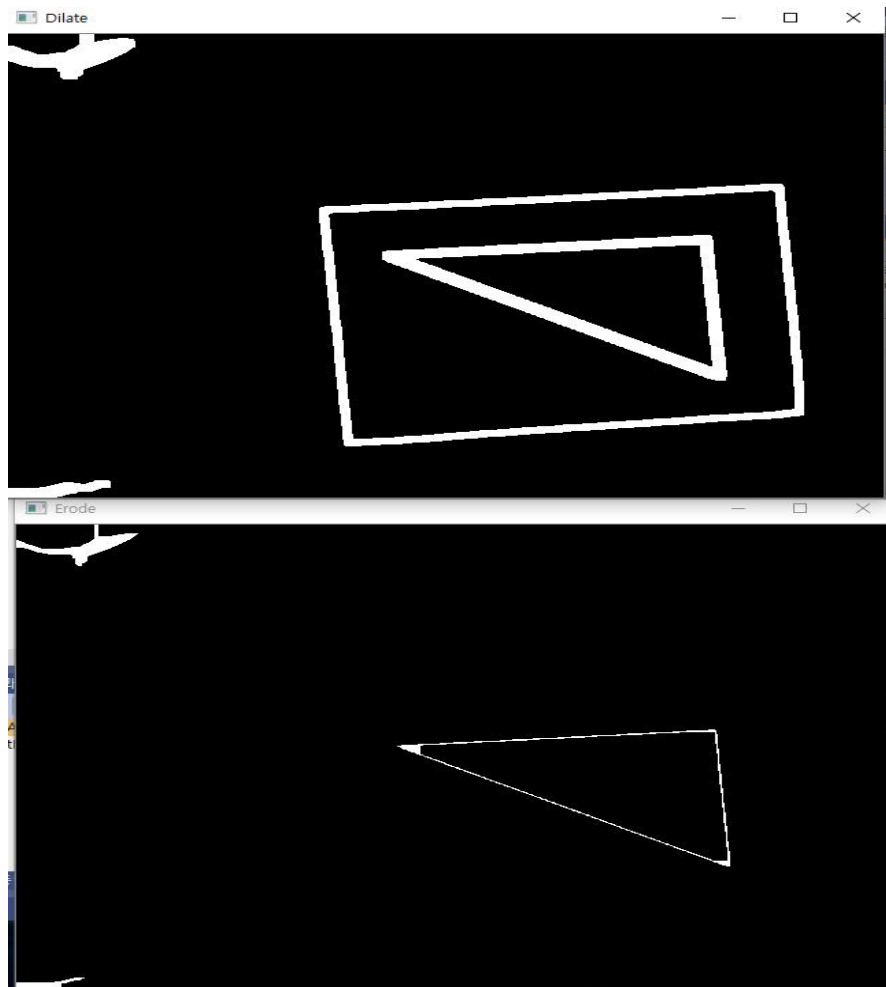


영상 반전, 좌우, 상하 체크박스 선택 시 결과물

