**TỰ HỌC ANDROID STUDIO**

1. **Tạo project**

* Cửa sổ đầu tiên
  + Application name: Đặt tên ứng dụng của mình
  + Company Domain: Khá quan trọng mặc định sẽ là example sẽ dễ trùng, nó như ID của ứng dụng sẽ dễ bị trùng nên đổi và duy nhất và .com cuối cùng
  + Project location: Nơi lưu file
* Cửa sổ tiếp theo
  + Phiên bản để code (dựa vào thị trường điện thoại)
* Chọn Empty Activity
  + Activity Name: thường sẽ không đổi
  + Layout name: tên file giao diện của nó
* Sau đó ấn finish

1. **Ôn tập lại Java cơ bản**
   1. **Kiểu dữ liệu**

* Int: số nguyên
* Double: số thực
* Boolean: kiểu dữ liệu nguyên thủy (true, false)
* Char: ký tự
* String: chuỗi
  1. **Toán tử so sánh**
* = = so sánh bằng trả về giá trị true hoặc false
* ! = không bằng (khác)
* < so sánh bé
* > so sánh lớn
* < = so sánh bé hơn hoặc bằng
* > = so sánh lớn hơn hoặc bằng
  1. **Hàm IF**
* Dùng để kiểm tra vấn đề gì đó, điều kiện đúng hay không, nếu đúng sẽ thực thi đúng và ngược lại sẽ thực thi sai
* Cú pháp:

if(điều kiện){

//đúng

Khối lệnh

}else{

//sai

Khối lệnh

}

* 1. **Hàm Switch Case**
* Khác if – else, switch case sẽ kiểm tra theo từng trường hợp
* Cú pháp:

Switch(tên biến cần xét){

Case <giá trị 1>:

//khối lệnh

Break; //ngắt chương trình

Case <giá trị 2>:

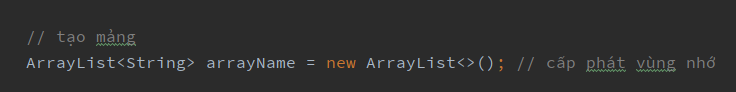
//khối lệnh

Break;

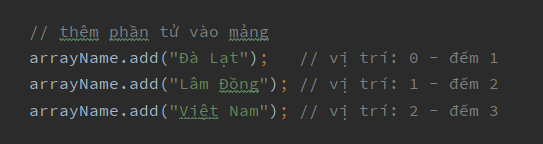
Defautl: <khối lệnh>// nếu kiểm tra tất cả điều kiện trên đều sai

}

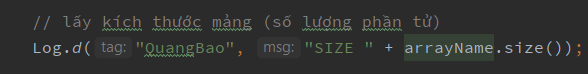
* 1. **Vòng lặp FOR**
* Khi nào sử dụng => nếu kiểm tra vấn đề gì đó cần lặp đi lặp lại nhiều lần đến khi nào đúng và hết dữ liệu thì ngừng
  1. **Mảng – ArrayList**
* Khai báo



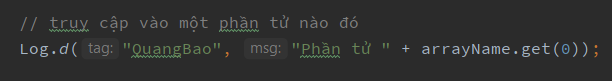
* Thêm



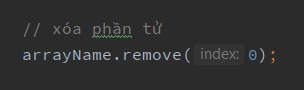
* Lấy kích thước



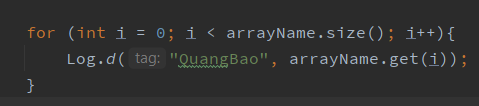
* Truy cập phần tử



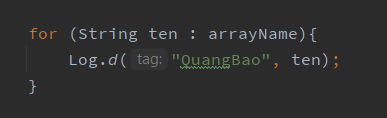
* Xóa phần tử



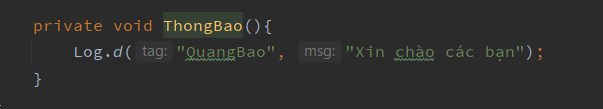
* Duyệt mảng:

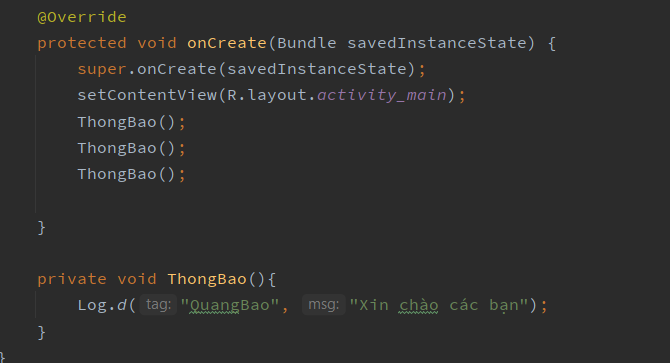


* Hoặc

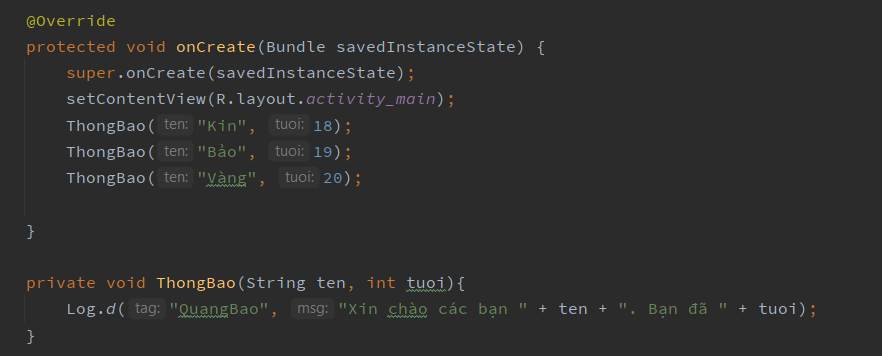


* 1. **Function**
* Đơn giản (không tham số)

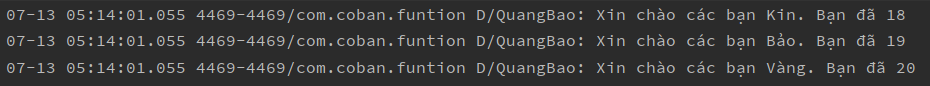




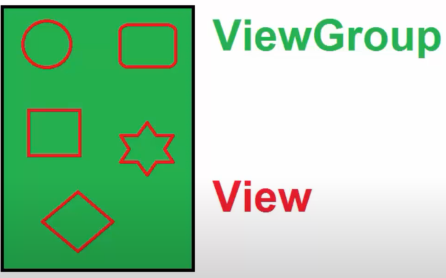
* Function có tham số



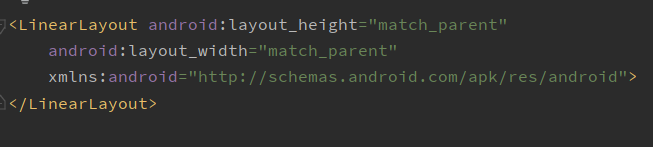
* Kết quả:



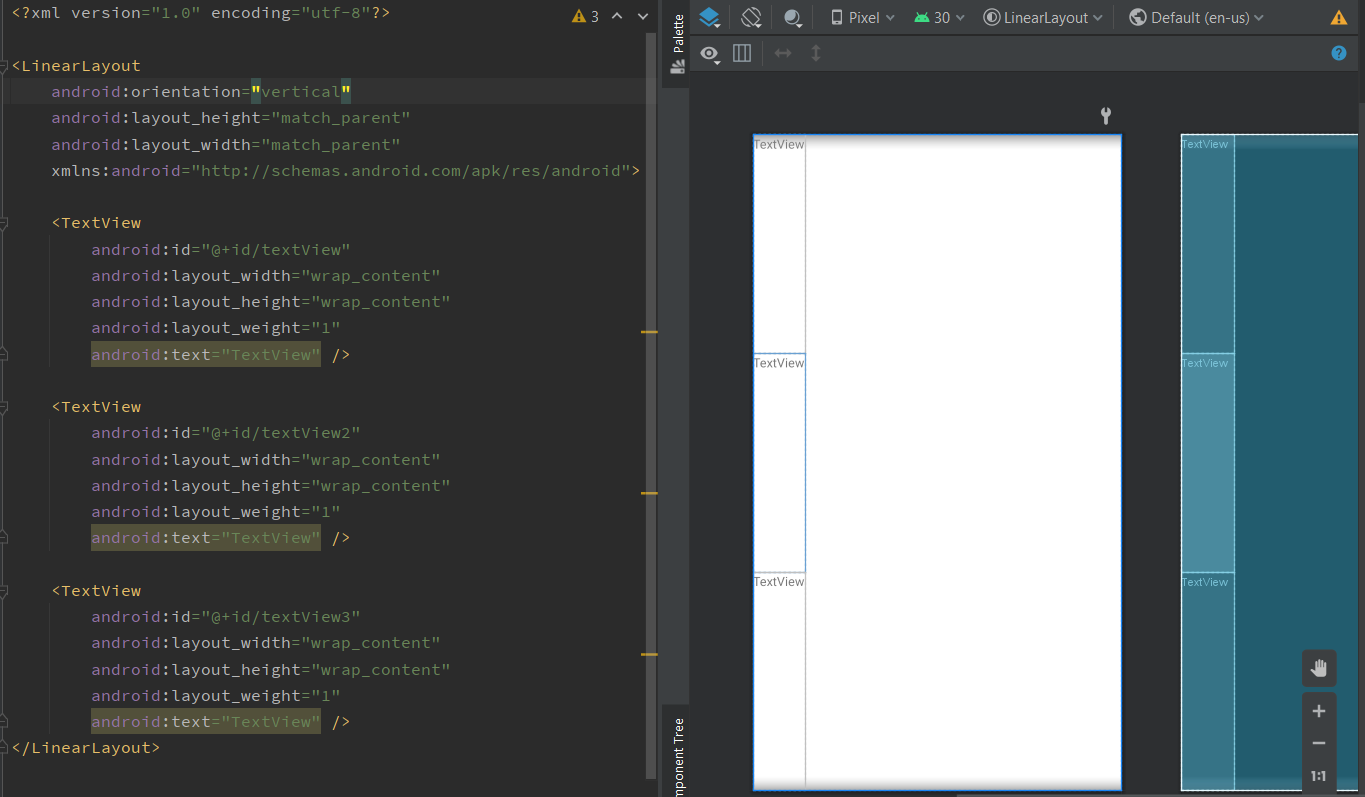
1. **Layout trong android**



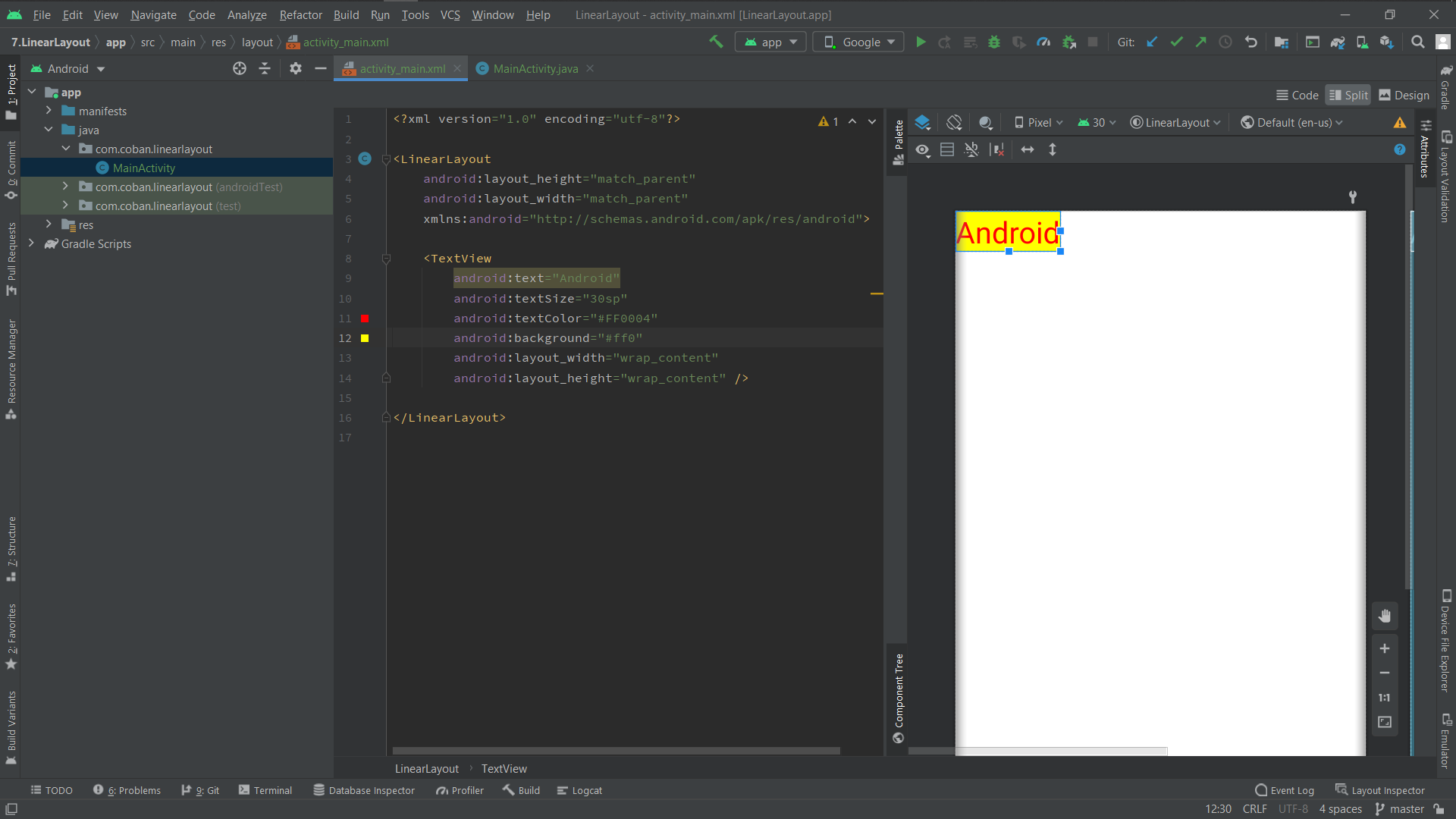
* View là những gì hiển thị được chứa trong ViewGroup
* ViewGroup: các cách sắp xếp, bố trí
  + \* LinearLayout:
* Khởi tạo:

