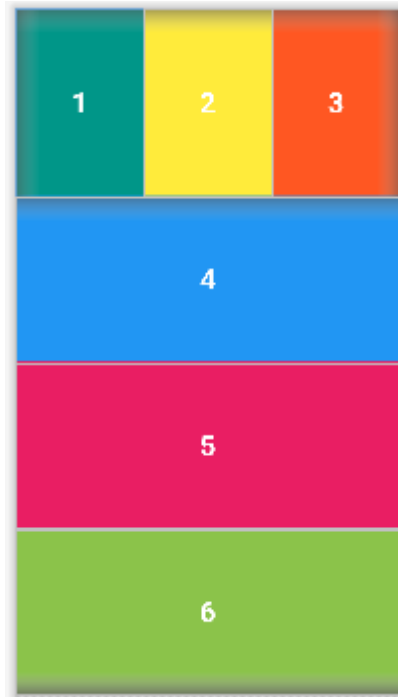


## Lab03

### Bài 1.

#### *Yêu cầu:*

Sử dụng LinearLayout thiết kế giao diện như sau:



#### *Hướng dẫn:*

- ☐ Tạo Project có sử dụng LinearLayout làm layout gốc.
- ☐ Trong LinearLayout gốc tạo 1 LinearLayout thứ 2 có thuộc tính:
  - `android:orientation="vertical"`
- ☐ Trong LinearLayout thứ 2 này tạo 3 TextView lần lượt là 1-2-3 như sau:

```

<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content">

    <TextView
        android:layout_width="0dp"
        android:layout_height="200dp"
        android:layout_weight="1"
        android:background="#009688"
        android:gravity="center"
        android:text="1"
        android:textColor="@color/white"
        android:textSize="30sp"
        android:textStyle="bold"/>

    <TextView...>

    <TextView...>

</LinearLayout>

```

- ❖ **android:layout\_width="0dp"** kết hợp với **android:layout\_weight="1"** là ràng buộc để phân chia không gian

□ Tương tự, tạo LinearLayout thứ 3 có thuộc tính:

- **android:orientation="vertical"**

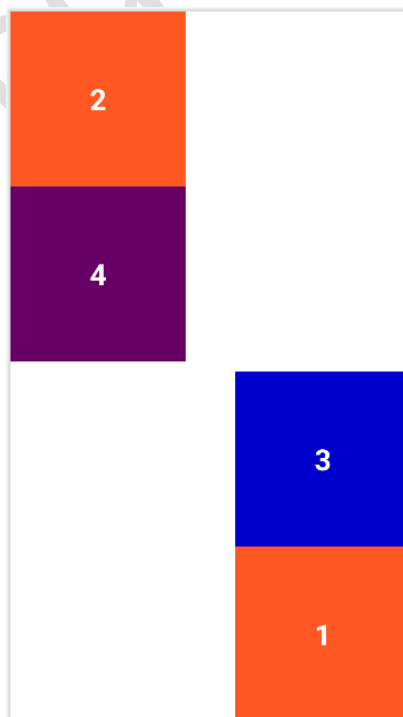
Trong LinearLayout thứ 3 này chứa 3 TextView 4-5-6 như sau:



## Bài 2.

### ***Yêu cầu***

Sử dụng RelativeLayout và thiết kế giao diện như hình sau:

