

OSAMA SAGHEER
BCS211074
SECTION 1

PRACTICE TASK LAB 4

TASK 1

CODE:

```
.model data
```

```
.data
```

```
A db 1
```

```
B db 1
```

```
C db 1
```

```
D db 4
```

```
E db 1
```

```
F db 2
```

```
G db 1
```

```
var1 db ?
```

```
var2 db ?
```

```
var3 db ?
```

```
var4 db ?
```

```
var5 db ?
```

```
msg1 db 0ah,0dh, "A=1:","$"
```

```
msg2 db 0ah,0dh, "B=1:","$"
```

```
msg3 db 0ah,0dh, "C=1:","$"
```

```
msg4 db 0ah,0dh, "D=4:","$"
```

```
msg5 db 0ah,0dh, "E=1:","$"
```

```
msg6 db 0ah,0dh, "F=2:","$"
```

```
msg7 db 0ah,0dh, "G=1:","$"
```

```
res db 0ah,0dh, "A+B+(C+D)+E+(F+G):","$"
```

```
.code
```

```
mov ax,@data
```

```
mov ds,ax
```

```
;(c+d)
```

```
;display msg 3
```

```
lea dx,msg3
```

```
mov ah,9
```

int 21h

;print a msg 4

lea dx,msg4

mov ah,9

int 21h

;store c in al reg

mov al,C

mov bl,al

;store al in bl reg

mov al,D

;store from al to cl reg

mov cl,al

;add bl and cl

add bl,cl

mov var1,bl

mov dl,var1

;add dl,30h

add dl,48

mov ah,02

int 21h

;(f+G)

;display msg 6

lea dx,msg6

mov ah,9

int 21h

;display msg 7

lea dx,msg7

mov ah,9

int 21h

;store c in al reg

mov al,F

;store al in bl reg

mov bl,al

;store al in bl reg

mov al,G

;store from al to cl reg

mov cl,al

;add bl and cl

add bl,cl

mov var2,bl

mov dl,var2

;add dl,30h

add dl,48

mov ah,02

int 21h

```
;display msg 5
lea dx,msg5
mov ah,9
int 21h
;Add var2(F+G) with E
;store c in al reg
mov al,var2
mov bl,al
mov al,E
mov cl,al
add bl,cl
mov var3,bl
mov dl,var3
add dl,48
mov ah,02
int 21h
```

```
;A+B
;display msg 1
lea dx,msg1
mov ah,9
int 21h
;display msg 2
lea dx,msg2
mov ah,9
int 21h
```

```
;store c in al reg
mov al,A
;store al in bl reg
mov bl,al
```


```
;store al in bl reg
mov al,B
;store from al to cl reg
mov cl,al
;add bl and cl
add bl,cl
mov var4,bl
mov dl,var4
;add dl,30h
add dl,48
mov ah,02
int 21h
```

```
;add var1 with var4
```

```
lea dx,res
mov ah,09
int 21h
```

```
mov al,var4
mov bl,al
mov al,var1
mov cl,al
add bl,cl
mov var5,bl
mov dl,var5
add dl,48
mov ah,02
int 21h
```

OUTPUT:



The screenshot shows a window titled "Scit emulator screen (80x25 chars)" with a black background and white text. The text displays the output of the assembly program, showing the values of variables C, D, F, G, E, A, and B, followed by a calculation of A+B+(C+D)+E+(F+G) resulting in 7. The window has standard Windows-style window controls (minimize, maximize, close) in the top right corner. At the bottom of the window, there is a status bar with two buttons: "clear screen" and "change font", and a small display showing "0/16".

```
C=1 :
D=4 : 5
F=2 :
G=1 : 3
E=1 : 4
A=1 :
B=1 : 2
A+B+(C+D)+E+(F+G) : 7
```

TASK 2

CODE:

```
.model small
```

```
.data
```

```
A db 6
```

```
B db 5
```

```
C db 4
```

```
D db 3
```

```
E db 2
```

```
F db 4
```

```
G db 1
```

```
Res db 0
```

```
var1 db 0
```

```
var2 db 0
```

```
.code
```

```
mov ax, @data
```

```
mov ds, ax
```

```
; Calculate (C-D)
```

```
mov al, C
```

```
sub al, D
```

```
mov var1, al
```

```
; Calculate (F-G)
```

```
mov al, F
```

```
sub al, G
```

```
mov var2, al
```

```
; Calculate the final result
```

```
mov al, A
```

```
add al, B
```

```
sub al, var1
```

```
sub al, E
```

```
sub al, var2
```

```
mov Res, al
```

```
; Display the result
```

```
mov dl, Res
```

```
add dl, 48
```

