```
    AREA deneme, CODE
    EXPORT __main

__main

    ; Move Komutlari
    MOV R0,#0x11            ; R0 registerina 11 degerini hexadecimal olarak yazar
    MOV R1, #2398           ; R1 registerina 2398 degerini hexadecimal olarak yazar
    MVN R2, #4              ; R2 registerina 4 degerinin hexadecimal karsiliginin degilini yazar
    MOVW R3, #0xABCD        ; R3 registerinin [0,15] 16 bitine ABCD degerini yazar
    MOVT R3, #0xEFED        ; R3 registerinin [16,31] 16 bitine EFED degerini yazar
    MOV32 R4, #0x12345678   ; R4 registerina 32bitlik deger olan 12345678 degerini yazar
    MOV R5,R1               ; R1 registerindaki degeri R5'e kopyalar

    ; Load/Store Komutlari
    ; Koseli parantezin ici veriyi almak istedigimiz bellegin adresini tutar
    LDR R5,[R1]             ; Adresin gosterdigi degeri R5 registerina yazar, pasif bir koddur.
    STR R3,[R1]             ; Adresin gosterdigi degeri R3 registerina yazar, aktif bir koddur.
                            ; Aktif olmasi degerin bellege yazilmasidir.

    ; Bit Islemleri
    ORR R0,#(1<<4)          ; OR Islemi
    AND R0,#~(1<<4)         ; AND Islemi

    ; Aritmatik Islemler
    ADD R1,R0,R0            ; R0 ile R0'i toplayip sonucu R1'e yazar
    ADD R2,R0,#2            ; R0 ile 2 yi toplayip sonucu R2'e yazar
    ADD R2,R0               ; R0 ve R2 yi toplayip sonucu R2'e yazar
    MUL R2,R0,R1            ; R0 ve R1 i carpip sonucu R2'e yazar
    UDIV R2,R0,R1           ; Isaretsiz bolme, R0'i R1'e boler ve sonucu R2'e yazar
    SUB R0,R1,#1            ; R1 den 1 cikartip sonucu R0'a yazar

    ; Karsilastirma islemleri
    CMP R3,#0               ; Hedef: R3, Kaynak: #0
                            ; Hedef > Kaynak ise ZeroFlag=0 , CarryFlag=0
                            ; Hedef < Kaynak ise ZeroFlag=0 , CarryFlag=1
                            ; Hedef = Kaynak ise ZeroFlag=1 , CarryFlag=0

    ; LDR Kullanimi
    MOV R0,#0x1000          ;PortC CRH registerinin adresi: 0x40011000
    MOVT R0, #0x4001        ;Iki komutta bu adresi R0 registerina yukledik
    LDR R1,[R0,#0x04]       ;PortC CRH registerinin ofseti: 0x04 '[R*,#Data_Ofset]' olarak yazilir
                            ;Portc nin CRH register verisi R1 registerina yüklenir

    ; STR Kullanimi
    ; RCC->APB2ENR |= (1<<4) 4.bit PortC yi aktiflestirir adres 0x40021000, ofset: 0x18
    MOV R0,#0x1000
    MOVT R0, #0x4002        ; RCC biriminin adresi
    LDR R1,[R0,#0x18]       ; APB2ENR registerinin adresi
    MOV R2, #(1<<4)         ;
    STR R2,[R0,#0x18]       ; APB2ENR registerinin 4.biti set olur

    MOV R0,#0x1000
    MOVT R0, #0x4001        ; PortC nin adresi
    LDR R1,[R0,#0x04]       ; CRH registerinin adresi
    ORR R1, #(1<<21)        ; 21. bit 1 yapilir
    AND R1,#~(1<<22)        ; 22. bit 0 yapilir
    STR R1,[R0,#0x04]       ;


loop
    LDR R1,[R0,#0x0C]       ; ODR registerinin adresi
    ORR R1, #(1<<13)        ; 13. bit 1 yapilir
    STR R1,[R0,#0x0C]       ;

    LDR R1,[R0,#0x0C]       ; ODR registerinin adresi
    AND R1, #~(1<<13)       ; 13. bit 0 yapilir
    STR R1,[R0,#0x0C]       ;
    B loop ; Sonsuz dongu


    END
```