

# 2. 실습 보고서  
- 소수 찾기 실습 -

보 고 자 성 명	20181151 김 지 원
보 고 자 학 과	산 업 경 영 공 학 과
담 당 교 수 성 명	정 진 만 교 수 님
과 목 명	모 바 일 프 로 그 래 밍 01분 반
제 출 날 짜	2020.03.25

- 목차 -

1. 개요	-----	3p
2. 소스 코드( MainActivity )	-----	4p
3. 디버깅	-----	5p
4. AVD( Android Virtual Device ) 결과 화면	-----	8p

## 1. 개요

1부터 10000까지 수 중에서 소수를 찾아 그 개수를 헤아리기 위한 소스코드를 작성하였다. 소수 중에서 2는 유일한 짝수이므로 미리 count 한 후 3부터 소수인지 판단하는 알고리즘을 통해 확인한다. 또한, 짝수는 약수로 무조건 약수로 2를 포함하고 있어 홀수에서만 판단한다. 홀수 중 해당 수의 제곱근보다 작은 수(1을 제외한)로 나누었을 때 나누어떨어지지 않을 경우, 소수임을 확인하여 개수를 count 한다.

## 2. 소스 코드( MainActivity )

```
package com.example.primality;

import androidx.appcompat.app.AppCompatActivity;

import android.os.Bundle;
import android.widget.TextView;

public class MainActivity extends AppCompatActivity {

    TextView text1, text2, text3;
    int result;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        text1 = findViewById(R.id.textView1);
