Store [8,29,50,81,4,23,24,30,1,7] in list one and store 10 spaces for in list two. Address r1 as list one and address r2 as list two.r2 acts as an auxiliary array and final sorted array will store in r1. MOV r8,#0 is r8 equals 0,it will be the counter for how many numbers have been compared. If r8 equals 10 means that all numbers have been compared. In layerone we will compare r8 with 10 first if it is not equal to 10, r3 will load the first number of listone and r4 will load the second number of listone compare r3 with r4, if r3 is greater than r4, it will store value of r4 in listtwo first and store value of r3 in next index, if r3 is smaller it will store value of r3 first and store value of r4 in next index. After that we will add 2 to r8 as we sorted 2 elements and repeat the loop until r8 equals to 10. When r8 equals to 10 it will jump to resettotwo. In resettotwo MOV,r8,#0 is to resets the “stored” counter. ADR,r1,listone and ADR,r2,listtwo to reset them.MOV r5,r1 and ADD r5,#2, r5 will be the second pointer that points to the beginning of second two-elements list. And r11 will be the anchor that signals end of rhs list, r7 is the anchor that signals end of lhs list, once r11 or r7 equals to the respective pointers, branches to copy a 4-elements-list. Load respective values into registers using LDRB r3,[r1],#1 and LDRB r4[r5],#1, after that branch to layertwo. Layertwo is to make two four-elements-lists.CMP r8,#4 if they are equal re-indexes the pointers to point at the start of the 3rd and 4th two-elements-list, if r8 is equals to 8 ,re-indexes the pointers to point at respective values. Compare r1 to r11 is to check if everything in right hand side list has been merged. If r1 is greater it will branch to endrhs2, Compare r5 with r7 is to check if everything in left hand side list has been merged. If r5 is greater, it will branch to endlhs2 to store every leftover right hand side list down to 4-element-list. Compare r3 with r4 to store whichever element that is smaller into the 4-elements-list and load whichever element that is stored with next element on the line, and increments the related pointer. Increments stored counter and branch back to layer to until the counter equals to 8. When the counter equals to 8 it will branch to resettothree. In resettothree it will reset the stored counter address r1 to point to the beginning of 4-elements-list and r5 will be the second pointer that points to the beginning of the second 4-elements-pointer. Register r11 will be the anchor that signals end of rhs list and r7 will be the anchor that signals end of lhs list, once r11/r7 equals to respective pointer, branches to copy the elements into an 8-element-list. After that loads respective values into registers and branch to layerthree. In layerthree we will compare r8 with 8, r8 will equals to 8 when all 8 elements from two 4-elements-lists have merged into one 8-elements-list. After that it will branch to resettofour. Compare r1 with r11 to check if everything in right hand side list has been merged. When r1 is greater it will store every leftover left-hand-side list down to 8-element-list. Compare r5 with r7 , if r5 is greater it will store every leftover right-hand-side list down to 8-elements-list.Compare r3 with r4 and stores whichever element that is smaller into the 4-elements-list and load whichever element that is stored with next element on the line and increments the related pointer. After that stored counter will increase and loop layerthree until r8 equals to 8. In resettofour, resets the stored counter and address 1 points to the beginning of the 8-elements-list. Register r5 will be the second pointer that points to the leftover 2-elements list. R11 is anchor that signals end of rhs list and r7 is anchor that signals end of lhs list, once r11 or r7 equals to respective pointer, branches to copy the elements into a 10-elements list and the list is fully sorted. In layerfour compare r8 with 10 as it will be equal when all 8 and 2 elements from two lists have been merged into one 10-element-list. Compare r1 with r11 to check if everything in right-hand-side list has been merged, if r1 is greater it will store every leftover left-hand-side list down to 10-elements list. Compare r5 with r7 to check if everything in left-hand-side list has been merged. If r5 is greater, it will store every leftover right-hand-side list down to 10-elements list, add counter by 1 and loop layerfour until r8 equals to 10, the program will end as all 10 elements have been sorted in a 10-elements-list at r1.