Compare two numbers and store the larger one

|  |
| --- |
| .global \_start  .data  num1: .word 15 @ First number  num2: .word 22 @ Second number  .text  \_start:  LDR R3, =num1 @ Load address of num1  LDR R0, [R3] @ Load value of num1 into R0  LDR R3, =num2 @ Load address of num2  LDR R1, [R3] @ Load value of num2 into R1  CMP R0, R1 @ Compare num1 and num2  BGT first\_is\_greater  MOV R2, R1 @ If num2 is greater or equal, store it in R2  B done  first\_is\_greater:  MOV R2, R0 @ If num1 is greater, store it in R2  done:  @ End of program |

Even or Odd checker:

|  |
| --- |
| .global \_start  .data  number: .word 13 @ Change this to test different numbers  .text  \_start:  LDR R0, =number @ Load address of 'number'  LDR R0, [R0] @ Load value from memory into R0  AND R2, R0, #1 @ Check if LSB is 1 (odd) or 0 (even)  CMP R2, #0  BEQ is\_even @ Branch if even  B is\_odd @ Branch if odd  is\_even:  MOV R1, #0 @ 0 for even  B done  is\_odd:  MOV R1, #1 @ 1 for odd  done:  @ Program ends here |

Find maximum of three numbers:

|  |
| --- |
| .global \_start  \_start:  MOV R0, #10 @ First number  MOV R1, #20 @ Second number  MOV R2, #15 @ Third number  CMP R0, R1  BGT r0\_greater  MOV R3, R1 @ R1 is greater than or equal to R0  B compare\_r3\_r2  r0\_greater:  MOV R3, R0 @ R0 is greater than R1  compare\_r3\_r2:  CMP R3, R2  BGT done  MOV R3, R2 @ Update max if R2 is greater  done:  @ R3 now contains the max value |