

main.asm

```

1 ;-----
2 ; MSP430 Assembler Code Template for use with TI Code Composer Studio
3 ;
4 ;
5 ;-----
6     .cdecls C,LIST,"msp430.h"      ; Include device header file
7
8 ;-----
9     .def      RESET                ; Export program entry-point to
10                                ; make it known to linker.
11 ;-----
12 ;-----
13     .data                          ; Assemble into program memory.
14     .retain                        ; Override ELF conditional linking
15                                ; and retain current section.
16     .retainrefs                    ; And retain any sections that have
17                                ; that have references to current
18                                ; section
19 a:     .byte    10, 7, -5, 6, 9, -12, 5, 21, -4, -1
20 temp:  .space  1
21 ;-----
22     .text                          ; Assemble into program memory.
23     .retain                        ; Override ELF conditional linking
24                                ; and retain current section.
25     .retainrefs                    ; And retain any sections that have
26                                ; references to current section.
27
28 ;-----
29 RESET    mov.w    #__STACK_END,SP    ; Initialize stackpointer
30 StopWDT  mov.w    #WDTPW|WDTHOLD,&WDTCTL ; Stop watchdog timer
31
32
33 ;-----
34 ; Main loop here
35 ;-----
36     mov.w    #0, R5                ; i = 0
37 for1_cond:
38     cmp.w    #9, R5                ; i >= 9
39     jge for1_break
40
41 ;----- for_loop2 begin -----
42
43     mov.w    #1, R6                ; j = 1
44     add.w    R5, R6                ; j = i + 1
45 for2_cond:
46     cmp.w    #10, R6 ; j >= 10
47     jge for2_break
48
49 ;----- if-structure begin -----
50
51     cmp.b    a(R6), a(R5)         ; a[j] <= a[i]
52     jge if_break
53
54     mov.b    a(R5), &temp         ; temp = a[i]
55     mov.b    a(R6), a(R5)         ; a[i] = a[j]
56     mov.b    &temp, a(R6)        ; a[j] = temp
57

```

main.asm

```
58 if_break:
59 ;----- if-structure end -----
60
61         inc.w   R6           ; ++j
62         jmp    for2_cond
63 for2_break:
64 ;----- for_loop2 end -----
65
66         inc.w   R5           ; ++i
67         jmp    for1_cond
68 for1_break:
69 ;----- for_loop1 end -----
70
71 loop:   jmp    loop
72
73
74
75 ;-----
76 ; Stack Pointer definition
77 ;-----
78         .global __STACK_END
79         .sect   .stack
80
81 ;-----
82 ; Interrupt Vectors
83 ;-----
84         .sect   ".reset"           ; MSP430 RESET Vector
85         .short  RESET
86
87
```