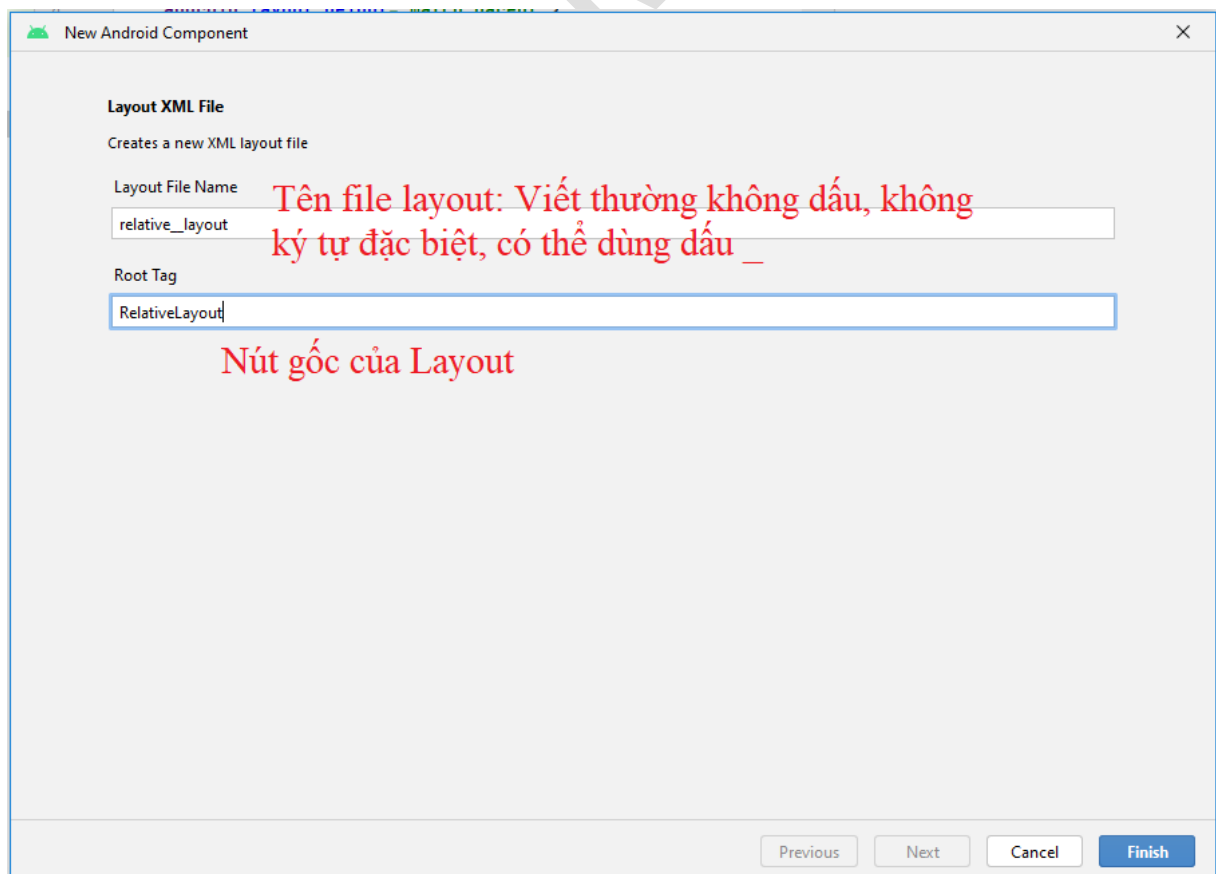
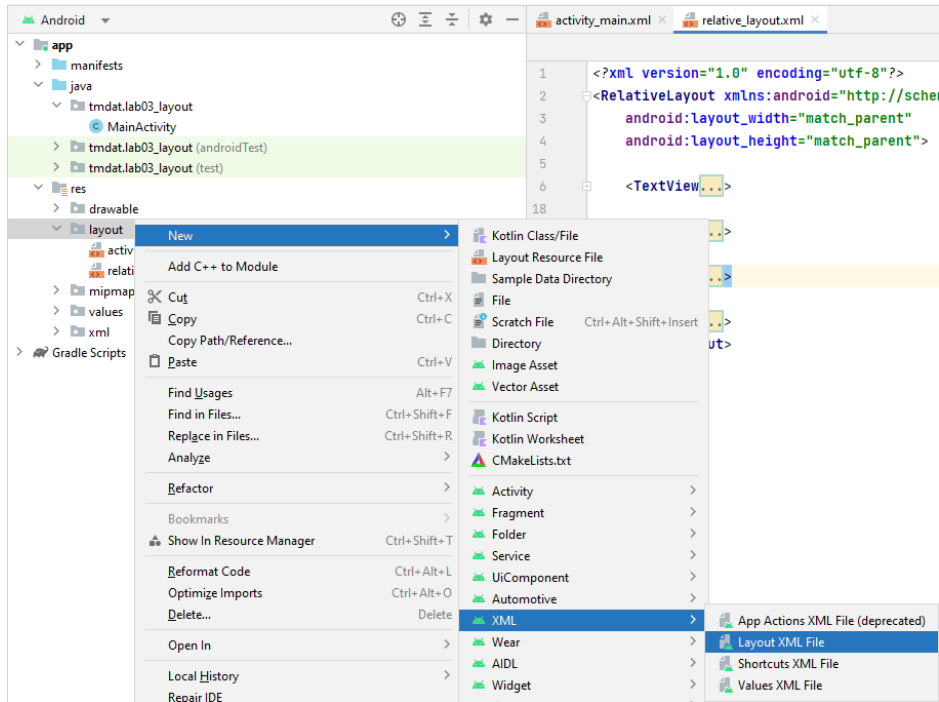


