```
//*********************************************************************************
*****************
// Name: Samer AlDayoub
// Date: 09/20/2021
// Course: ELEC3371-00
// Description: Project 1 code. This program devides pre-assigned values using
successive subtraction.
// The devidant and devisor are declared and given values befoer building the code.
// The result is stored i   n a variable (Q) and the remainder is (R)
// the code dont do operations if the denominator is 0.
//
// Enjoy!!
//
***********************************************************************************
***************

//
***********************************************************************************
***************
//GLOBAL VARIABLES
unsigned long int num=56, den=23, Q=0, R=0;
//
***********************************************************************************
***************
//MAIN FUNCTION
void main() {
//
***********************************************************************************
***************
//ASSEMBLY
        asm{

                  MOVW R0, #LO_ADDR(_num+0)      ; Load the numerator value to
register R1
                  MOVT R0, #HI_ADDR(_num+0)
                  LDR  R1, [R0]

                  MOVW R0, #LO_ADDR(_den+0)      ; Load the denuminator value to
register R2
                  MOVT R0, #HI_ADDR(_den+0)
                  LDR  R2, [R0]


                  MOVW R0, #LO_ADDR(_Q+0)        ; Set R0 to store the quotient
                  MOVT R0, #HI_ADDR(_Q+0)


                  MOVW R5, #LO_ADDR(_R+0)        ; Set R5 to store the remainder
                  MOVT R5, #HI_ADDR(_R+0)


                  CMP                R2, #0      ; if the denominator is 0 go to
end
                  BNE                REPEAT      ; otherwise do the math.
                  B                  LABEL3

            REPEAT:                        ; successive subtraction loop
                  CMP          R1, R2
                  BLE          LABEL1      ; if the numenator <= denuminator finish
up
                  ADD          R3, #1      ; R3 would contain the final quotient
                  SUB          R1, R2      ; successive subtraction
                  B            REPEAT
            LABEL1:

                  CMP          R1, R2
                  BNE          LABEL2      ; if they are not equal then we got the
quotient and remainder
                  SUB          R1, R2      ; if they are equal at this point no
remainder left.
                  ADD          R3, #1      ; and the denuminator could go one more
time in the numenator

            LABEL2:

                  STR          R3, [R0]    ; store quotient to Q
                  STR          R1, [R5]    ; store remainder to R
                  MVN          R1, R1      ; Reverse the bits in the remainder
register
                  B            END
            LABEL3:
                   NOP                     ; Denominator cant be zero

            END:
                 NOP
        }
              }
```