        text2 = findViewById(R.id.textView2);
        text3 = findViewById(R.id.textView3);

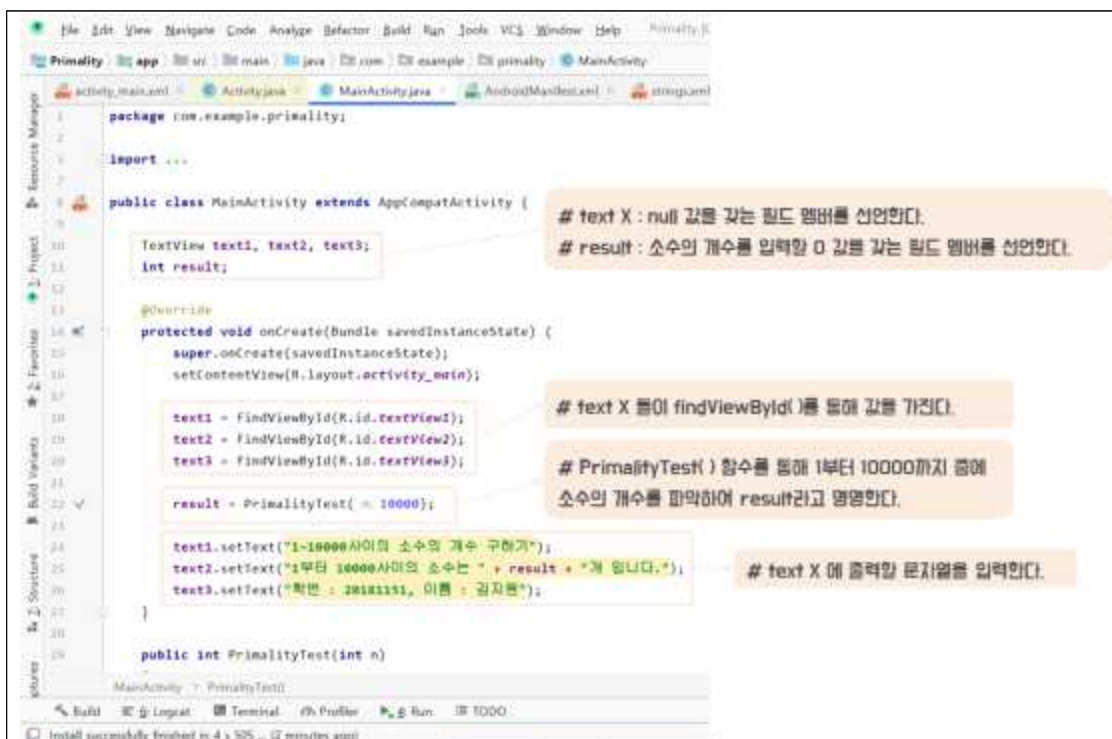
        result = PrimalityTest(10000);

        text1.setText("1~10000사이의 소수의 개수 구하기");
        text2.setText("1부터10000사이의 소수는" + result + "개 입니다.");
        text3.setText("학번: 20181151, 이름: 김지원");
    }
}
```

```

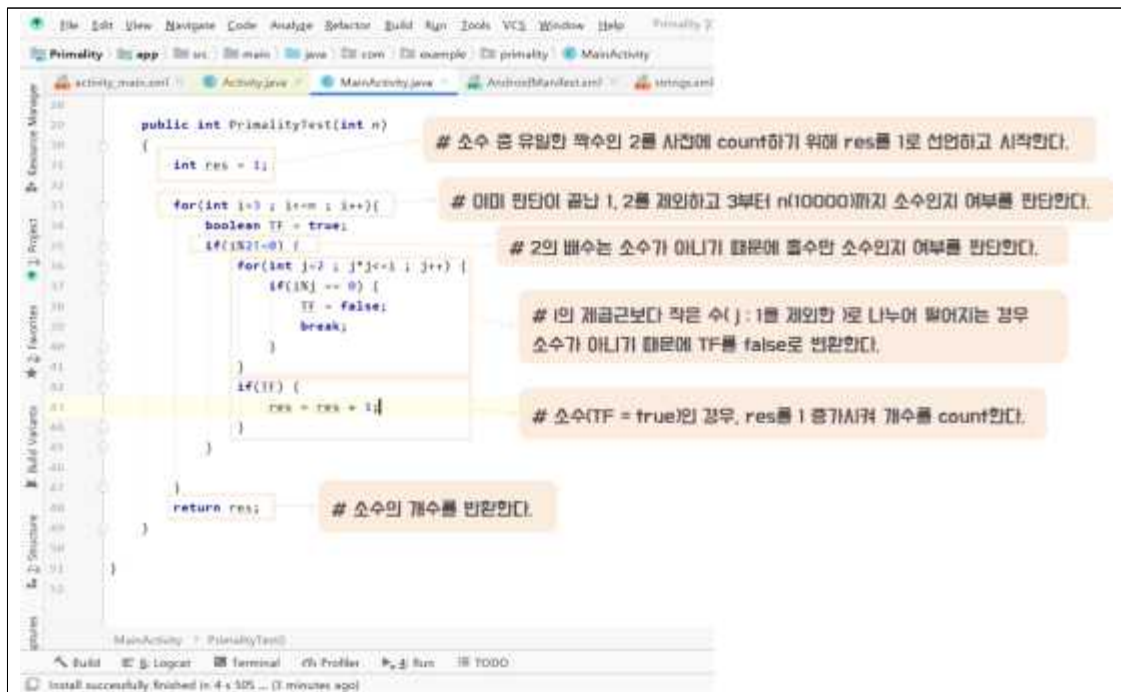
public int PrimalityTest(int n)
{
    int res = 1;
    for(int i=3 ; i<=n ; i++){
        boolean TF = true;
        if(i%2!=0) {
            for(int j=2 ; j*j<=i ; j++) {
                if(i%j == 0) {
                    TF = false;
                    break;
                }
            }
        }
        if(TF) {
            res = res + 1;
        }
    }
    return res;
}

```



1. text1, text2, text3에 값을 넣어주는 작업을 진행한다.