# 실습2번 (삼각형 꼭지점 검출)



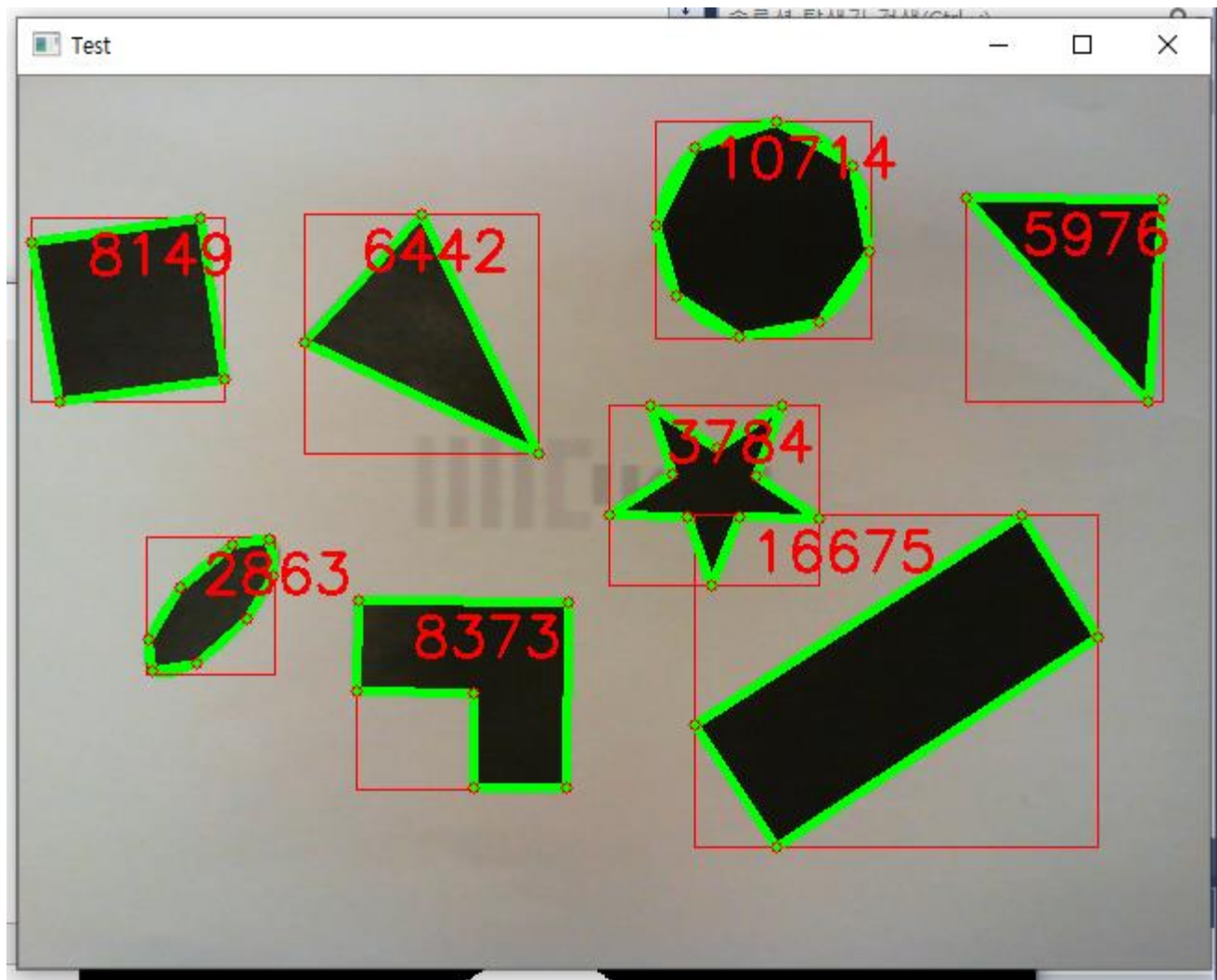
# 실습2번 (삼각형 꼭지점 검출)



