```
;-------------------------------------------------------------------------------
; MSP430 Assembler Code Template for use with TI Code Composer Studio
;
;
;-------------------------------------------------------------------------------
            .cdecls C,LIST,"msp430.h"       ; Include device header file

;-------------------------------------------------------------------------------
            .def    RESET                   ; Export program entry-point to
                                            ; make it known to linker.
;-------------------------------------------------------------------------------
            .data                           ; Assemble into program memory.
            .retain                         ; Override ELF conditional linking
                                            ; and retain current section.
            .retainrefs                     ; And retain any sections that have
                                            ; references to current section.
result1:    .space  2
result2:    .space  2
;-------------------------------------------------------------------------------
            .text                           ; Assemble into program memory.
            .retain                         ; Override ELF conditional linking
                                            ; and retain current section.
            .retainrefs                     ; And retain any sections that have
                                            ; references to current section.

;-------------------------------------------------------------------------------
RESET       mov.w   #__STACK_END,SP         ; Initialize stackpointer
StopWDT     mov.w   #WDTPW|WDTHOLD,&WDTCTL  ; Stop watchdog timer


;-------------------------------------------------------------------------------
; Main loop here
;-------------------------------------------------------------------------------
; call subroutine MyDiv
; Divide 880 by 4
            mov.w  #880, R12                ; prepare input for MyDiv
            ; Divide by 4,
            ; log2 of 4 is 2, i.e., 2^2 = 4
            mov.w  #2, R13
            call   #MyDiv                   ; call subroutine MyDiv
            mov.w  R14, &result1            ; move result from the output of
                                            ; MyDiv to variable result1

; call subroutine MyDiv
; Divide 3520 by 8
            mov.w  #3520, R12               ; prepare input for MyDiv
            ; Divide by 8
            ; log2 of 8 is 3, i.e., 2^3 = 8
            mov.w  #3, R13
            call   #MyDiv                   ; call subroutine MyDiv
            mov.w  R14, &result2            ; move result from the output of
                                            ; MyDiv to variable result2

Loop:       jmp     Loop

;-------------------------------------------------------------------------------
;                             Subroutine: MyDiv
```

```
58 ;                                ------------------
59 ;   Input: R12 (Number to divide); R12 will be modified by the Subroutine
60 ;   Input: R13 (Log2 of the Number to divide by); R13 will not be modified by the subroutine
61 ;   Output: R14 : R12/2^R13; R14 will be modified by the subroutine
62 ;
63 ;-------------------------------------------------------------------------------
64 MyDiv:
65             ; R11 will be used as array index
66             push.w  R11         ; store R11's value on the stack
67
68             mov.w   #0, R11
69 cond_for:
70             cmp.w   R13, R11
71             jge     break_for
72
73             rra.w   R12
74             inc.w   R11
75
76             jmp     cond_for
77 break_for:
78             mov.w   R12, R14
79             pop.w   R11         ; restore R11's value from the stack
80             ret
81 ;-------------------------------------------------------------------------------
82 ; Stack Pointer definition
83 ;-------------------------------------------------------------------------------
84             .global __STACK_END
85             .sect   .stack
86
87 ;-------------------------------------------------------------------------------
88 ; Interrupt Vectors
89 ;-------------------------------------------------------------------------------
90             .sect   ".reset"                ; MSP430 RESET Vector
91             .short  RESET
92
93
```