## Hướng dẫn

Tạo Layout: Có thể sửa trên layout cũ hoặc tạo layout mới:

Click chuột phải vào thư mục layout, chọn

New/XML/Layout XML File



Trong nút gốc RelativeLayout, tạo 4 TextView hiển thị các số 1-2-3-4 và sắp xếp bố cục như đề bài.

Gợi ý thuộc tính:

- ❖ **android:layout\_alignParentTop = "true"** – cạnh trên của TextView trùng với cạnh trên của layout chứa nó
- ❖ **android:layout\_alignParentStart="true"** - cạnh bên trái của TextView trùng với cạnh bên trái của layout chứa nó
- ❖ Tương tự với các vị trí khác: **Bottom, End**

Ví dụ:

```
<TextView
    android:id="@+id/textView2"
    android:layout_width="180dp"
    android:layout_height="180dp"
    android:layout_alignParentStart="true"
    android:layout_alignParentTop="true"
    android:background="#ff5722"
    android:gravity="center"
    android:text="2"
    android:textColor="@color/white"
    android:textSize="30sp"
    android:textStyle="bold"/>
```

- ❖ Sử dụng **android:layout\_above = "id widget nằm dưới"** – TextView nằm trên widget có id được thuộc tính trỏ tới.
- ❖ Tương tự **android:layout\_below = "id widget nằm trên"** – TextView nằm dưới widget có id được thuộc tính trỏ tới.

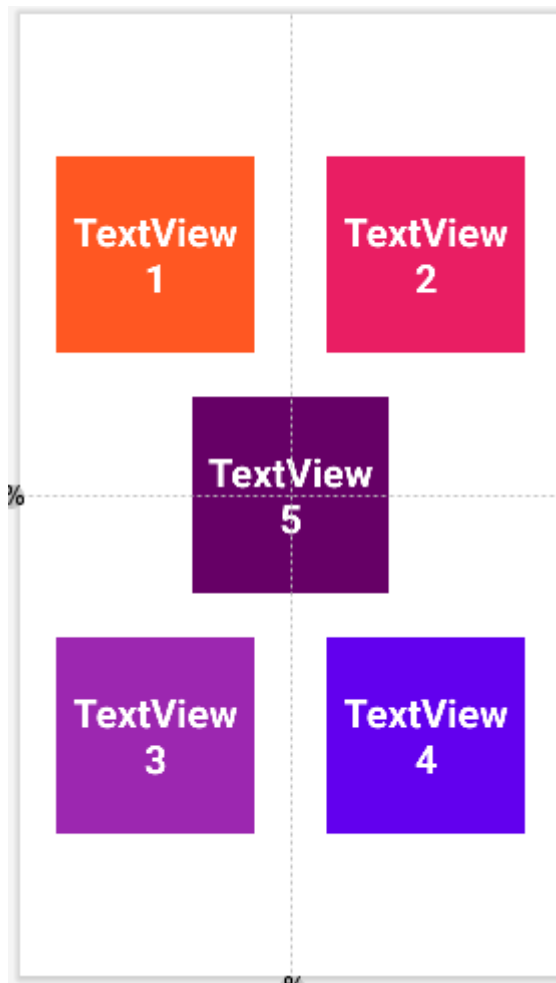
Ví dụ:

```
<TextView
    android:id="@+id/textView3"
    android:layout_width="180dp"
    android:layout_height="180dp"
    android:layout_alignParentEnd="true"
    android:layout_above="@id/textView1"
    android:background="#000c"
    android:gravity="center"
    android:text="3"
    android:textColor="@color/white"
    android:textSize="30sp"
    android:textStyle="bold"/>
```

### Bài 3.

#### ***Yêu cầu:***

Sử dụng ConstraintLayout và thiết kế giao diện như hình dưới đây:



#### ***Hướng dẫn***

Sử dụng layout ConstraintLayout làm layout gốc.