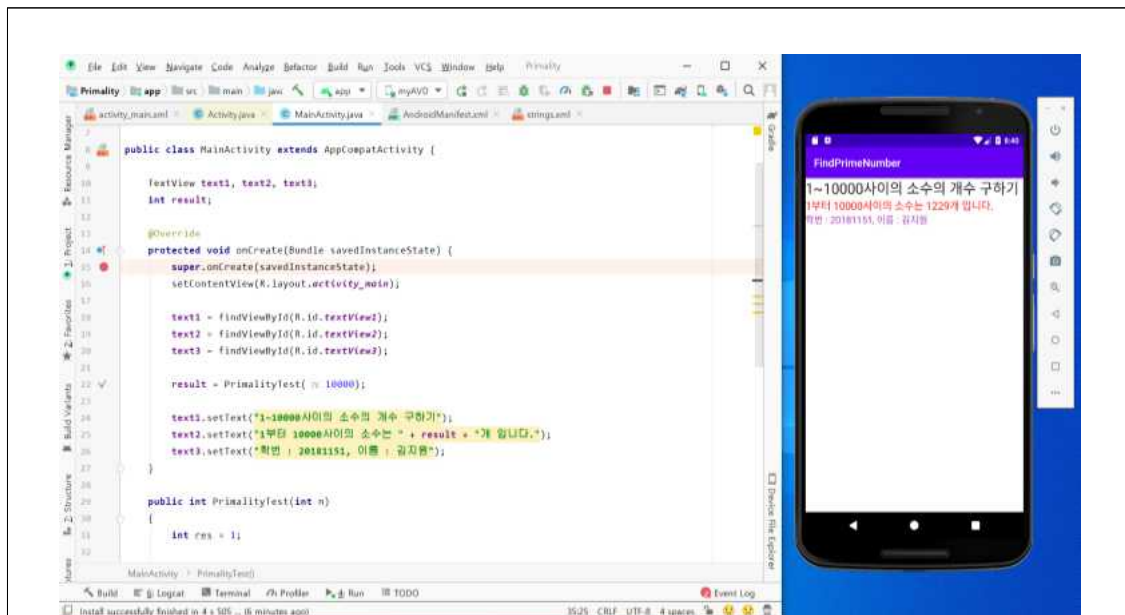
필드멤버를 선언한 후, findViewById()를 통해 값을 넣어주고 text2에 있는 result값을 설정하기 위해 PrimalityTest()를 호출한다.



2. PrimalityTest( )로 소수인지를 판단하여 개수를 count하는 함수를 호출한다.

소수 중 유일한 짝수인 2를 count하여 res를 1로 시작한다. 짝수는 2를 약수로 포함하고 있어 소수가 아니므로 i를 3이상, n이하의 값으로 제한한다. i의 제곱근보다 작은 수로 나누어떨어지는 경우 TF를 false로 변환하고 TF가 true일 경우에만 res의 값을 증가시킨다. 총 res를 도출한 뒤 res 값을 반환한다.

### 3. 디버깅



1. 브레이크포인트를 설정한다. ( Ctrl + F8 )

