1. Write an ARM assembly program that compares three integers stored in R0, R1 and R2. Store the greater number in R3 using conditional branch instructions.
2. Write an ARM assembly program that checks whether the value in R0 is greater than the value in R1. If so, swap the values of R0 and R1. If not, leave them unchanged.

**Write equivalent ARM assembly for the following program:**

3:

|  |
| --- |
| for (int i = 0; i < 10; i++) {  sum += i;  } |

4:

|  |
| --- |
| if (R0 < R1)  R3 = 1;  else if (R0 == R1)  R3 = 0;  else  R3 = 2; |

|  |
| --- |
| if (score >= 80)  grade = 'A';  else if (score >= 60)  grade = 'B';  else  grade = 'F'; |

5:

**6:**

|  |
| --- |
| int add(int a, int b) {  return a + b;  }  int main() {  int x = add(2, 3);  } |

**7.**

|  |
| --- |
| int a = 20, b = 6;  int quotient = a / b;  int remainder = a % b;  **solution:**  .global \_start  \_start:  MOV R0, #20 @ Dividend (a)  MOV R1, #6 @ Divisor (b)  MOV R2, #0 @ Quotient = 0  @ R3 will hold remainder  div\_loop:  CMP R0, R1 @ Compare dividend and divisor  BLT done\_div @ If dividend < divisor, exit loop  SUB R0, R0, R1 @ R0 = R0 - R1 (subtract divisor)  ADD R2, R2, #1 @ Quotient++  B div\_loop @ Repeat  done\_div:  MOV R3, R0 @ Remainder = leftover in R0  end:  B end |

**Peripherals:**

8. Write an ARMv7 Assembly program that displays the digits 0 to 9 on the 7-segment display, with a certain delay.

9. Implement a countdown timer starting from 9 to 0 on 7-Segment

10. Display the digit "8" five times on the 7-segment display with on (e.g., 2 seconds) and off intervals.

11. Display Even Numbers (0, 2, 4, 6, 8) on 7-Segment. Loop through even digits and show each for 1 second.