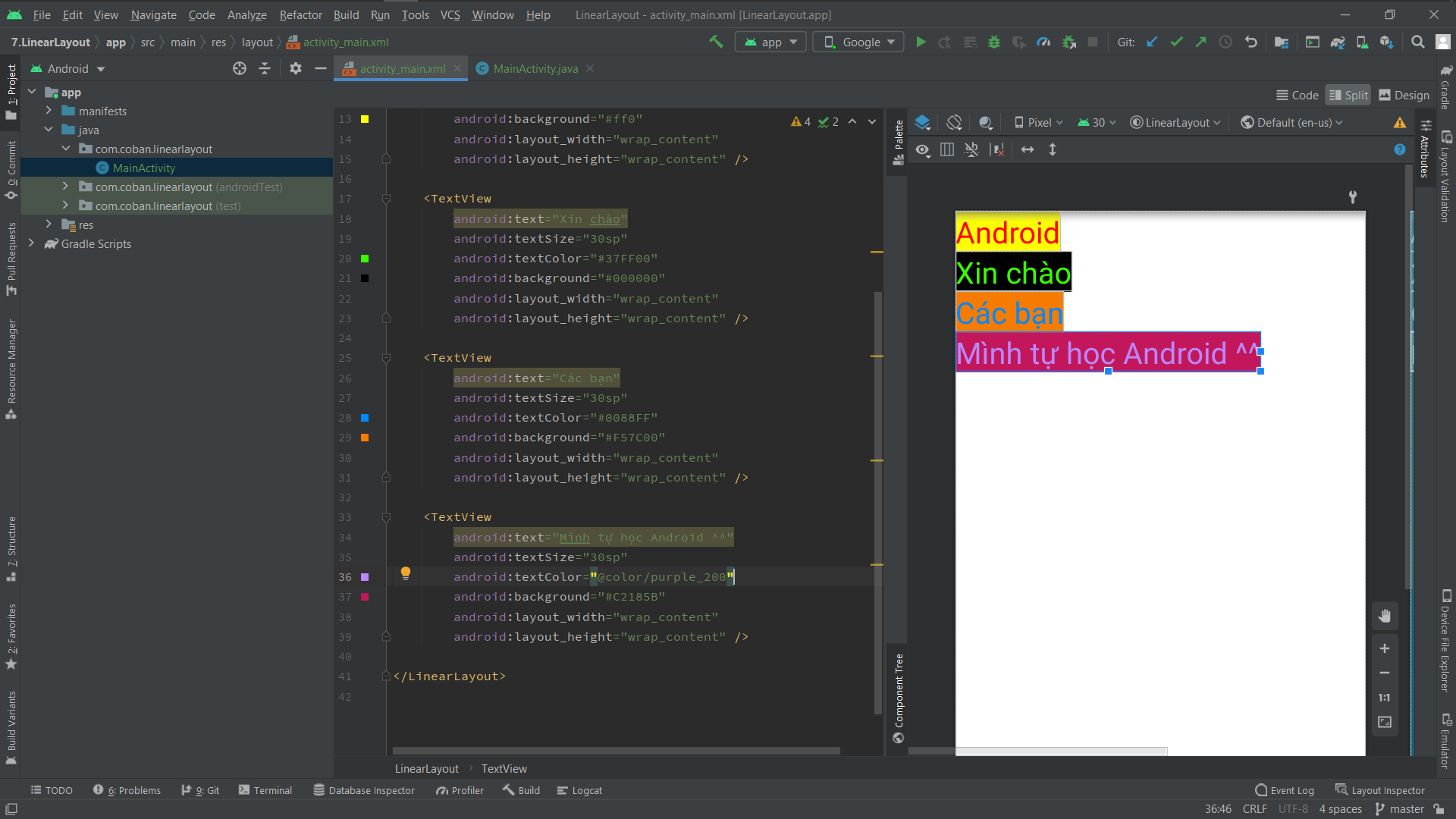


* Kéo thả chỉ sắp xếp theo hàng ngang, khộng cho kéo thả thoải mái được
* Muốn chuyển dọc

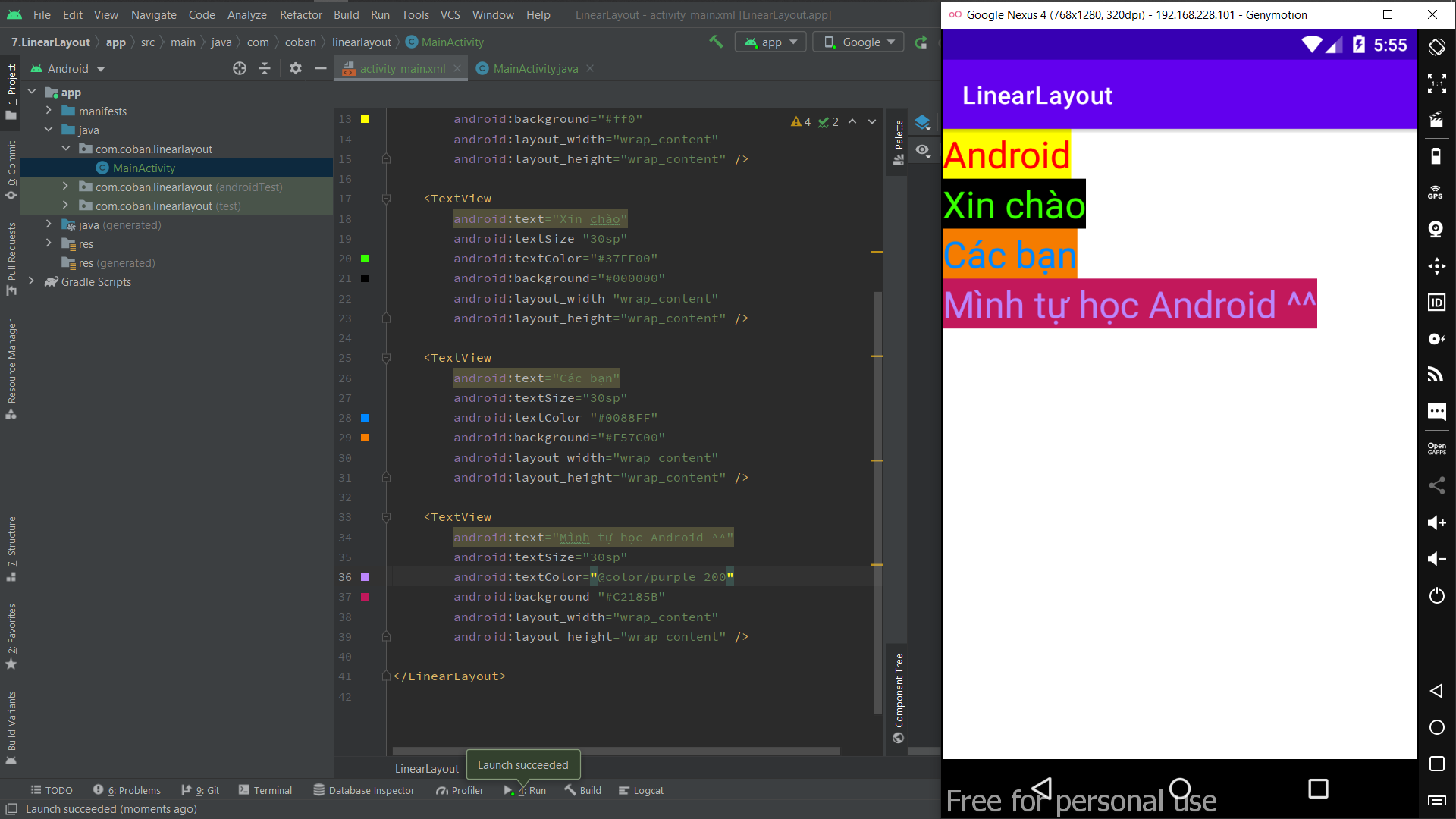


* Chỉ được 1 hướng ngang hoặc dọc
* Match\_parent: có bao nhiêu thì lấy bấy nhiêu của màn hình
* Wrap\_content: lấy vừa đủ nội dung

