```asm
18        pushad              ;to preserve register values
19        call print_num      ;number in eax register
20        popad               ;restore register values
21
22        mov eax,4           ;for new line
23        mov ebx,1
24        mov ecx,n
25        mov edx,1
26        int 0x80
27
28        mov    eax, 1       ;sys_exit
29        int    0x80         ;interupt
30
31  section .data
32      x dw 400              ;operand 1
33      y dw 500              ;operand 2
34      n db '',0xA           ;new line
35      z db '-'
36      msg db 'The Sum is:'
37      len equ $-msg
38
39  section .bss
40  res resb 1                ;result variable
41
```

```asm
33      mov eax,4           ;for new line
34      mov ebx,1
35      mov ecx,n
36      mov edx,1
37      int 0x80
38
39      mov     eax, 1      ;sys_exit
40      int     0x80        ;interupt
41
42  section .data
43      x dw 200            ;operand 1
44      y dw 159            ;operand 2
45      z db '-'            ;'-' sybmol
46      n db '',0xA         ;new line
47      max dw 65535        ;max value for word
48      msg db 'The Differnce is:'
49      len equ $-msg
50
51  section .bss
52  res resb 1              ;result variable
53
```

File   Edit   Selection   View   Go   Run   Terminal   Help

EXPLORER

ASM function.asm   ASM q3.asm   ✕

∨ ASSEMBLY_CODE

week-2 > ASM q3.asm

∨ week-2

ASM function.asm

ASM hello.asm

ASM q1.asm

ASM q2.asm

≡ q3

ASM q3.asm

≡ q3.o

ASM q4.asm

ASM q5.asm

ASM q6.asm

ASM q7.asm

ASM q8.asm

ASM q9.asm

ASM q10.asm

ASM q11.asm

⅄ ics312_datasize.pdf

⅄ M2L1.pdf

⅄ nasmdoc.pdf

⅄ ProgrammingGround...

```asm
20      pushad          ;to preserve register values
21      call print_num  ;number in eax register
22      popad           ;restore register values
23
24      mov eax,4       ;for new line
25      mov ebx,1
26      mov ecx,n
27      mov edx,1
28      int 0x80
29
30      mov     eax, 1  ;sys_exit
31      int     0x80    ;interupt
32
33  section .data
34      x dd 675        ;operand 1
35      y dd 234        ;operand 2
36      n db '',0xA
37      z db '-'
38      msg db 'The Product is:'
39      len equ $-msg
40
41  section .bss
42  res resb 1          ;result variable
43
```

TERMINAL   PROBLEMS   OUTPUT   DEBUG CONSOLE

1: wsl

manas@LAPTOP-U09MU8FI:/mnt/d/assembly_code/week-2$ nasm -f elf q2.asm
manas@LAPTOP-U09MU8FI:/mnt/d/assembly_code/week-2$ ld -m elf_i386 -s -o q2 q2.o
manas@LAPTOP-U09MU8FI:/mnt/d/assembly_code/week-2$ nasm -f elf q2.asm
manas@LAPTOP-U09MU8FI:/mnt/d/assembly_code/week-2$ ld -m elf_i386 -s -o q2 q2.o
manas@LAPTOP-U09MU8FI:/mnt/d/assembly_code/week-2$ ./q2
The Differnce is:41
manas@LAPTOP-U09MU8FI:/mnt/d/assembly_code/week-2$ nasm -f elf q3.asm
manas@LAPTOP-U09MU8FI:/mnt/d/assembly_code/week-2$ ld -m elf_i386 -s -o q3 q3.o
manas@LAPTOP-U09MU8FI:/mnt/d/assembly_code/week-2$ ./q3
The Product is:157950
manas@LAPTOP-U09MU8FI:/mnt/d/assembly_code/week-2$