이미지를 반대로 하여도 꼭지점 검출



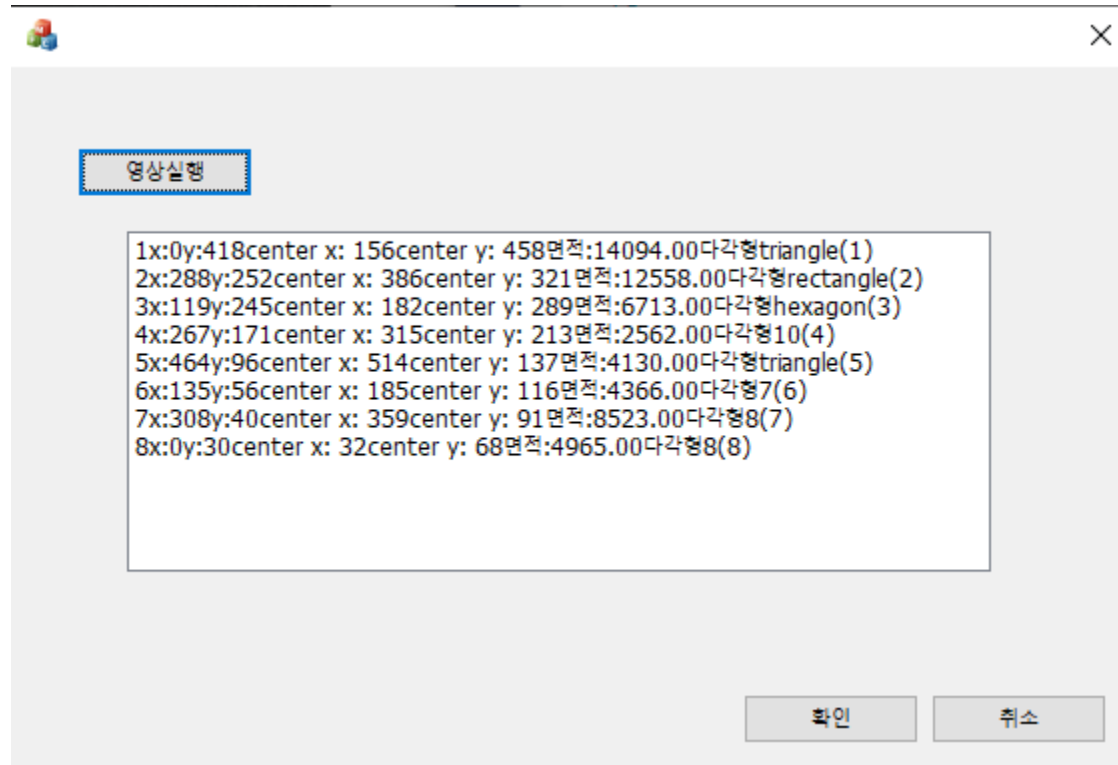
# 실습3번 (다각형 검출)