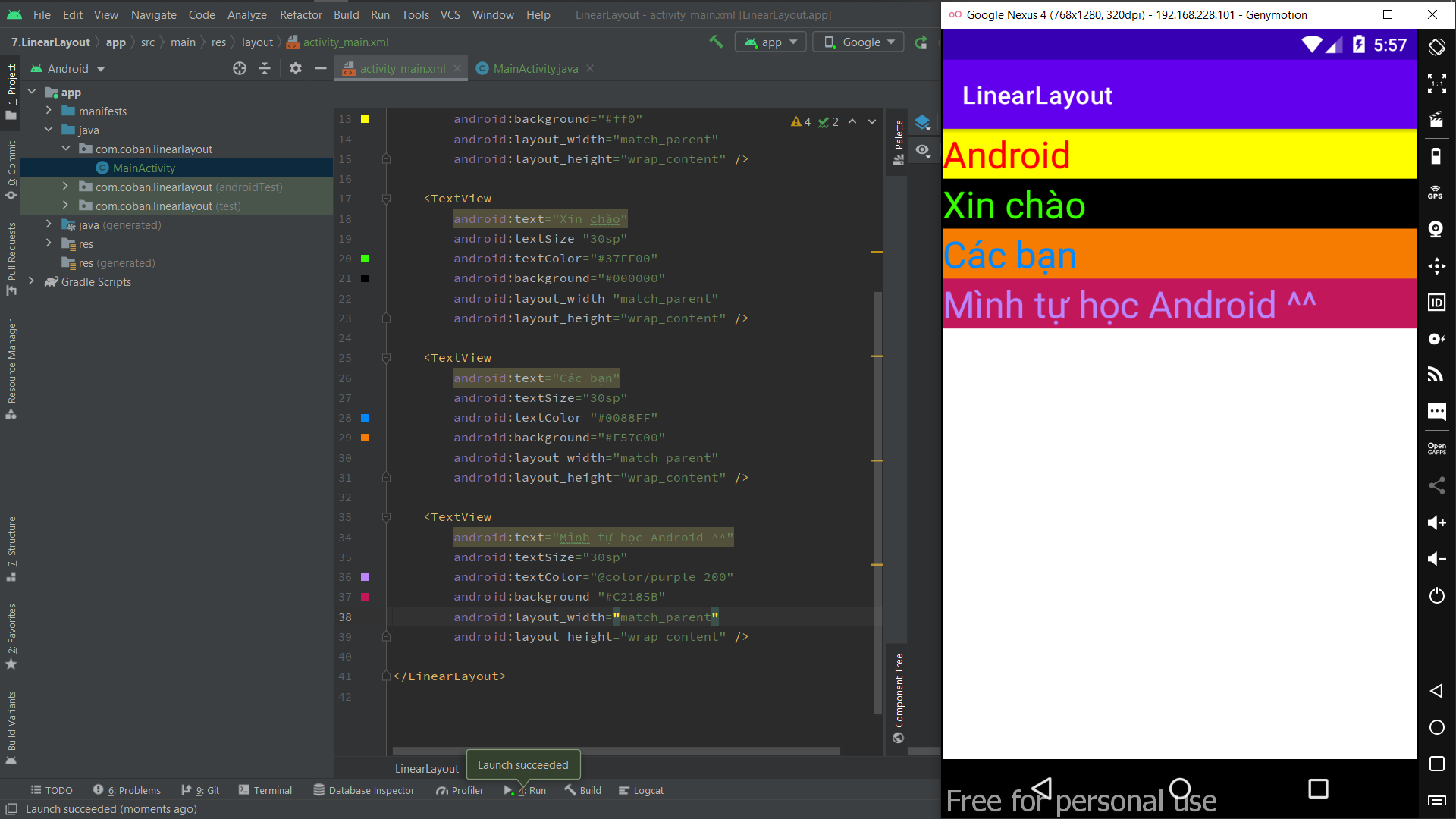




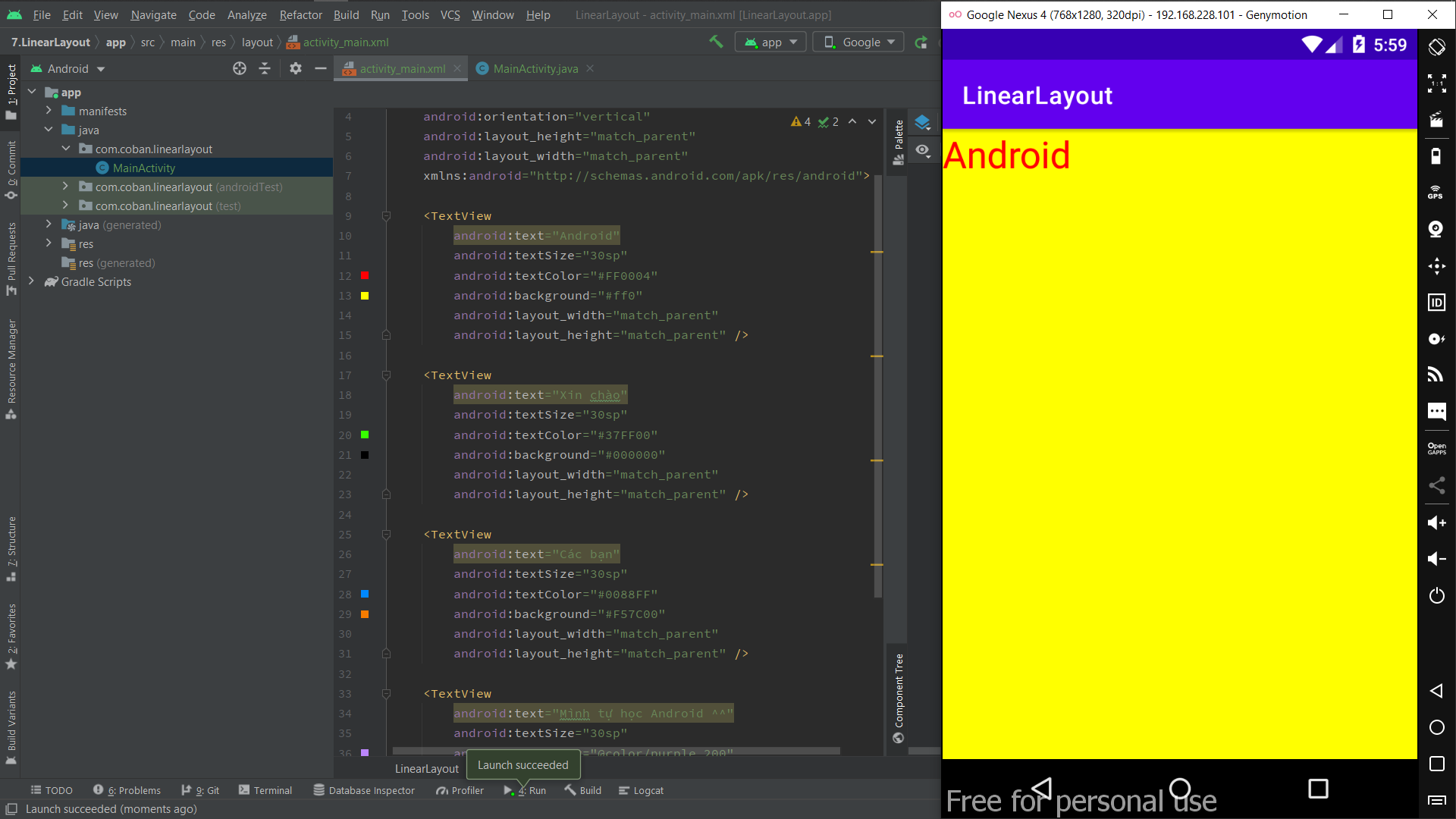
* Kết quả:



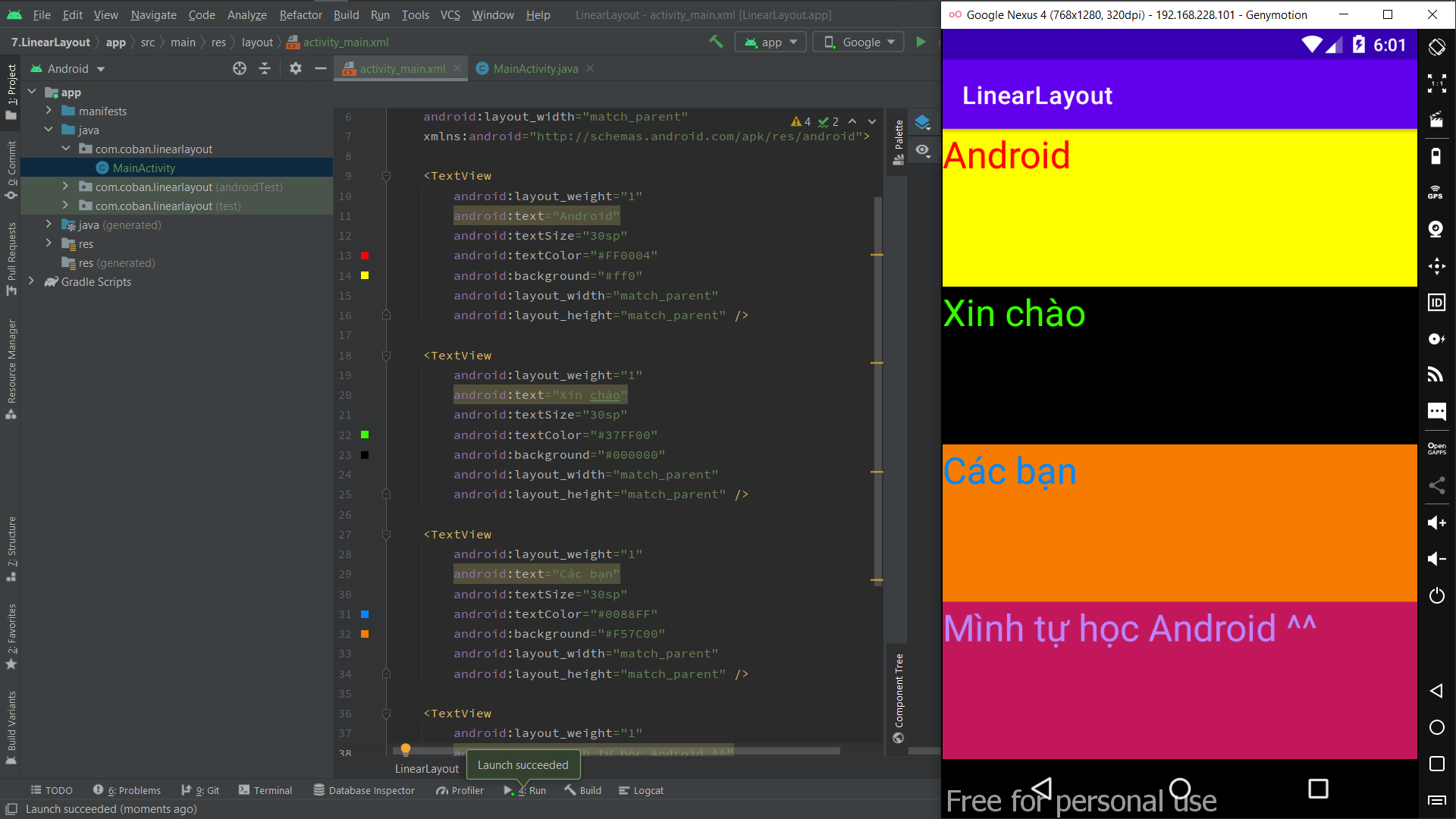
* Bao hết từ 4 cái textview đó:
* Kết quả