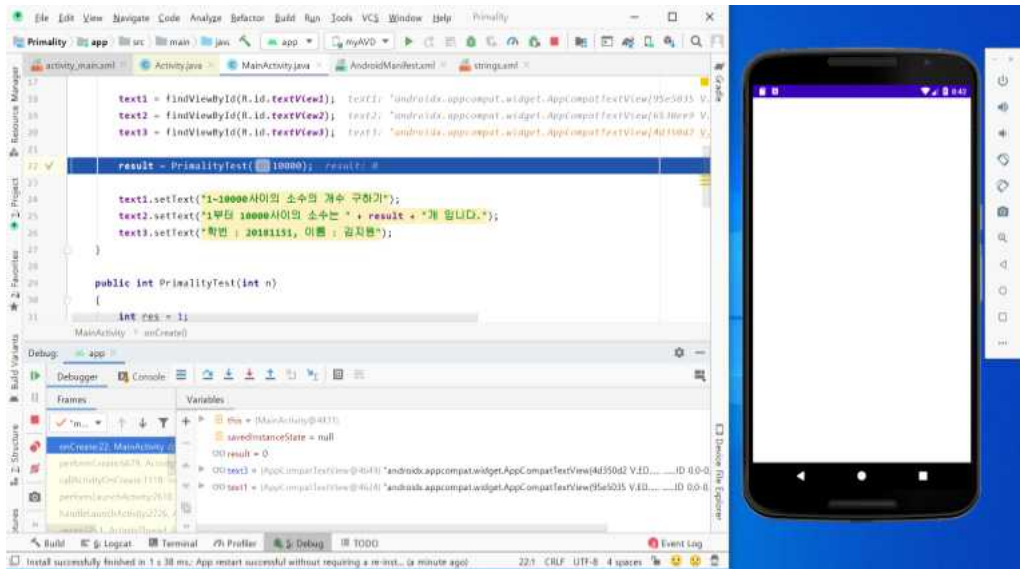
Activity가 실행될 때 'onCreate( )'가 가장 먼저 실행되기 때문에 브레이크포인트로 설정한다.

```
public class MainActivity extends AppCompatActivity {

    TextView text1, text2, text3; text1: null text2: null text3: null
    int result; result: 0

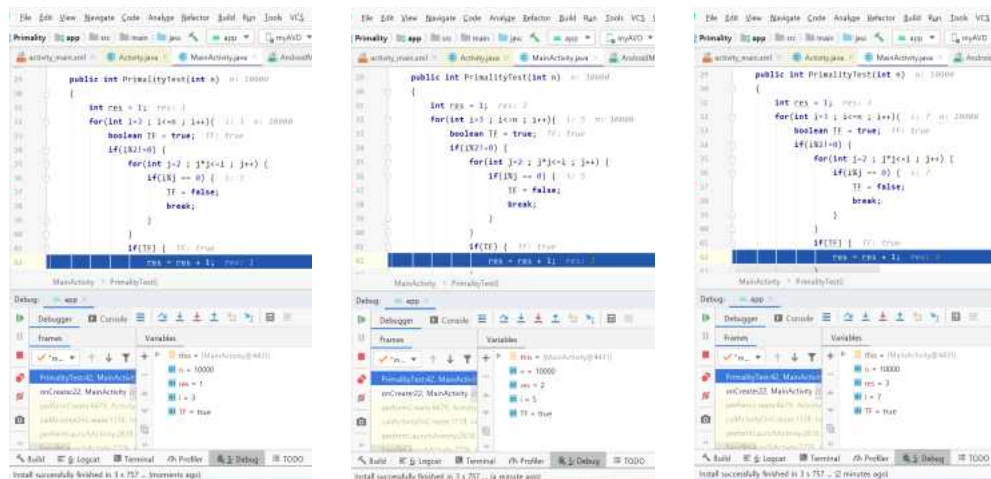

```

↓ ( F8 로 Step Over)



