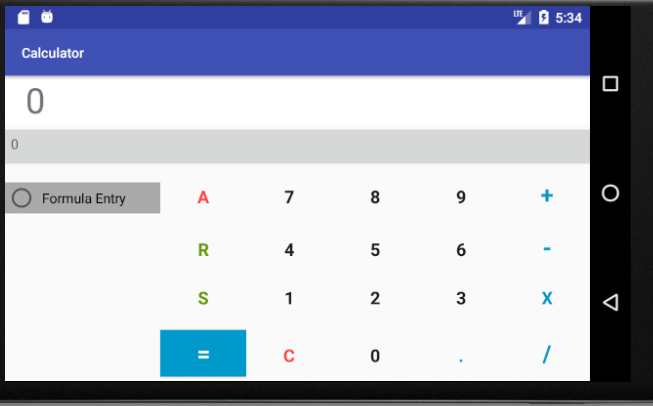
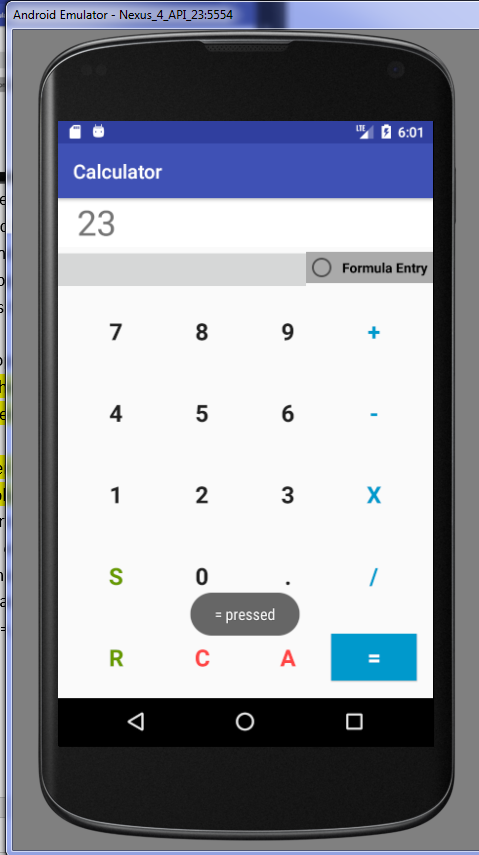
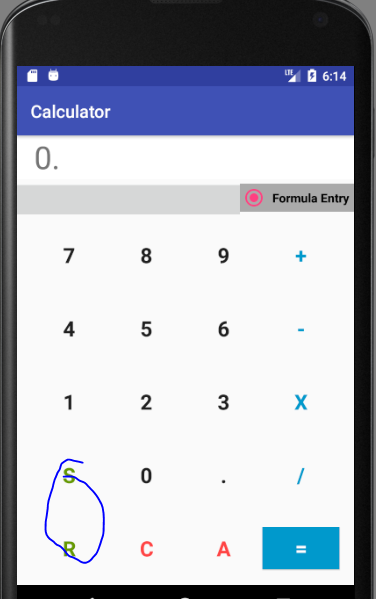
Documentations

1. Which API it is running on:
   1. API 23 (Marshmallow), and it is running on virtual device I created
   2. 
2. Basic function implemented
   1. Employed a layout that arranges the display and buttons such that they make reasonably effective use of the screen space
   2. Create separate layouts for portrait and landscape views
      1. This will take action when you flip (they are implemented using separate xmls)
      2. 
   3. Press a number key to append to the current operand
      1. If the display has just been cleared, or an operator was the last key pressed, this starts creating a new operand
   4. Press an operator key to select the operation
      1. My calculator will only takes the first operator, it will not take the second operator.
      2. In basic mode, pressing an operator after operand→operator→operand will first act as though you pressed the = key,
      3. and then pressed the selected operator, with the result of the prior calculation being the first operand
      4. Similarly, an = key will either perform a null operation if a single operand has been entered (e.g. 23= simply yields 23), or will perform the operation if the second operand has been provided (e.g. 2+3= yields 5)
      5. 
   5. Press your correct button (e.g. C) to either remove the last operator or the last digit (depending on which was pressed more recently)
      1. If it acts on an operand, it will edit (delete) the operant digit by digit
      2. If the last operation was an =, or to start with a clear display, there's no expected behaviour
   6. Press the All-Clear button (A, AC, Clear, etc.) to restore the display to 0, with no operators/operands selected
   7. If an operation is invalid (e.g. division by zero), either display an error, or reset the display to 0.
3. Additional Features: For full credit, you'll also need to make any three of the following six additions:
   1. Add and implement the decimal (.) key
      1. Pressing the decimal key a second time during the same operand will be ignored
      2. Pressing the decimal key after finishing an operation could will start a new operand with 0
   2. Add and implement a memory feature
      1. S for saving, R for recall
         1. 
      2. When you press to store, the operand/current working value is placed into memory. When you press to recall, the stored value becomes the current operand/working value
      3. **User will be able to add additional digits to the recalled value**
      4. If a new value is stored, it may simply replace the old one
      5. The default stored value (before you press Store) is 0
      6. A value will be remembered within the current execution of the application
   3. Formula entry
      1. Rather than only displaying a single operand at a time, queue in an entire expression, displaying as it's entered, to be calculated once = is pressed
      2. This must follow standard order of operations for operators (including parentheses, if present)
   4. If you do #5, a radio button (or comparable) for switching between formula entry and 'basic' mode
      1. Formula mode will shows like below
         1. 
      2. Basic mode will shows like below
         1. 