1. Begin.
2. Procedure Name: **main()**
3. img 🡨 Image Matrix. Read image using library function imread() and store it in it.
4. call play()
5. End.
6. Begin.
7. Procedure Name: **play() 🡨 Returns two integer lists.**

Parameter List: img 🡨 Image Matrix.

1. gray 🡨 Image Matrix. Convert read image to grey scale using library function cvtColor() and store it in it.
2. thresh🡨 Image Matrix. Threshold image gray to convert it to binary image to remove all unnecessary noise.
3. Define grid1\_number[] 🡨 Stores number that are in first grid.

grid1\_number\_postion[] 🡨 Stores position of numbers in first grid.

grid2\_number[] 🡨 Stores number that are in second grid.

grid2\_number\_position[]🡨 Stores position of numbers in second grid.

1. contours[] 🡨 library function findContours() creates this array and stores all the contours of the binary image thresh.
2. counter 🡨 1.
3. if counter<len(contours) is true
   * 1. if Area of contours is within range 800 to 2000
        1. Draw green outline on contours of width 2 using library function drawContours()
        2. area 🡨 Compute and store contour area by calling library function contourArea()
        3. perimeter 🡨 Compute and store contour area by calling library function arcLength().
        4. cx and cy 🡨 Compute and store x-coordinate and y-coordinate of centroid of contour by calling library function moments().
        5. found\_number 🡨 Compute and store number that the contour outlines by calling method number().
        6. if cx<300 then
           1. Insert the number in the list grid1\_number[].
           2. Insert the position of the number in the list grid1\_number\_postiton[].
        7. else
        8. Insert number in the list grid2\_number[].
        9. Insert position of the number in the list grid2\_number\_position[].
   1. counter 🡨 counter+1
   2. Goto 7.
4. Sort list grid1\_number\_position[] in ascending order using function sort() and use the indices of the same list to alter list grid1\_number[].
5. Sort list grid2\_number\_position[] in ascending order using function sort() and use the indices of the same list to alter list grid2\_number().
6. matrix 🡨 2D array to store the number as well as its position from the second grid.
7. i 🡨 0.
8. if i< len(grid2\_number)-1 then
   1. if grid2\_number\_postion[i+1]-grid2\_number\_position[i]<60

13.1.1 two\_digit\_number 🡨 grid2\_number[i]\*10+grid2\_number[i+1]

13.1.2 insert number and its position in array matrix.

* + 1. i 🡨 i+2
  1. else
     1. insert number and its position to array matrix.
     2. i 🡨 i+1
  2. goto 13.

1. if i== len(grid2\_number) -1  
   14.1 add number and position to array matrix.
2. Return grid1\_number and matrix.
3. End.