1)

```
           CMP     R1, #0         ; Compare R1 to 0
           BLE     else           ; if R1 < 0, branch to else

           ADDS    R0, R0, R1  ; R0 = R0 +R1
           B       end_if         ; branch to end if

else_part                         ; else
           SUBS    R0, R0, R1; R0 = R0 - R1
end_if                            ; end if
```

2)

```
loop                              ; while true
           LDR     R1, =MaxCount   ; load address of MaxCount
           LDR     R1, [R1]        ; load MaxCount value
           CMP     R0, R1          ; Compare values
           BGE     end_loop        ; if R0 >= MaxCount, branch to break
                                  ;
           ADDS    R0, R0, #1      ; R0 = R0 + 1, Add 1 to Ro
           B       loop            ; repeat loop

end_loop                          ; break
```

3)

```
           LDR     R0, =Time        ; Load address of variable Time into R0
           MOVS  R1, #100          ; R1 = 100, initialize Time value
           STR     R1, [R0]         ; Time = 100 → store initial value to memory

Loop                                ; While

           LDR     R1, [R0]         ; Load value of Time from memory
           ADDS  R1, R1, #1        ; R1 = R1 + 1 → increment Time
           STR     R1, [R0]         ; Store updated Time back to memory

           CMP     R1, #250         ; Compare Time with 250
           BNE     loop             ; if Time = 250, break
```

4)

```
        MOVS    R1, #10                ; R1 = 10 → initialize loop counter

for
        LDR     R2, =N                 ; Load address of variable N
        LDR     R2, [R2]               ; Load value of N into R2
        CMP     R1, R2                 ; Compare R1 with N
        BCS     end_for                ; If R1 > N, branch to end loop

        LSLS    R0, R0, #1             ; R0 = R0 << 1, shift left one bit
        ADDS    R1, R1, #1             ; R1 = R1 + 1, increment loop counter
        B       for                    ; Repeat loop

end_for                                ; End loop
```