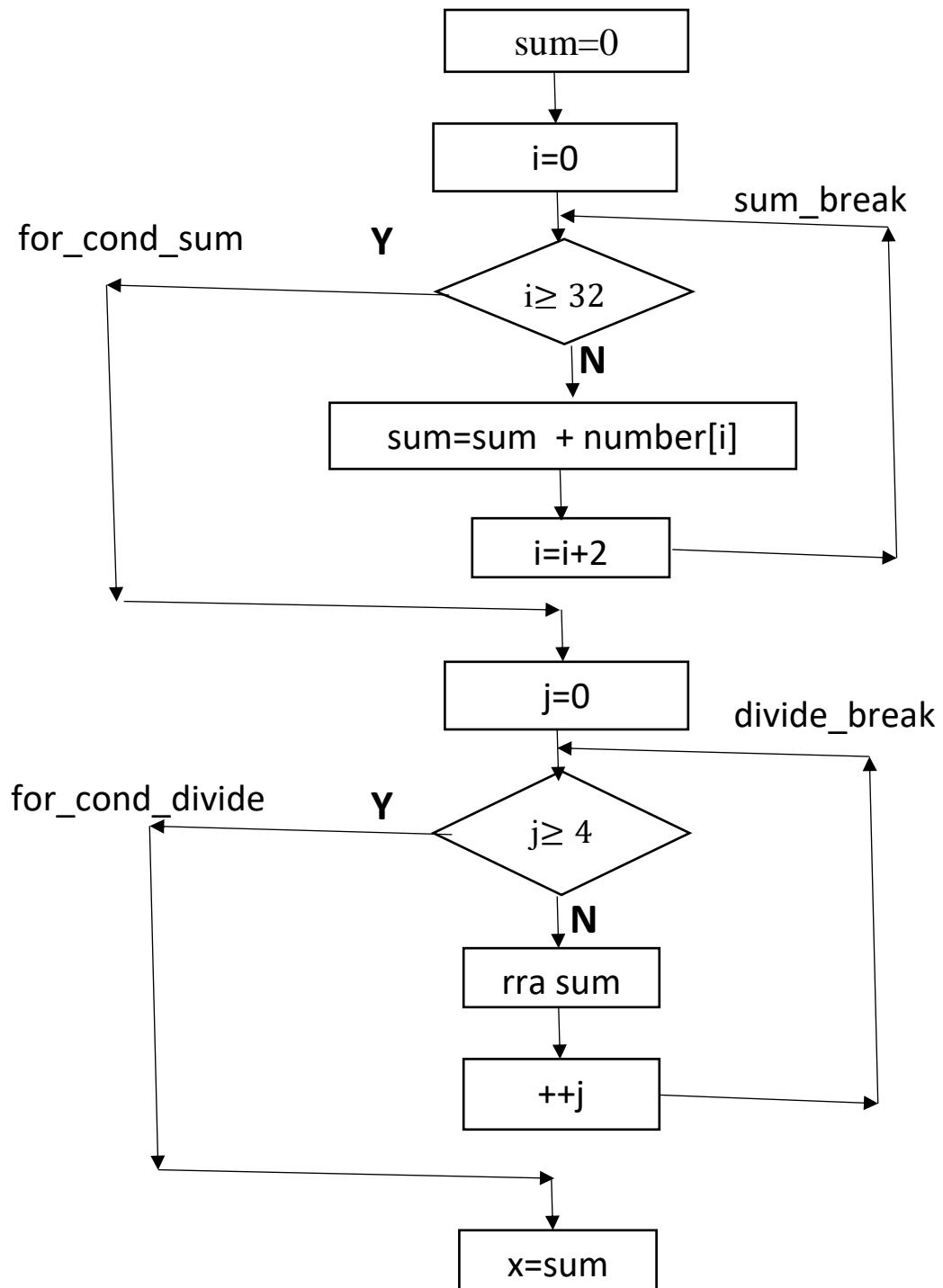


ECE2560 SP2017 Midterm 2 Solution

Pseudo code:

```
sum=0;  
For (i=0, i<32, i=i+2)  
{  
    sum=sum+Numbers[i]  
}  
//divide sum by 16=24 by using bit shifting  
For(j=0, j<4, ++j)  
{  
    Roll right sum by one bit;  
}  
x=sum;
```

Flowchart:



Code:

```
;-----  
        .cdecls C,LIST, "msp430.h"      ; Include device header file  
  
;-----  
        .def    RESET                  ; Export program entry-point to  
                                ; make it known to linker.  
;  
        .data  
Numbers:     .word  -2000,-1000,-550,-450,-400,-600,-1600,-1400,-2500,500,-  
1250,250,-2000,-1100,-1400,-100  
x:           .space 2  
sum:         .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  
;  
        mov.w   #0, R10                 ; use R10 as index i  
for_cond_sum:  
        cmp.w   #32, R10               ; compare index with 16  
        jge     sum_break             ; jump out of loop if  
index i>=16  
        add.w   Numbers(R10), &sum   ; add element to sum  
        incd.w R10                  ; increase index i by 2  
        jmp     for_cond_sum  
sum_break:  
        mov.w   #0, R11                 ; use R11 as index j  
for_cond_divide:  
        cmp.w   #4, R11                ; compare index with 4  
        jge     divide_break          ; Jump if content R11 >= k=4  
        rra.w   &sum                  ; bit-shift right by one  
        inc.w   R11                  ; increase index j by 1  
        jmp     for_cond_divide  
divide_break:  
        mov.w   &sum, &x              ; average should be -975  
Loop:      jmp     Loop  
;  
; Stack Pointer definition  
;  
        .global __STACK_END  
.sect   .stack  
;
```

```
; Interrupt Vectors  
;  
.sect  ".reset"           ; MSP430 RESET Vector  
.short RESET
```

Screenshot of the memory browser:

The screenshot shows a memory browser window titled "Memory Browser". The main pane displays memory starting at address 0x0200, specifically the range 0x0200 - 0x0200. The data is presented in a grid format with 16-bit signed integers. The column headers are addresses from 0x0200 to 0x0250 and values from -2000 to -6149. The value at address 0x021E is highlighted with a blue background.

0x0200	-2000	-1000	-550	-450	-400
0x020A	-600	-1600	-1400	-2500	500
0x0214	-1250	250	-2000	-1100	-1400
0x021E	-100	-975	-975	16900	5158
0x0228	9248	4108	8226	-27100	3097
0x0232	5176	11456	10816	19016	7680
0x023C	16417	10416	-8449	-4125	-3859
0x0246	-65	-9225	-614	-4225	-546
0x0250	30078	-1549	-1155	-4741	-6149