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    android:layout_width="match_parent"
    android:layout_height="match_parent">
```

Trong layout gốc, tạo 2 widget Guideline, đặt id để phân biệt Guideline dọc và ngang

```

<!-- Guideline ngang ở giữa màn hình (50% chiều ngang) -->
<androidx.constraintlayout.widget.Guideline
    android:id="@+id/lineHorizontal"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:orientation="horizontal"
    app:layout_constraintGuide_percent="0.5"/>

<!-- Guideline dọc ở giữa màn hình (50% chiều dọc) -->
<androidx.constraintlayout.widget.Guideline
    android:id="@+id/lineVertical"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:orientation="vertical"
    app:layout_constraintGuide_percent="0.5"/>

```

Trong đoạn mã trên, `app:layout_constraintGuide_percent="0.5"` đặt một hướng dẫn (guide) ở giữa ConstraintLayout. Nó có thể giúp thiết lập các ràng buộc cho các phần tử con dựa trên phần trăm của không gian tổng của ConstraintLayout.

Tiếp theo tạo ra 5 TextView hiển thị các số 1-2-3-4-5, sau đó sắp xếp các TextView như đề bài theo các gợi ý thuộc tính dưới đây:

- `app:layout_constraintTop_toTopOf="parent"` – thuộc tính này dùng để xác định top của TextView có điểm neo với top của layout chứa nó hoặc widget nào đó (cụ thể ở đây là layout chứa TextView)
- `app:layout_constraintBottom_toBottomOf="@+id/lineHorizontal"` – tương tự như trên thuộc tính này để xác định Bottom của TextView có điểm neo trùng với Guideline ngang.

Ví dụ: TextView5

```

<!-- TextView 5: Nằm ở trung tâm của màn hình -->
<TextView
    android:id="@+id/textView5"
    android:layout_width="150dp"
    android:layout_height="150dp"
    android:background="#606"
    android:gravity="center"
    android:text="TextView 5"
    android:textColor="@color/white"
    android:textSize="30sp"
    android:textStyle="bold"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintEnd_toEndOf="parent"/>

```

### TextView1:

```

<!-- TextView 1: Nằm phía trên bên trái của TextView 5, nửa phía trên của lineHorizontal -->
<TextView
    android:id="@+id/textView1"
    android:layout_width="150dp"
    android:layout_height="150dp"
    android:background="#FF5722"
    android:gravity="center"
    android:text="TextView 1"
    android:textColor="@color/white"
    android:textSize="30sp"
    android:textStyle="bold"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintBottom_toTopOf="@+id/lineHorizontal"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintEnd_toEndOf="@+id/lineVertical"/>

```

Đoạn mã này đặt các ràng buộc (constraints) cho TextView1:

- **app:layout\_constraintTop\_toTopOf="parent"**: Đặt cạnh trên của TextView1 sao cho nó trùng với cạnh trên của parent. Điều này đưa TextView1 lên trên cùng của màn hình hoặc phần trên của ConstraintLayout.
- **app:layout\_constraintBottom\_toTopOf="@+id/lineHorizontal"**: Đặt cạnh dưới của TextView1 sao cho nó nằm ngay dưới lineHorizontal (được định danh bằng @+id/lineHorizontal). Điều này tạo ra khoảng trống giữa cạnh dưới của TextView1 và lineHorizontal.



- `app:layout_constraintStart_toStartOf="parent"`: Đặt cạnh bắt đầu của TextView1 sao cho nó trùng với cạnh bắt đầu của parent. Điều này đảm bảo TextView1 nằm ở bên trái của parent.
- `app:layout_constraintEnd_toEndOf="@+id/lineVertical"`: Đặt cạnh kết thúc của TextView1 sao cho nó nằm ngay bên phải của lineVertical (được định danh bằng @+id/lineVertical). Điều này tạo ra khoảng trống giữa cạnh kết thúc của TextView1 và lineVertical.

Tổng cộng, các ràng buộc này định vị TextView1 ở trên cùng của màn hình (do `layout_constraintTop_toTopOf="parent"`), với khoảng trống giữa cạnh dưới của nó và lineHorizontal. Nó cũng đảm bảo rằng TextView1 sẽ nằm ở bên trái và cạnh phải của nó sẽ kết thúc ngay bên cạnh lineVertical, tạo ra sự cân đối theo cả chiều ngang và chiều dọc.

Tương tự với các TextView còn lại.