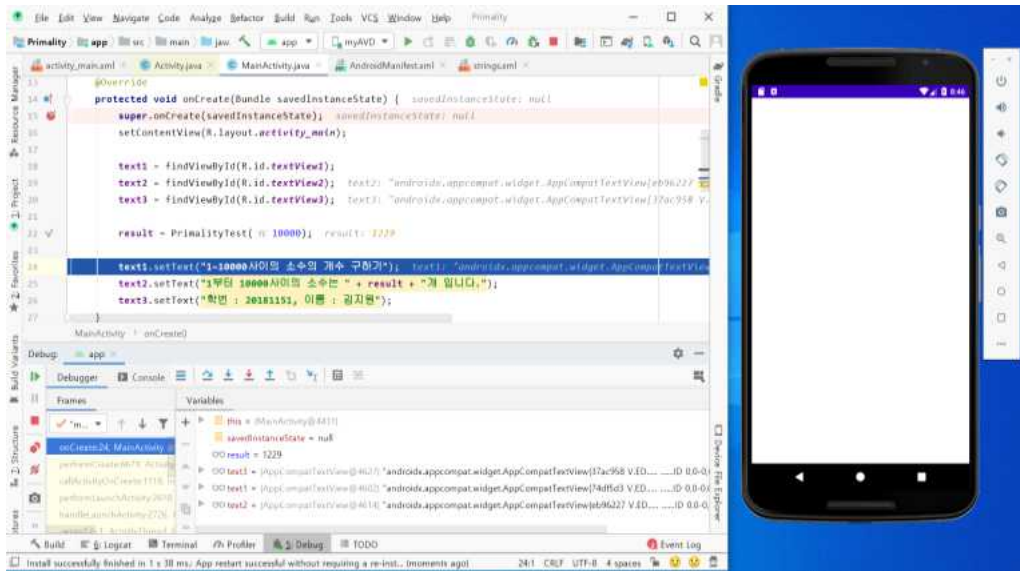
2. 브레이크포인트부터 디버깅을 실행한 후 ( Shift + F9 ), 한 줄씩 옮겨가며 변화를 확인한다.( F8 )

비어있던 text1, text2, text3가 findViewById( )를 사용하면서 값이 생긴다.  
result는 초기 값이 0으로 설정되어 있고, PrimalityTest( )를 사용하기 전은 result가 0이다.

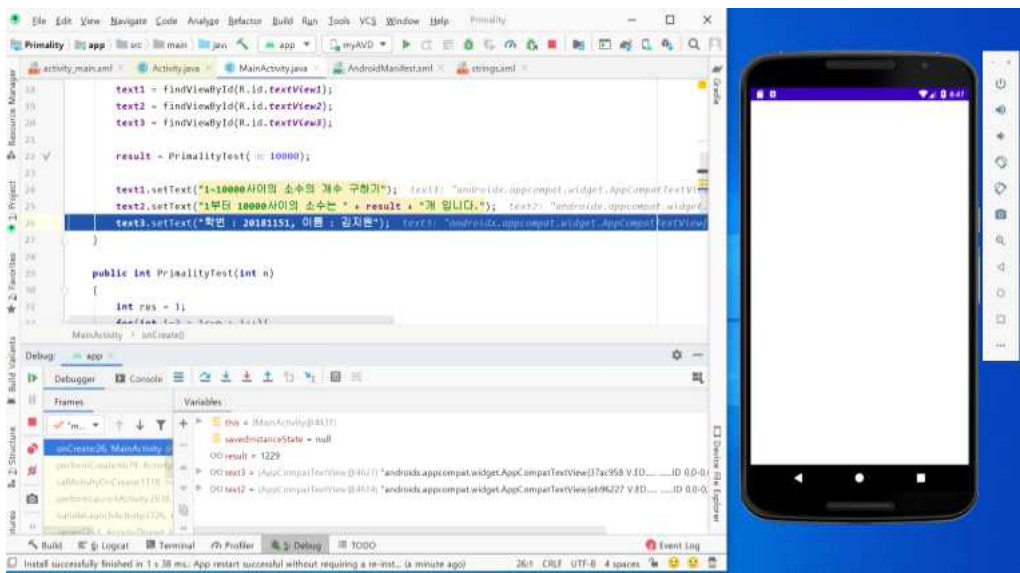


3. PrimalityTest( ) 함수로 Step into하여 변화를 확인한다.( F7 )

TF가 true일 경우 i가 소수를 의미하며 res의 값이 1씩 증가한다.



4. 이전에 실행하던 Step으로 돌아가( Shift + F8 ) 한 줄씩 디버깅한다.  
1부터 10000까지의 소수의 개수인 1229가 result에 입력된다.



5. text1, text2, text3 값 중 text2의 result 값인 1229가 입력된다.

#### 4. AVD( Android Vitual Device ) 결과 화면

