

# 4. 실습 보고서  
- 완전수(Perfect number) 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	2020.04.11

- 목차 -

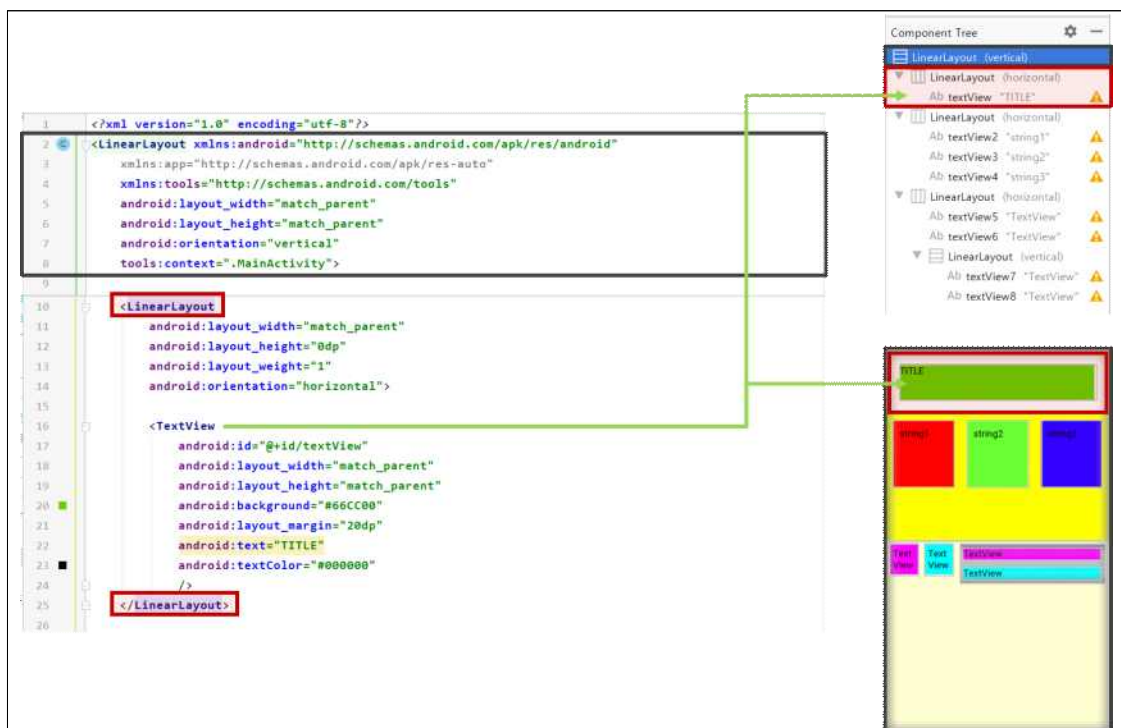
1. 개요	-----	3p
2. View와 Layout ( activity_main.xml )	-----	3p
3. 소스 코드 ( MainActivity.java )	-----	7p
4. AVD ( Android Virtual Device ) 결과 화면	-----	9p

## 1. 개요

완전수(Perfect number)는 자기 자신을 제외한 양의 약수를 더했을 때 자기 자신이 되는 양의 정수이다. 이번 실습에선 1-999 사이의 완전수를 구하는 안드로이드 프로그램을 View와 Layout을 제약조건에 맞게 설정하여 프로그램을 제작한다.

또한, onTouchEvent 메소드를 사용하여 Button과 같은 효과를 내는 TextView를 추가하여 이를 통해 완전수의 출력을 제어한다.