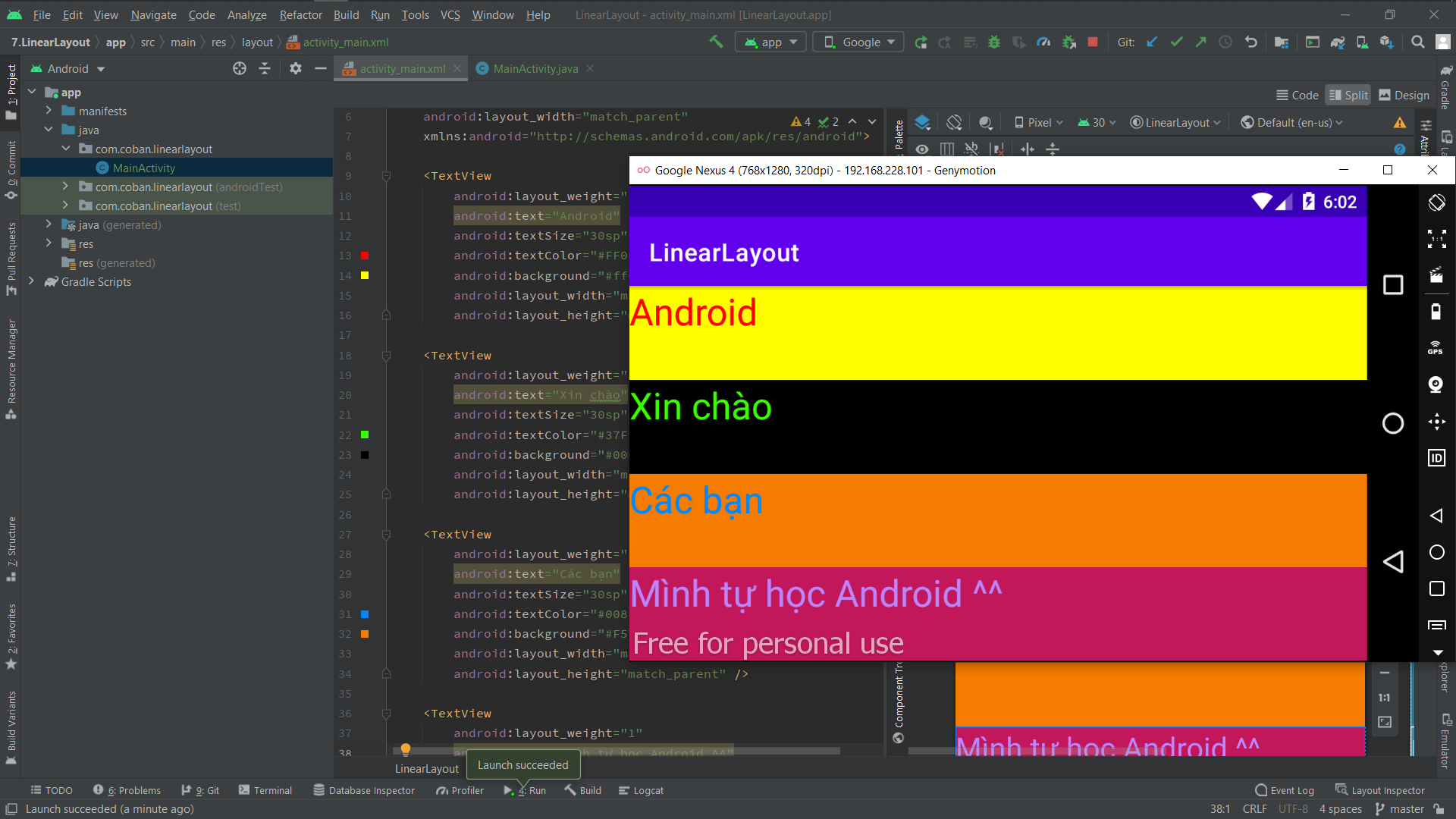
* Trường hợp này vì màu vàng đã lấy hết chiều rộng và cao nên tất cả cái khác sẽ biến mất



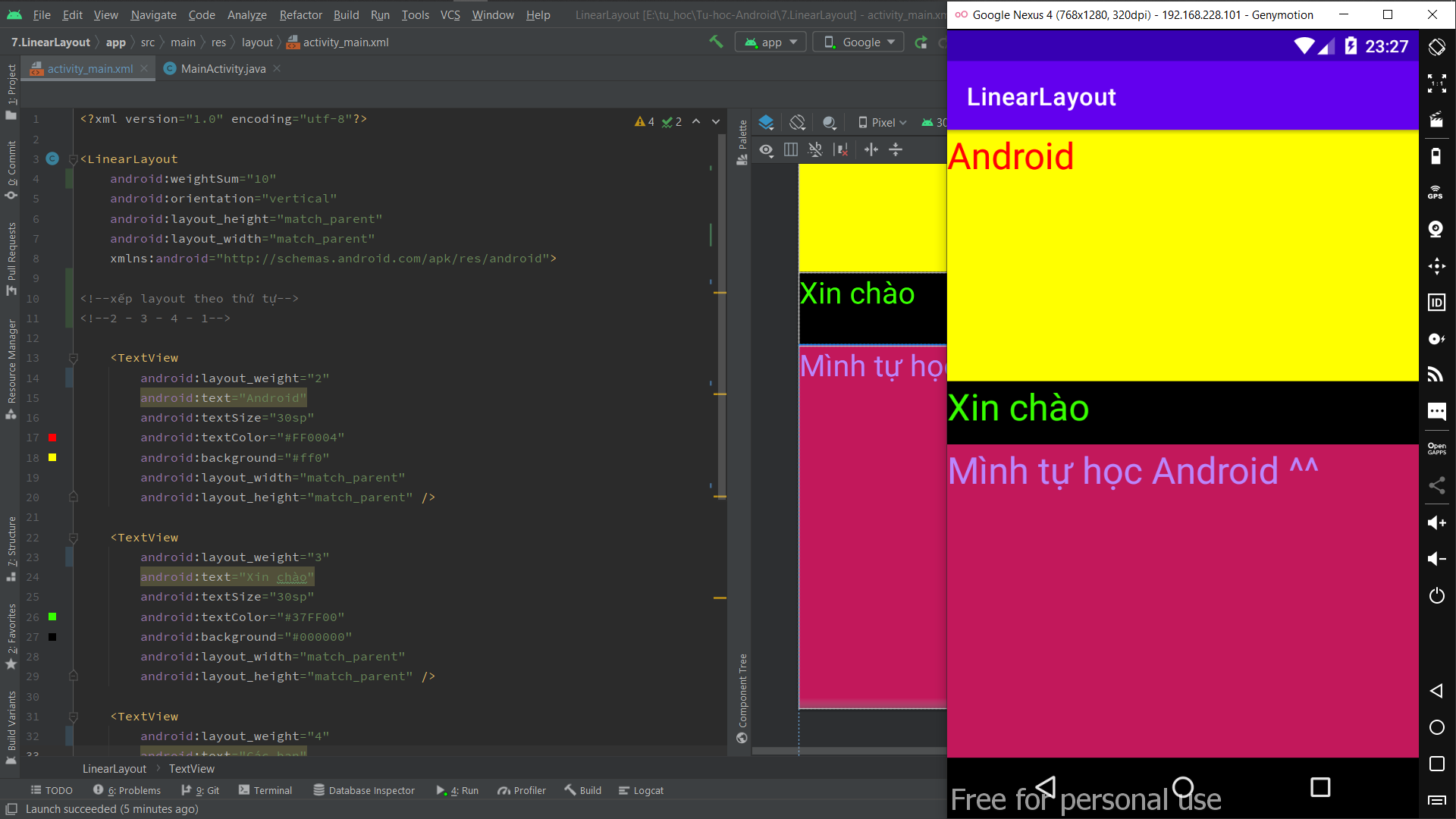
* Trường hợp đã chia đều các vùng của textview:
* Kết quả



