

## ECE2560 SP2017 Midterm 2 Solution

### Pseudo code:

```
sum=0;
```

```
For (i=0, i<32, i=i+2)
```

```
{
```

```
    sum=sum+Numbers[i]
```

```
}
```

```
//divide sum by  $16=2^4$  by using bit shifting
```

```
For(j=0, j<4, ++j)
```

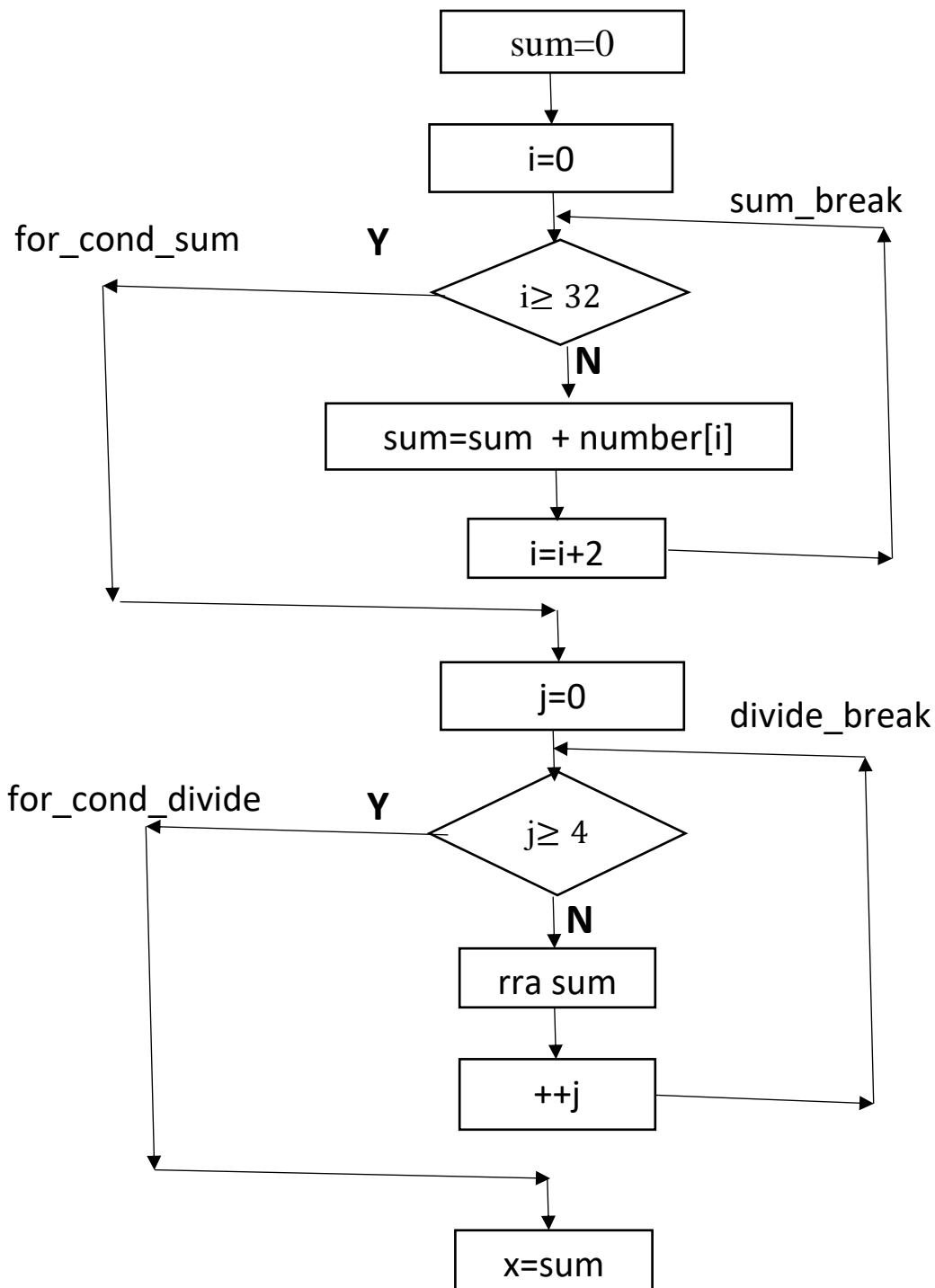
```
{
```

```
    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:**