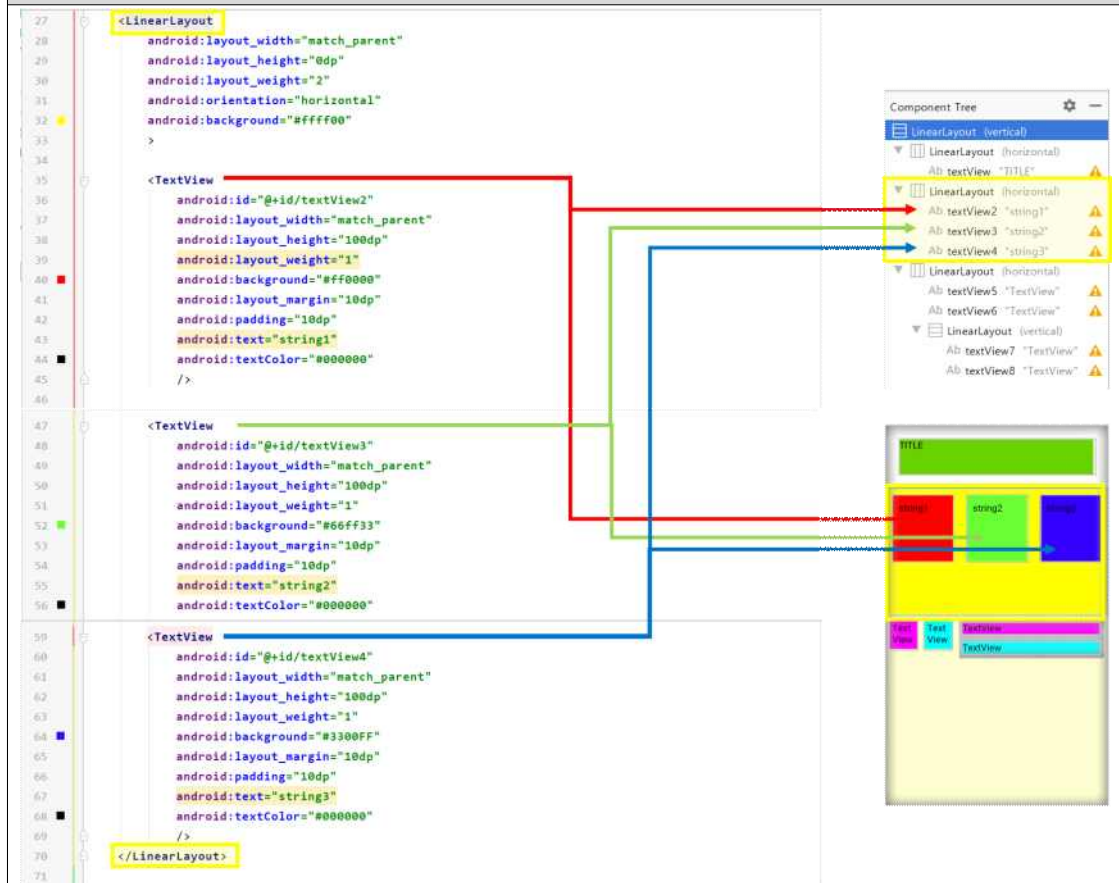
## 2. View와 Layout ( activity\_main.xml )



Linear Layout							
Linear Layout	Linear Layout			Linear Layout			
TextView	TextView	TextView	TextView	TextView	TextView	Linear Layout	
						TextView	TextView

- 화면 전체를 Linear Layout으로 설정
  - orientation : vertical
- 전체 Layout의 첫 번째 Linear Layout 추가 (제약조건 : Linear Layout vertical 안에 Horizontal 3개)
  - orientation : horizontal
  - layout\_weight : 1 (제약조건 : Linear Layout A:B:C 비율은 1:2:3)
- Linear Layout에 TextView 추가
  - id : textView

- layout\_margin : 20dp (제약조건 : A구역 TextView 마진 : 20)
- text : "TITLE"



Linear Layout							
Linear Layout	Linear Layout			Linear Layout			
TextView	TextView	TextView	TextView	TextView	TextView	Linear Layout	
						TextView	TextView

- 전체 Layout의 두 번째 Linear Layout 추가
  - orientation : horizontal
  - layout\_weight : 2 (제약조건 : Linear Layout A:B:C 비율은 1:2:3)
- Linear Layout에 첫 번째 TextView 추가
  - id : textView2
  - layout\_height : 100dp (제약조건 : B구역 TextView 세로길이 : 100dp)
  - layout\_margin : 10dp (제약조건 : B구역 TextView 마진 : 10)
  - layout\_padding : 10dp (제약조건 : B구역 TextView 패딩 : 10)
- Linear Layout에 두 번째 TextView 추가
  - id : textView3
- Linear Layout에 세 번째 TextView 추가
  - id : textView4

```

72 <LinearLayout
73     android:layout_width="match_parent"
74     android:layout_height="0dp"
75     android:layout_weight="3"
76     android:orientation="horizontal"
77     android:background="#ffffff"
78 >
79
80 <TextView
81     android:id="@+id/textView5"
82     android:layout_width="wrap_content"
83     android:layout_height="wrap_content"
84     android:layout_weight="1"
85     android:background="#ff00ff"
86     android:layout_margin="5dp"
87     android:padding="5dp"
88     android:text="TextView"
89     android:textColor="#000000"
90 />
91
92 <TextView
93     android:id="@+id/textView6"
94     android:layout_width="wrap_content"
95     android:layout_height="wrap_content"
96     android:layout_weight="1"
97     android:background="#00ffff"
98     android:layout_margin="5dp"
99     android:padding="5dp"
100     android:text="TextView"
101     android:textColor="#000000"
102 />
103
104 <LinearLayout
105     android:layout_width="match_parent"
106     android:layout_height="wrap_content"
107     android:layout_weight="4"
108     android:orientation="vertical"
109     android:background="#000000"
110     android:layout_margin="5dp"
111 >
112
113 <TextView
114     android:id="@+id/textView7"
115     android:layout_width="match_parent"
116     android:layout_height="wrap_content"
117     android:background="#ff00ff"
118     android:layout_margin="5dp"
119     android:text="TextView"
120     android:textColor="#000000"
121 />
122
123 <TextView
124     android:id="@+id/textView8"
125     android:layout_width="match_parent"
126     android:layout_height="wrap_content"
127     android:background="#00ffff"
128     android:layout_margin="5dp"
129     android:text="TextView"
130     android:textColor="#000000"
131 />
132 </LinearLayout>
133 </LinearLayout>
134
135 </LinearLayout>