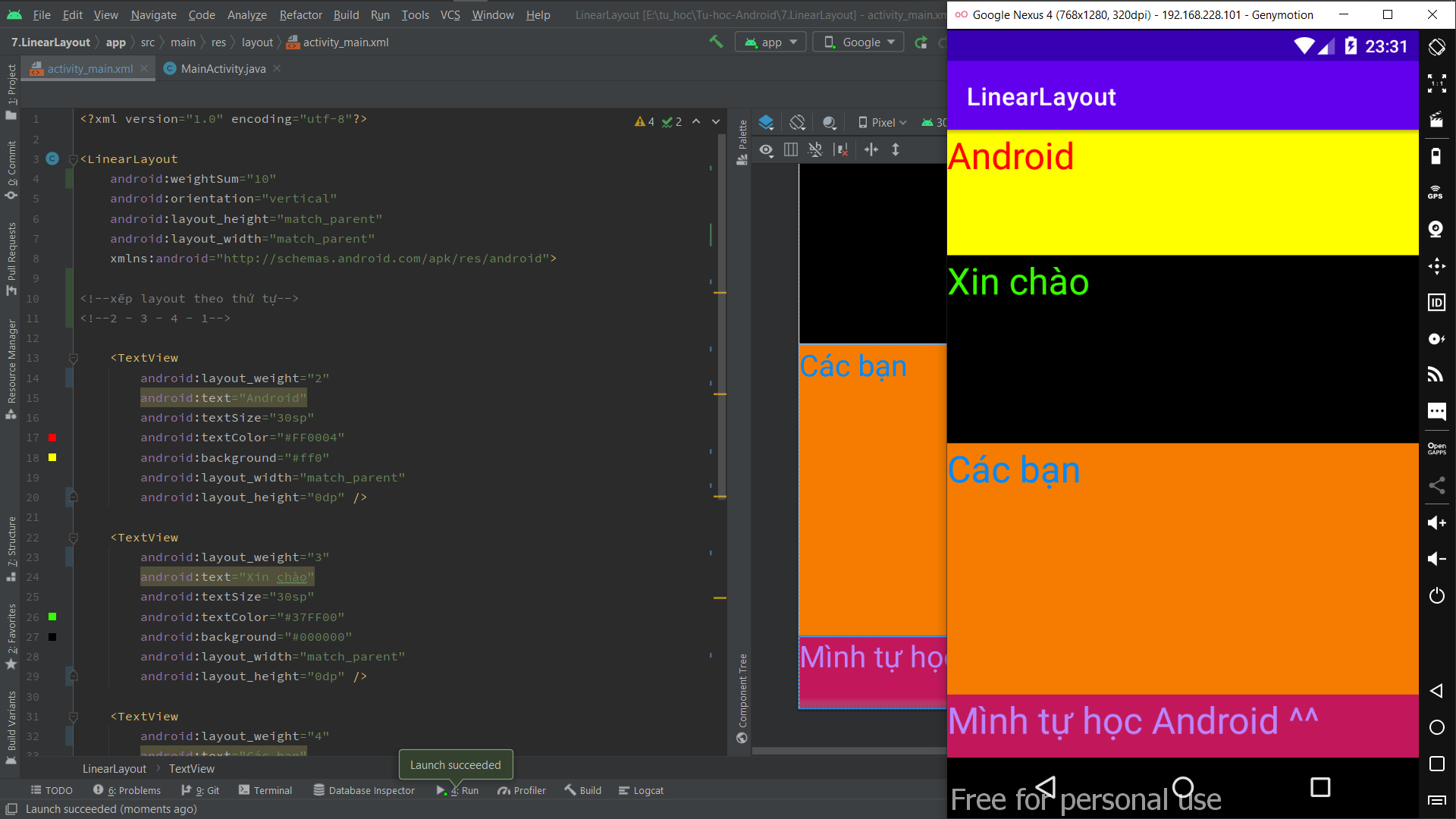
* Và khi xoay điện thoại
* Kết quả:



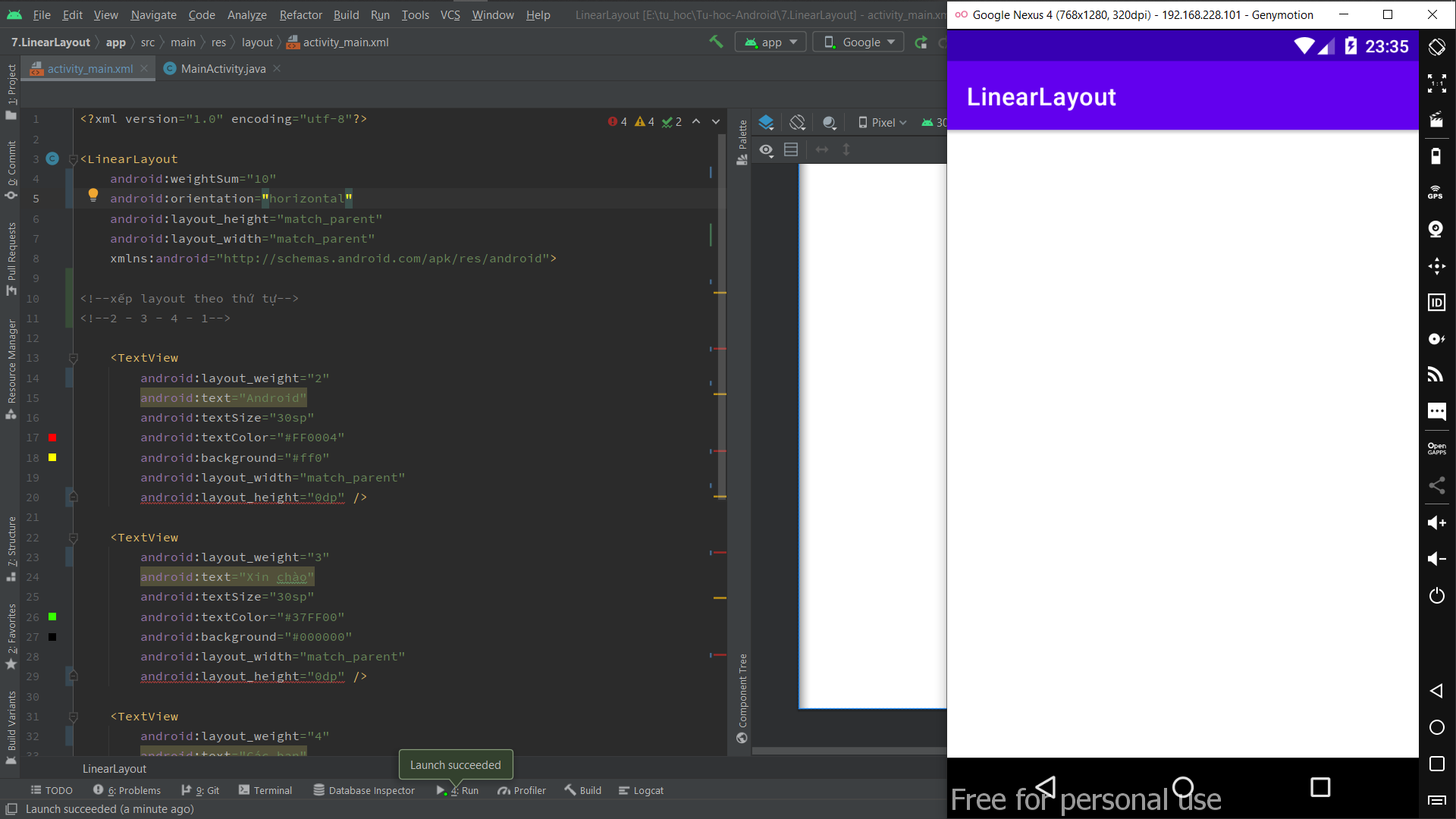
* Nếu muốn chia theo tỉ lệ nhất định
* Kết quả