> OUTLINE

⊗ 0 ⚠ 0   Live Share   Ln 35, Col 13   Spaces: 4   UTF-8   CRLF   ARM   Go Live

Type here to search   ENG   13:58   24-02-2021

```asm
47          mov edx,1
48          int 0x80
49
50          mov     eax,1     ;sys_exit
51          int     0x80      ;interupt
52
53   section .data
54          x dw 900          ;operand 1
55          y dw 400          ;operand 2
56          n db '',0xA
57          z db '-'
58          msg db 'The Quotient is:'
59          len equ $-msg
60          msg2 db 'The remainder is:'
61          len2 equ $-msg2
62
63   section .bss
64   res resb 1               ;result variable
65
```

```asm
32      mov eax,3
33      int 0x80
34
35      mov eax,4        ;new line
36      mov ebx,1
37      mov ecx,n
38      mov edx,1
39      int 0x80
40
41
42      mov eax,1        ;sys_exit
43      int 0x80         ;interupt
44
45      section .data
46      num db '100'     ;operand
47      msg db 'after increment:'
48      len equ $-msg
49      n db '',0xA
50
```

```asm
27
28      mov [num+esi],al        ;print num
29      mov eax,4
30      mov ebx,1
31      mov ecx,num
32      mov edx,3
33      int 0x80
34
35      mov eax,4        ;new line
36      mov ebx,1
37      mov ecx,n
38      mov edx,1
39      int 0x80
40
41      mov eax,1        ;sys_exit
42      int 0x80         ;interupt
43
44      section .data
45      num db '500'
46      msg db 'after decrement:'
47      len equ $-msg
48      n db '',0xA
49
```

```asm
17
18         pushad          ;to preserve register values
19         call print_num  ;number in eax register
20         popad           ;restore register values
21
22         mov eax,4       ;for new line
23         mov ebx,1
24         mov ecx,n
25         mov edx,1
26         int 0x80
27
28         mov     eax, 1  ;sys_exit
29         int     0x80    ;interupt
30
31     section .data
32         x dw 127        ;operand 1
33         y dw 300        ;operand 2
34         n db '',0xA
35         z db '-'
36         msg db 'The Result after Logical-And is:'
37         len equ $-msg
38
39     section .bss
40     res resb 1          ;result variable
41
```

TERMINAL   PROBLEMS   OUTPUT   DEBUG CONSOLE

```
manas@LAPTOP-U09MU8FI:/mnt/d/assembly_code/week-2$ nasm -f elf q6.asm
manas@LAPTOP-U09MU8FI:/mnt/d/assembly_code/week-2$ ld -m elf_i386 -s -o q6 q6.o
manas@LAPTOP-U09MU8FI:/mnt/d/assembly_code/week-2$ ./q6
after decrement:499
manas@LAPTOP-U09MU8FI:/mnt/d/assembly_code/week-2$ nasm -f elf q7.asm
manas@LAPTOP-U09MU8FI:/mnt/d/assembly_code/week-2$ nasm -f elf q7.asm
manas@LAPTOP-U09MU8FI:/mnt/d/assembly_code/week-2$ ld -m elf_i386 -s -o q7 q7.o
manas@LAPTOP-U09MU8FI:/mnt/d/assembly_code/week-2$ ./q7
The Result after Logical-And is:44
manas@LAPTOP-U09MU8FI:/mnt/d/assembly_code/week-2$
```

```asm
20      popad           ;restore register values
21
22      mov eax,4       ;for new line
23      mov ebx,1
24      mov ecx,n
25      mov edx,1
26      int 0x80
27
28      mov     eax, 1   ;sys_exit
29      int     0x80     ;interupt
30
31  section .data
32      x dw 256            ;operand 1
33      y dw 128            ;operand 2
34      n db '',0xA
35      z db '-'
36      msg db 'The Result after Logical-Or is:'
37      len equ $-msg
38
39  section .bss
40  res resb 1            ;result variable
41
```

```asm
20     popad              ;restore register values
21
22         mov eax,4      ;for new line
23         mov ebx,1
24         mov ecx,n
25         mov edx,1
26         int 0x80
27
28         mov     eax, 1    ;sys_exit
29         int     0x80      ;interupt
30
31 section .data
32     x dw 640             ;operand 1
33     y dw 766              ;operand 2
34     n db '',0xA
35     z db '-'
36     msg db 'The Result after Logical-Xor is:'
37     len equ $-msg
38
39 section .bss
40 res resb 1               ;result variable
41
```

```asm
21        mov eax,4        ;for new line
22        mov ebx,1
23        mov ecx,n
24        mov edx,1
25        int 0x80
26
27        mov     eax, 1    ;sys_exit
28        int     0x80      ;interupt
29
30    section .data
31        x dw 128            ;operand 1
32        n db '',0xA
33        z db '-'
34        msg db 'The result after Left-shift by 3 is:'
35        len equ $-msg
36
37    section .bss
38    res resb 1              ;result variable
39
```