```

Component Tree

- LinearLayout (vertical)
  - LinearLayout (horizontal)
    - Ab textView "TITLE"
  - LinearLayout (horizontal)
    - Ab textView2 "string1"
    - Ab textView3 "string2"
    - Ab textView4 "string3"
  - LinearLayout (horizontal)
    - Ab textView5 "TextView"
    - Ab textView6 "TextView"
  - LinearLayout (vertical)
    - Ab textView7 "TextView"
    - Ab textView8 "TextView"

Linear Layout							
Linear Layout	Linear Layout			Linear Layout			
TextView	TextView	TextView	TextView	TextView	TextView	Linear Layout	
						TextView	TextView

- 전체 Layout의 세 번째 Linear Layout 추가
  - orientation : horizontal
  - layout\_weight : 3 (제약조건 : Linear Layout A:B:C 비율은 1:2:3)
- Linear Layout에 첫 번째 TextView 추가
  - id : textView5

- layout\_weight : 1 (제약조건 : C구역 Width 비율은 1:1:4)
- layout\_margin : 5dp (제약조건 : C구역 TextView 마진 : 5)
- layout\_padding : 5dp (제약조건 : C구역 TextView 패딩 : 5)

### 3. Linear Layout에 두 번째 TextView 추가

- id : textView6
- layout\_weight : 1 (제약조건 : C구역 Width 비율은 1:1:4)

### 4. Linear Layout에 Linear Layout 추가

- orientation : vertical
- layout\_weight : 4 (제약조건 : C구역 Width 비율은 1:1:4)

### 5. Linear Layout에 첫 번째 TextView 추가

- id : textView7
- layout\_margin : 5dp (제약조건 : C-1구역 TextView 마진 : 5)



```

1 <resources>
2   <string name="app_name">week4_0409_assignment</string>
3   <string name="title">1~999 사이의 완전수 구하기 문제</string>
4   <string name="profile_id">20181151</string>
5   <string name="profile_name">김지원</string>
6 </resources>
  
```

res > strings.xml에서 Text의 내용을 지정하여 MainActivity 소스 코드에서 사용한다.

### 3. 소스 코드 ( MainActivity.java )

```
activity_main.xml x MainActivity.java x strings.xml x
1 package com.example.week4_0409_assignment;
2
3 import androidx.appcompat.app.AppCompatActivity;
4
5 import android.os.Bundle;
6 import android.view.MotionEvent;
7 import android.widget.TextView;
8 import android.widget.Toast;
9
10 public class MainActivity extends AppCompatActivity {
11
12     TextView text1;
13     TextView text2;
14     TextView text3;
15     TextView text4;
16     TextView text5;
17     TextView text6;
18     TextView text7;
19     TextView text8;
20
21     @Override
22     protected void onCreate(Bundle savedInstanceState) {
23         super.onCreate(savedInstanceState);
24         setContentView(R.layout.activity_main);
25
26         text1 = findViewById(R.id.textView);
27         text2 = findViewById(R.id.textView2);
28         text3 = findViewById(R.id.textView3);
29         text4 = findViewById(R.id.textView4);
30         text5 = findViewById(R.id.textView5);
31         text6 = findViewById(R.id.textView6);
32         text7 = findViewById(R.id.textView7);
33         text8 = findViewById(R.id.textView8);
34
35         text1.setText(getString(R.string.title));
36         text2.setText("");
37         text3.setText("");
38         text4.setText("");
39
40         text7.setText(getString(R.string.profile_id));
41         text8.setText(getString(R.string.profile_name));
42     }
43 }
```