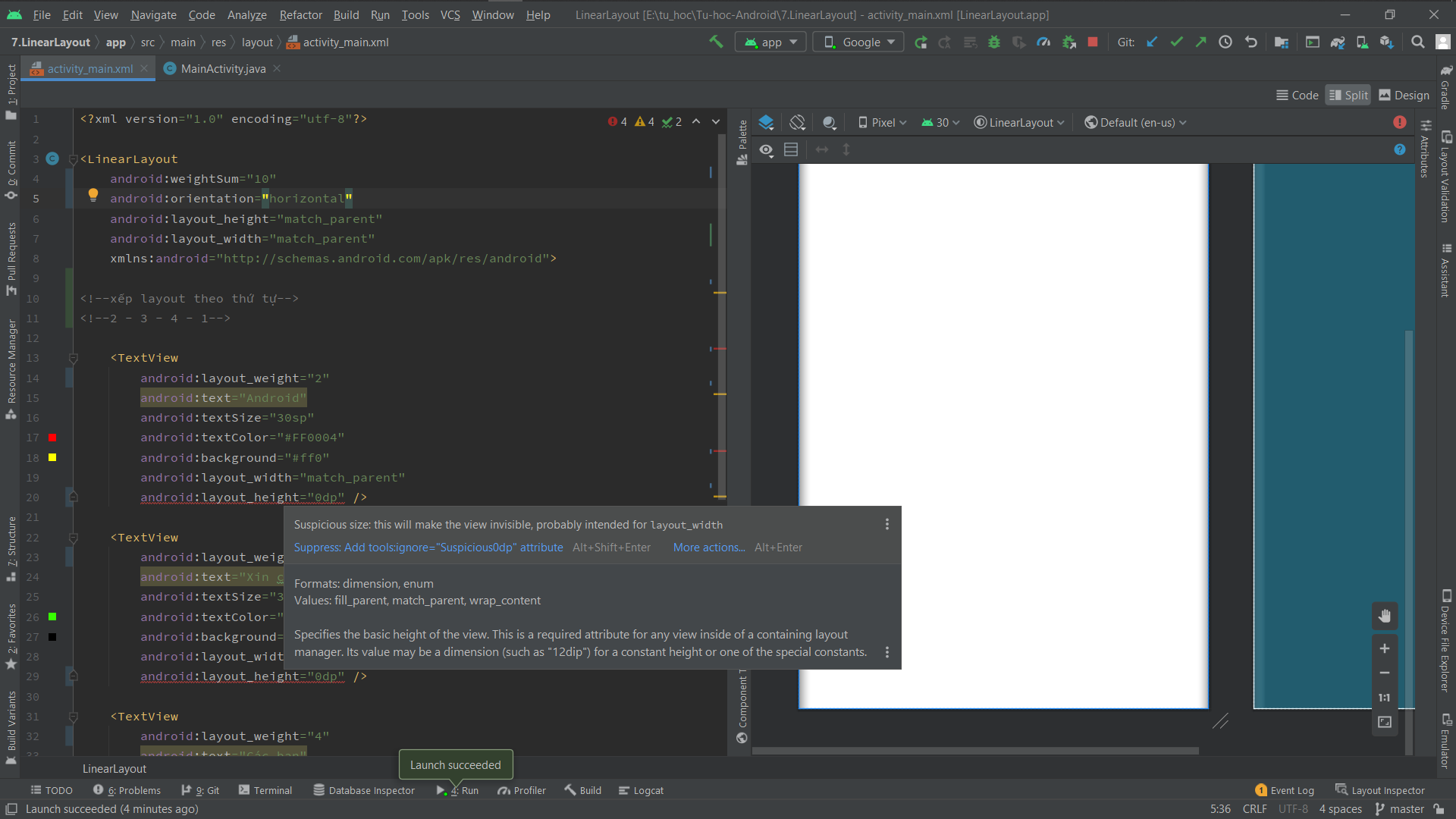
* Nếu như trên sẽ bị mất layout cam
* Vì khi dùng hàm weightsum thì không xài layout\_height = match\_parent nữa mà chuyển về 0dp
* Kết quả

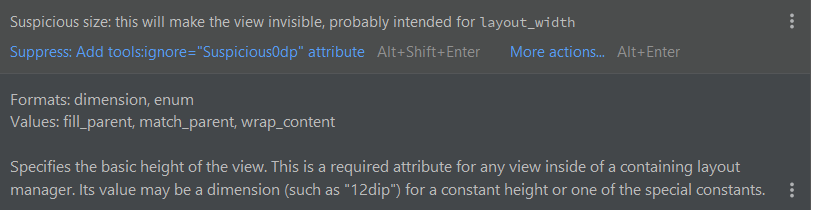


* Nếu giả sử theo chiều ngang
* Nếu giữ nguyên vậy và chuyển từ vertical sang horizonal
* Kết quả