week-2 > q11.asm

```asm
            mov ecx,n
24          mov edx,1
25          int 0x80
26
27          mov     eax, 1     ;sys_exit
28          int     0x80       ;interupt
29
30    section .data
31        x dw 513             ;operand 1
32        n db '',0xA
33        z db '-'
34        msg db 'The result after Right-shift by 2 is:'
35        len equ $-msg
36
37    section .bss
38    res resb 1               ;result variable
39
```

TERMINAL   PROBLEMS   OUTPUT   DEBUG CONSOLE

```
manas@LAPTOP-U09MU8FI:/mnt/d/assembly_code/week-2$ nasm -f elf q11.asm
manas@LAPTOP-U09MU8FI:/mnt/d/assembly_code/week-2$ ld -m elf_i386 -s -o q11 q11.o
manas@LAPTOP-U09MU8FI:/mnt/d/assembly_code/week-2$ ./q11
The result after Right-shift by 2 is:128
manas@LAPTOP-U09MU8FI:/mnt/d/assembly_code/week-2$
```

functions.asm

```asm
print_num:
push edx            ; push in stack for correct order
mov  edx, 0         ; set remainder to zero
cmp  eax, 0         ; check if num is zero
je   done
mov  ebx, 10        ; divide by 10 to get remainder in edx
div  ebx
call print_num
add  eax, '0'       ; convert to ascii digit
mov  [res], edx
mov  ecx, res       ; print digit
mov  edx, 1
mov  eax, 4
mov  ebx, 1
int  0x80
done:
pop  edx            ; pop value since it's no longer needed
ret
```

```
%         Question - 4
%- include "functions.asm"
section.text
global _start
_start:
mov eax, 4          ; print message
mov ebx, 1
mov ecx, msg
mov edx, len
int 0x80


xor dx, dx
mov ax, [x]
mov bx, [y]
div bx              ; divide


pushad
call print_num      ; print quotient
popad

pushad
mov eax, 4                  ; print message
mov ebx, 1
mov ecx, msg2
mov edx, len2
popad popad
                    pu
mov eax, 1          ; exit
int 0x80
```

Manas
COE19B003

```
Section.data
x  dw  900
y  dw  40
msg db 'Quotient:'
len equ $-msg
msg2db 'remainder'
len2 equ $-msg2


Section.bss
res resb 1
```

```
mov ax,dx
pushad              ; print
call print_num      ; remainder
popad
```

Question - # 9   Manas
                                                      COE19B003

```
%. include "functions.asm"
Section. text
global _start
_start:
    mov eax, 4          ; print message
    mov ebx, 1
    mov ecx, msg
    mov edx, len


    mov ax, [x]
    mov bx, [y]
    xor ax, bx          ; perform xor


pushad
call print_num          ; print number
popad


mov eax, 1              ; exit
int 0x80


Section. data
x dw   640
y dw   766
msg db ' logical - xor is '
len equ $-msg
Section. bss
res resb 1
```

Question - 10

```
%include "functions.asm"
section .text
global _start
_start
mov eax, 4          ; print msg
mov ebx, 1
mov ecx, msg
mov edx, len
int 0x80
mov ax, [x]
shl ax, 3           ; left shift by 3
pushad
call print_num      ; print_num
popad
mov eax, 1          ; exit
int 0x80


Section .data
x dw 128
msg db 'The result after lhs by 3'
len equ $ - msg


Section .bss
sum resb 1
```