1. onCreate( )에서 비어있는 text들에 findViewById( )를 사용해서 값을 넣어준다.
  - text1은 "1~999 사이의 완전수 구하기 문제"를 출력한다. ( res > strings.xml의 설정 값으로 setText 한다. )
  - text2~text4는 처음에 비어있는 TextView로 출력한다. ( setText("") )
  - text7, text8은 학번과 이름을 출력한다. ( res > strings.xml의 설정 값으로 setText 한다. )

```
44      @Override
45      public boolean onTouchEvent(MotionEvent event) {
46
47          float x = event.getX();
48          float y = event.getY();
49          int act = event.getAction();
50          {
51              if(act == MotionEvent.ACTION_DOWN)
52              {
53                  // Toast.makeText(getApplicationContext(), ""+x+", "+y, Toast.LENGTH_LONG).show();
54                  if(x>=10 && x<=115 && y>=1015 && y<=1135)
55                  {
56                      int order = 1;
57                      for(int n=1 ; n<1000 ; n++)
58                      {
59                          int perfect = PerfectTest(n);
60                          if(perfect != 0){
61                              if(order == 1){
62                                  text2.setText(""+perfect);
63                                  order++;
64                              }
65                              else if(order == 2) {
66                                  text3.setText(""+perfect);
67                                  order++;
68                              }
69                              else if(order == 3) {
70                                  text4.setText(""+perfect);
71                              }
72                          }
73                      }
74                  }
75
76                  if(x>=145 && x<=250 && y>=1015 && y<=1135)
77                  {
78                      text2.setText("");
79                      text3.setText("");
80                      text4.setText("");
81                  }
82              }
83          }
84          return super.onTouchEvent(event);
85      }
86
87  }
```

2. onTouchEvent( )에서 User의 행동에 따른 처리를 정의한다.( Button 유사 event 처리 )

- act는 MotionEvent 중 ACTION\_DOWN(마우스 누름)으로 x, y는 그때의 마우스 포인터의 좌표를 의미한다.
- x좌표 값이 10~115 사이의 값이고 y좌표 값이 1015~1135 사이의 값이면 PerfectTest()를 이용하여 text2, text3, text4에 완전수를 하나씩 출력한다.
- x좌표 값이 145~250 사이의 값이고 y좌표 값이 1015~1135 사이의 값이면 text2, text3, text4 값이 다시 사라진다.
- x좌표 값과 y좌표 값은 Toast로 표시하여 TextView의 위치를 알아낸다.



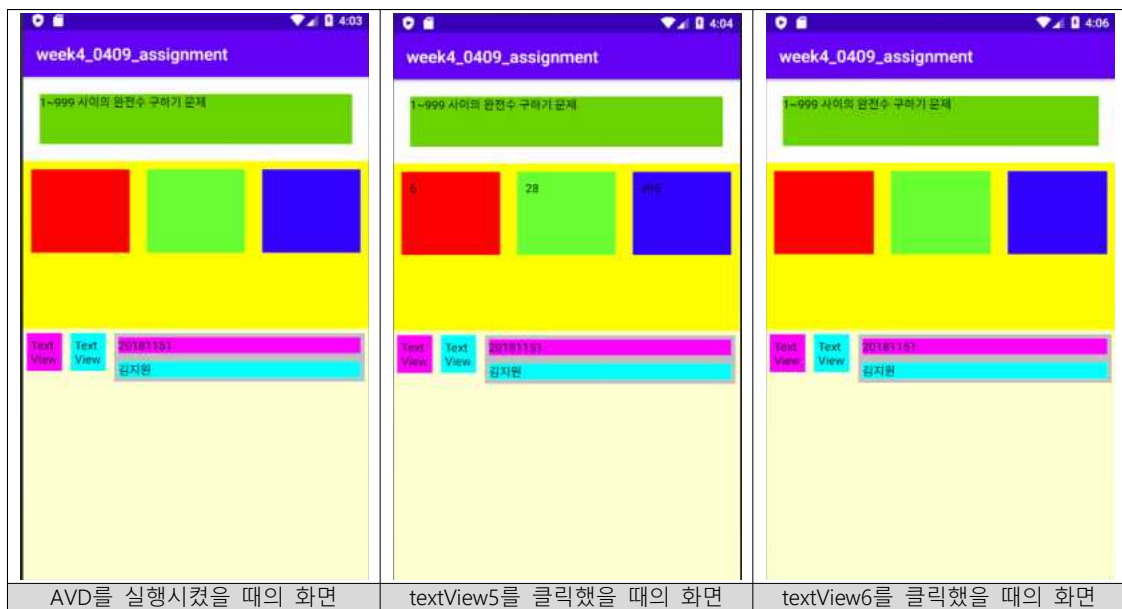
```

88      int PerfectTest(int n)
89      {
90          int total = 0;
91          for(int i=1 ; i<n ; i++){
92              if(n%i==0){
93                  total = total + i;
94              }
95          }
96          if(n == total){
97              return n;
98          }
99          return 0;
100      }
101
102  }

```

3. PerfectTest 함수를 정의하여 자기 자신을 제외한 양의 약수를 더했을 때 자기 자신이 되는 수를 반환하고 그렇지 않으면 0을 반환한다.

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



- 1과 999 사이의 완전수는 [ 6, 28, 496 ]이다.
- textView5를 클릭하면 textView2, textView3, textView4에 완전수가 하나씩 출력된다.
- textView6를 클릭하면 textView2, textView3, textView4에 쓰여있던 완전수가 제거된다.