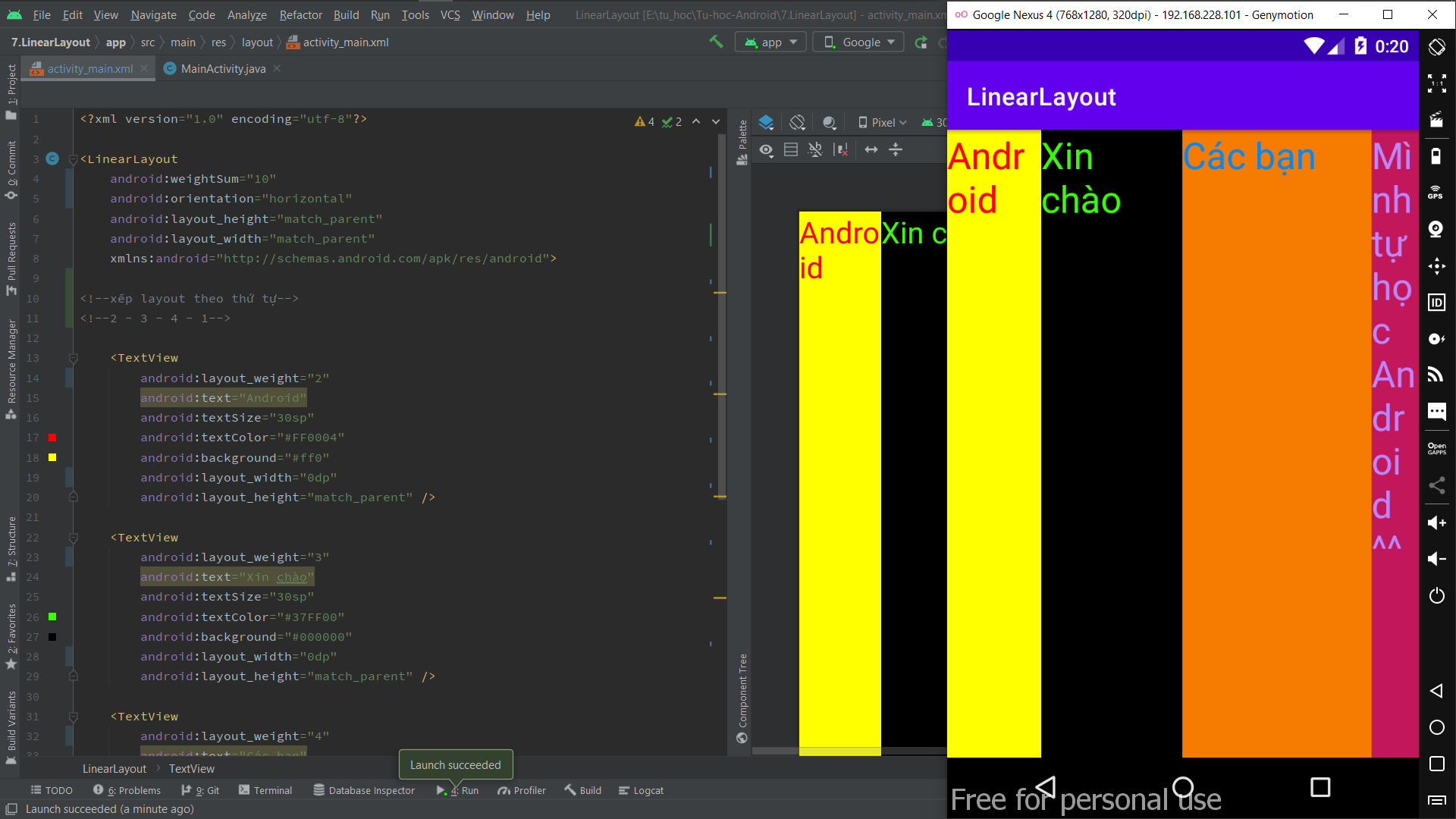


* Nó sẽ báo lỗi

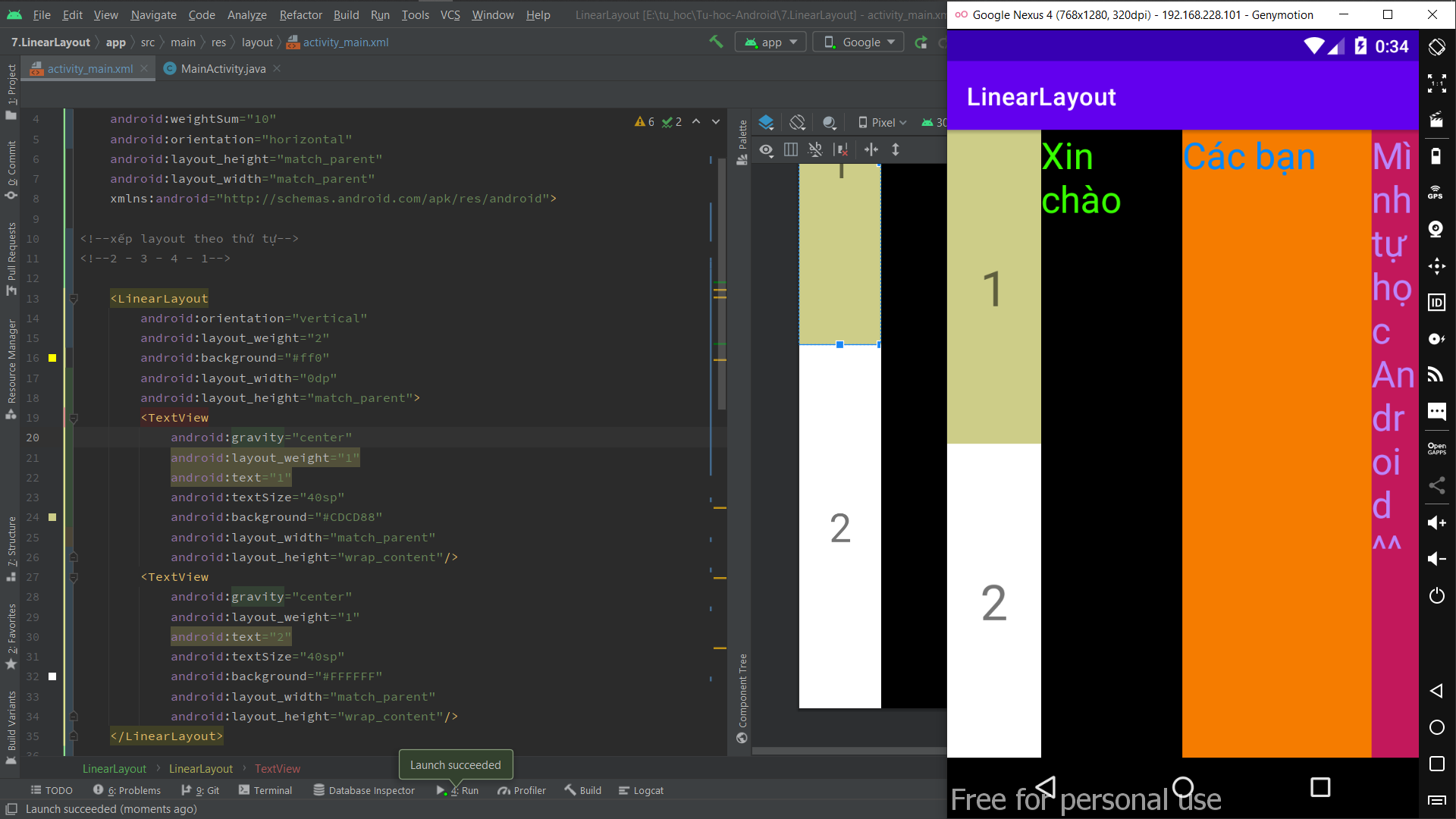


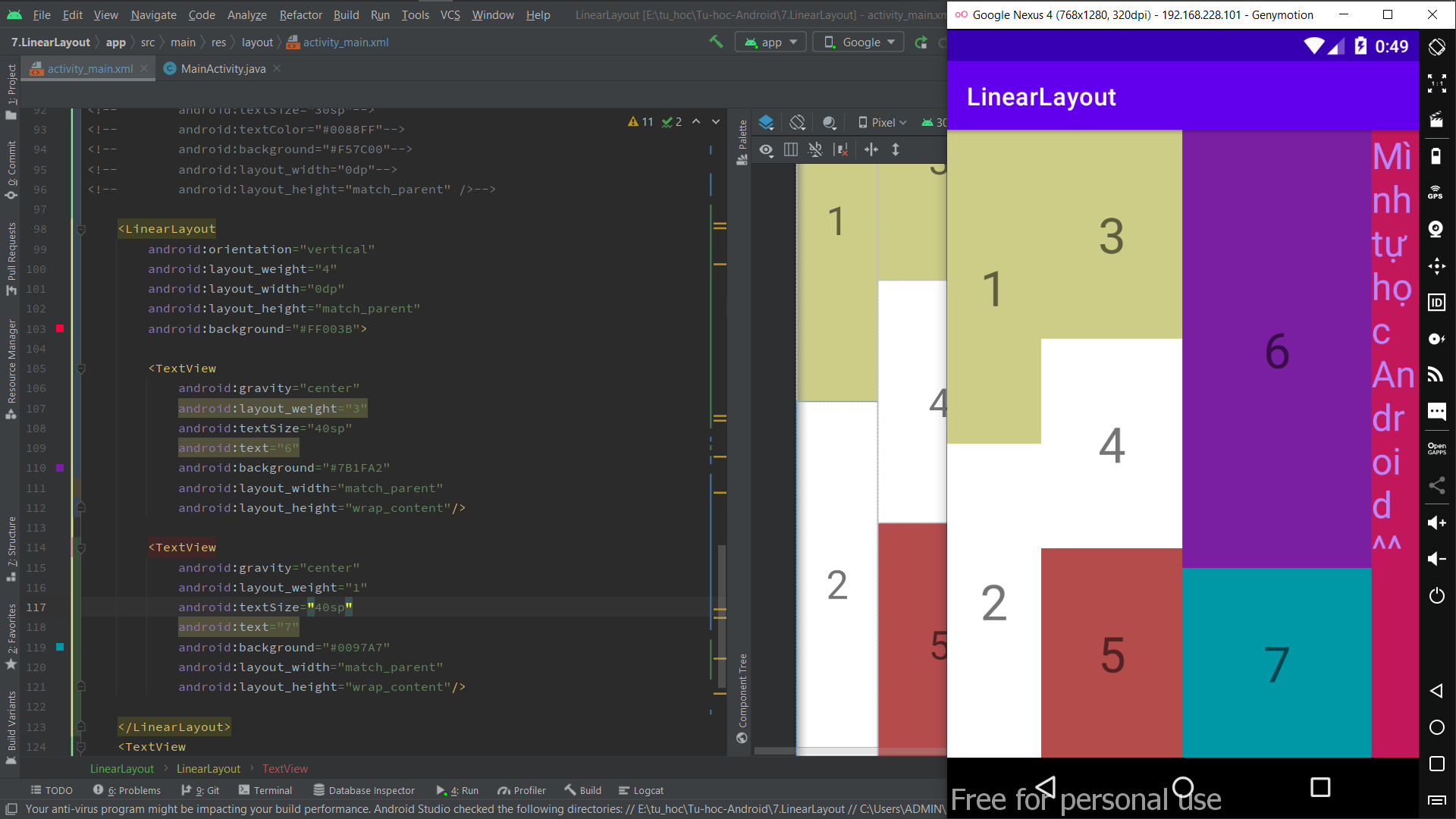


* Lỗi này ảnh hưởng đến layout và có thể bị ẩn
* Để sửa lại ta chỉ cần chỉnh layout\_width về 0dp
* Kết quả



* Chia các view con trong layout
* Kết quả





* + \* RelativeLayout:
  + FrameLayout
  + TableLayout

1. **Đặt icon cho ứng dụng**

* Cop file icon vào android studio => mipmap
* Mở androidMainfest đặt icon vào 2 mục icon và roundIcon, khi build thì ứng dụng sẽ sử dụng icon đã set

1. **Tạo view cho ứng dụng đầu tiên**
   1. **Bằng kéo thả**

* Vào file activity\_main.xml
* Cần gõ chữ được thì kéo plain text
* Nút => button
* Hiển thị văn bản => text view
* Neo các thành phần lại bằng guildeline (horizotnal)
* Sau đó kéo button và plantext vào gui
* Muốn button rộng sang phải hay trái, ấn vào nút sau đó nhìn vào layout bên tay phải có ô vuông xung quanh có dấu + tương ứng trái phải trên dưới
* Muốn to ra thì nhìn vào declared attributes => chọn layout width => match\_constranit
* Thay đổi màu nút search color