다각형 검출 및 각 다각형 면적

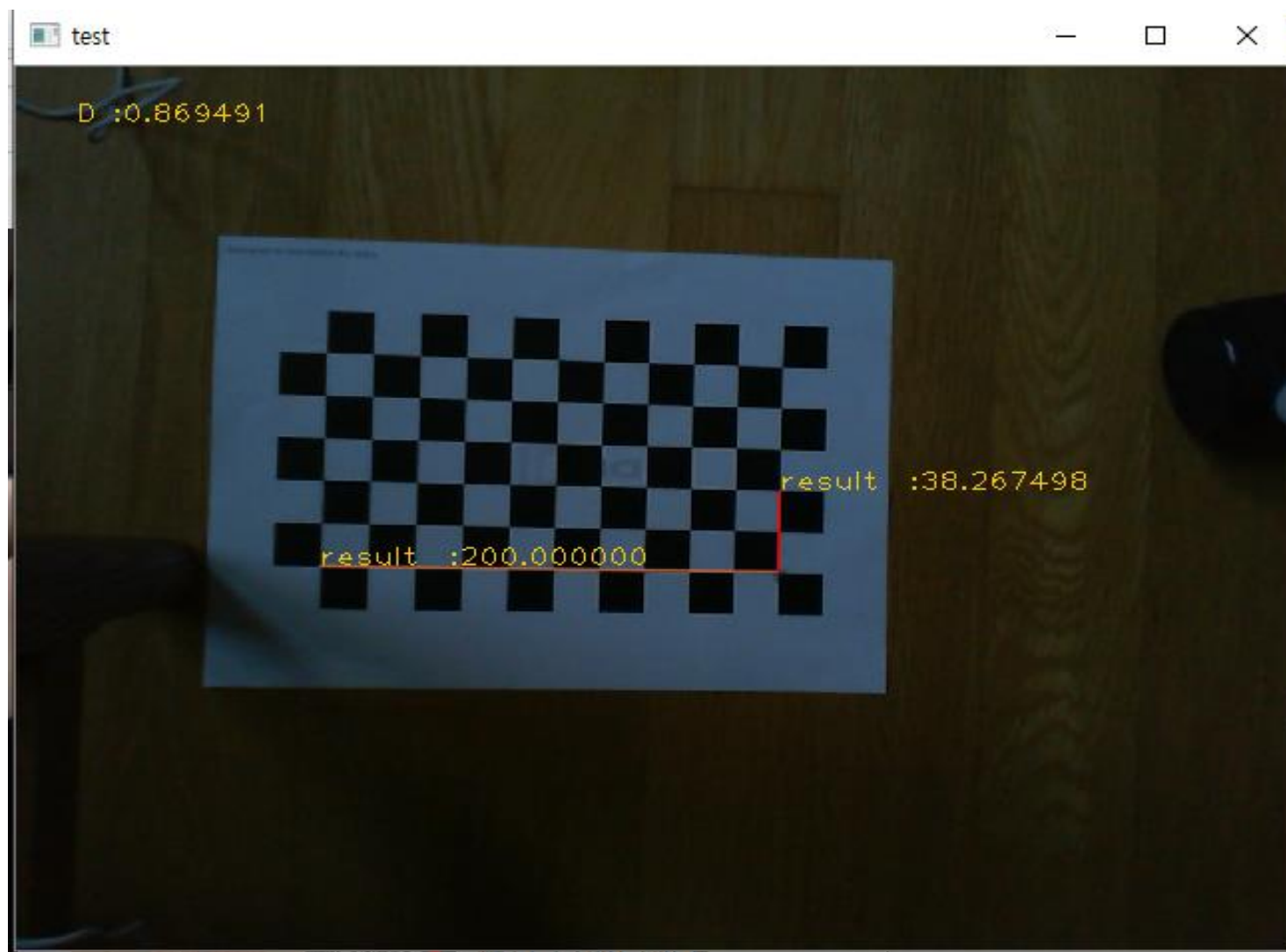


# 실습3번 (다각형 검출 정보 리스트 박스 출력)



리스트 박스 이용한 다각형 정보 출력

## 실습4번 (스케일 값 구하기)



## 실습4번 (스케일 값 구하고 저장하기)

스케일 값  
행렬  
 $X_{\text{값}}$   $y_{\text{값}}$



Save

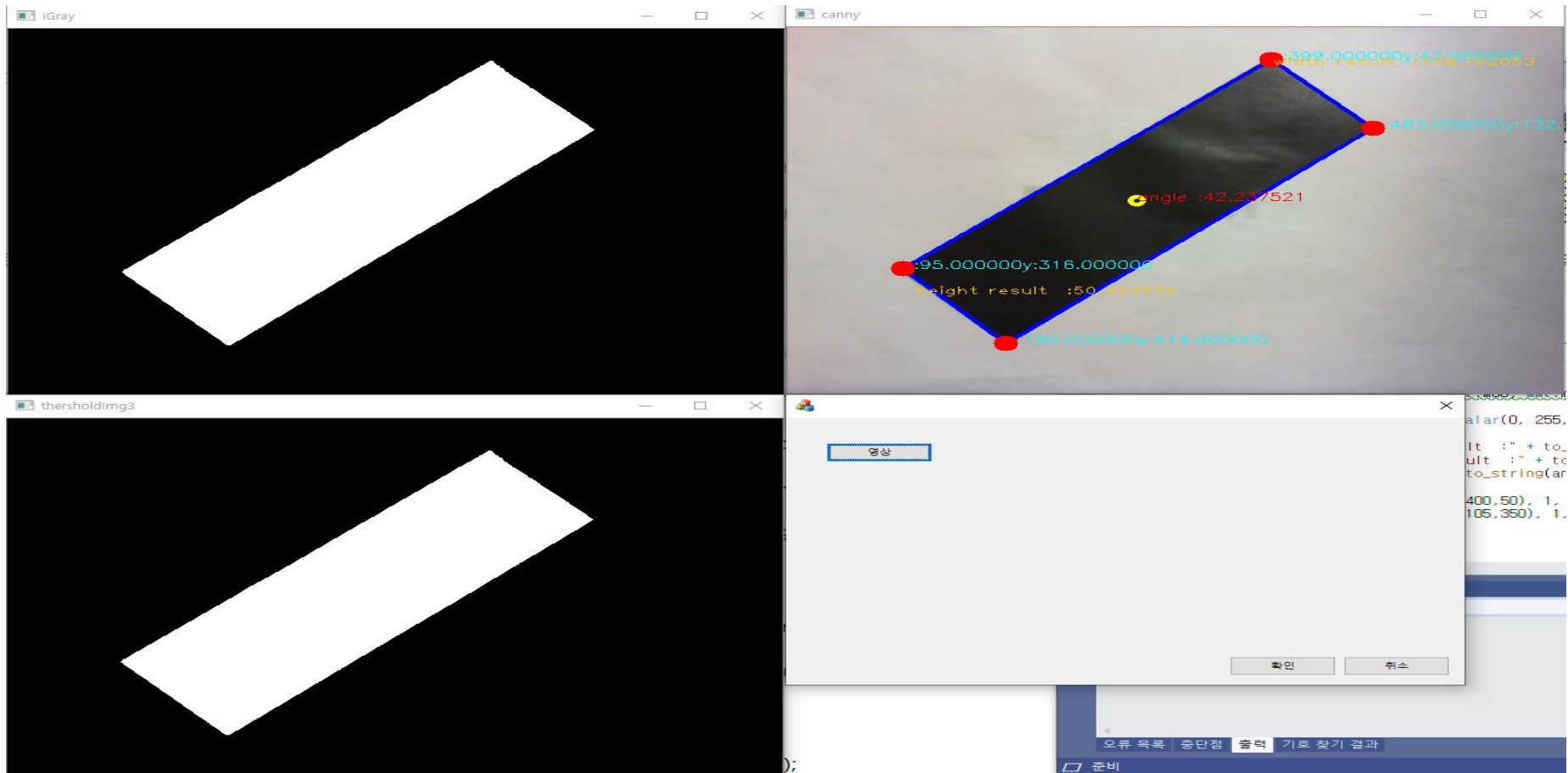


intrinsics.yml

```
171 result 0 0 - 10 10 : 4.0535743713378906e+01
172 result 0 0 - 22 22 : 200.
173 D: 9.0057200193405151e-01
```

두개 의 좌표 거리 측정 및 스케일 D 값

# 실습5번 (사각형 가로세로길이 추출 및 각도측정)



영상 측정 시 사각형이 화면안에 들어오도록 하고 측정하였습니다.  
<빛 으로 인한 사각형 모양에 외각라인이 안나 와서 사진을 이용하여 측정>

**\* (기존 위치와 다르므로) 스케일 값을 다시 측정하여 계산**

# 실습5번 (사각형 가로세로길이 추출 및 각도측정)

\* 두개 의 좌표 거리 측정 (좌표 값)을 이용한 거리측정

```
d-> sqrt(pow((approx[0].x - approx[1].x), 2) +(pow((approx[0].y - approx[1]), 2)));  
d2-> sqrt(pow((approx[2].x - approx[3].x), 2) +(pow((approx[2].y - approx[3]), 2)));
```

```
float D = 0.388; // 스케일 값
```

\* 두개 의 좌표 거리 측정 결과 가로 세로 값

```
D2 = d * D;  
D3 = d2 * D;
```

\* 화면 출력용

```
String str1 = "width result  :" + to_string(D2);  
String str2 = "height result  :" + to_string(D3);
```

```
putText(dst, str1, Point(400,50), 1, 1, Scalar(0, 200, 255));  
putText(dst, str2, Point(105,350), 1, 1, Scalar(0, 200, 255));
```

## 실습5번 (각도 측정 코드)

// 각도 측정 부분

```
Float angle = (approx.[3].x - approx.[2].x) * D;  D->스케일 값
```

```
Float value = angle / D2;      D2 -> D2 = d * D(스케일)
                                d -> sqrt(pow((approx[0].x - approx[1].x), 2) +
Double theta = acos(value);    (pow((approx[0].y - approx[1].y), 2)));
```

```
Double angleValue = theta * 180 / CV_PI;
```

// 가운데 위치를 알기위해서 (각도 표기위치를 넣기 위한 위치코드)

```
Moments mmt = moments(approx);
```

```
Point center(mmt.m10 / mmt.m00, mmt.m01 / mmt.m00);
```

```
circle(dst, center, 5, Scalar(0, 255, 255), 3);
```

// 가운데 원점 위치에 구한 각도를 표기

```
putText(dst, "angle :" + to_string(angleValue), center, FONT_HERSHEY_SIMPLEX, 0.5, Scalar(0, 0, 255), 1);
```