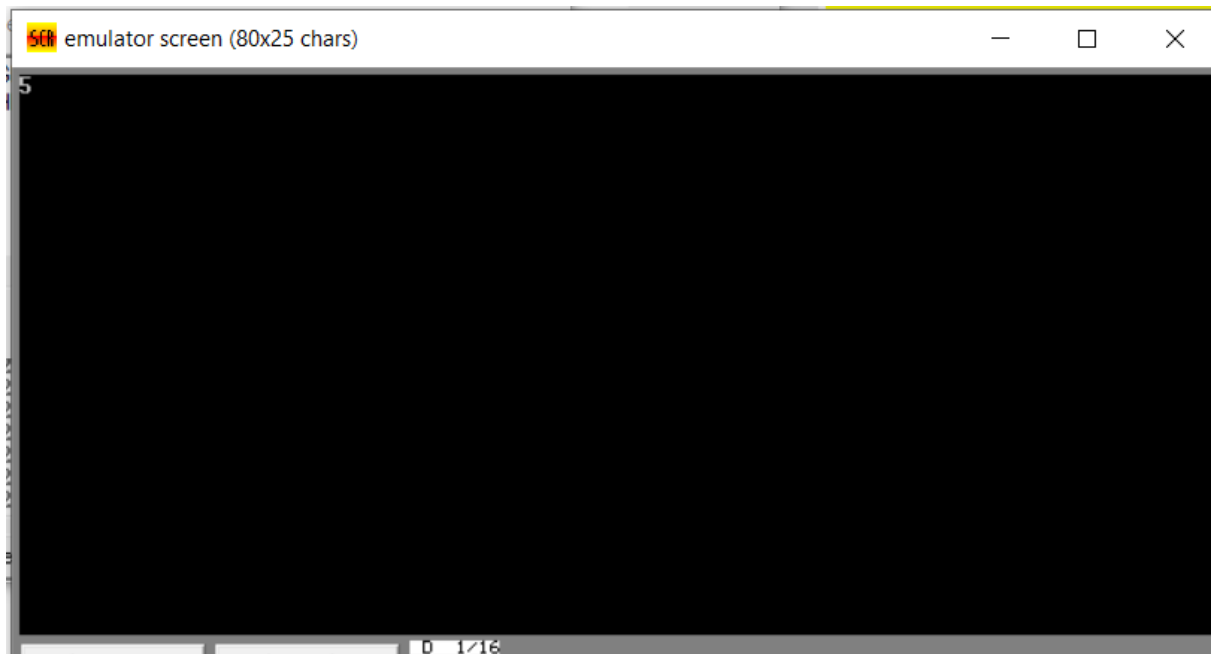
```
mov ah, 02h
```

```
int 21h
```

```
mov ah, 4Ch
```

```
int 21h
```

OUTPUT:



TASK 3

CODE:

```
.model small
.stack 100h
.data
A db 3
B db 2
C db 2
D db 1
E db 1
F db 0
G db 3

Res db ?

msg1 db 0ah,0dh, "A=3:","$"
msg2 db 0ah,0dh, "B=2:","$"
msg3 db 0ah,0dh, "C=2:","$"
msg4 db 0ah,0dh, "D=1:","$"
msg5 db 0ah,0dh, "E=1:","$"
msg6 db 0ah,0dh, "F=0:","$"
msg7 db 0ah,0dh, "G=3:","$"
resmsg db 0ah,0dh, "Res = A + B - (C + D) - E + (F - G) : ","$"

.code
main proc
mov ax,@data
mov ds,ax

;display msg 1
lea dx,msg1
mov ah,9
int 21h

;display msg 2
lea dx,msg2
mov ah,9
int 21h

;display msg 3
lea dx,msg3
mov ah,9
int 21h

;display msg 4
lea dx,msg4
mov ah,9
```

int 21h

```
;display msg 5  
lea dx,msg5  
mov ah,9  
int 21h
```

```
;display msg 6  
lea dx,msg6  
mov ah,9  
int 21h
```

```
;display msg 7  
lea dx,msg7  
mov ah,9  
int 21h
```

```
; calculate Res = A + B - (C + D) - E + (F - G)  
mov al, C  
add al, D  
sub al, F  
add al, G  
mov bl, A  
add bl, B  
sub bl, E  
sub bl, al  
mov R
```

es, bl

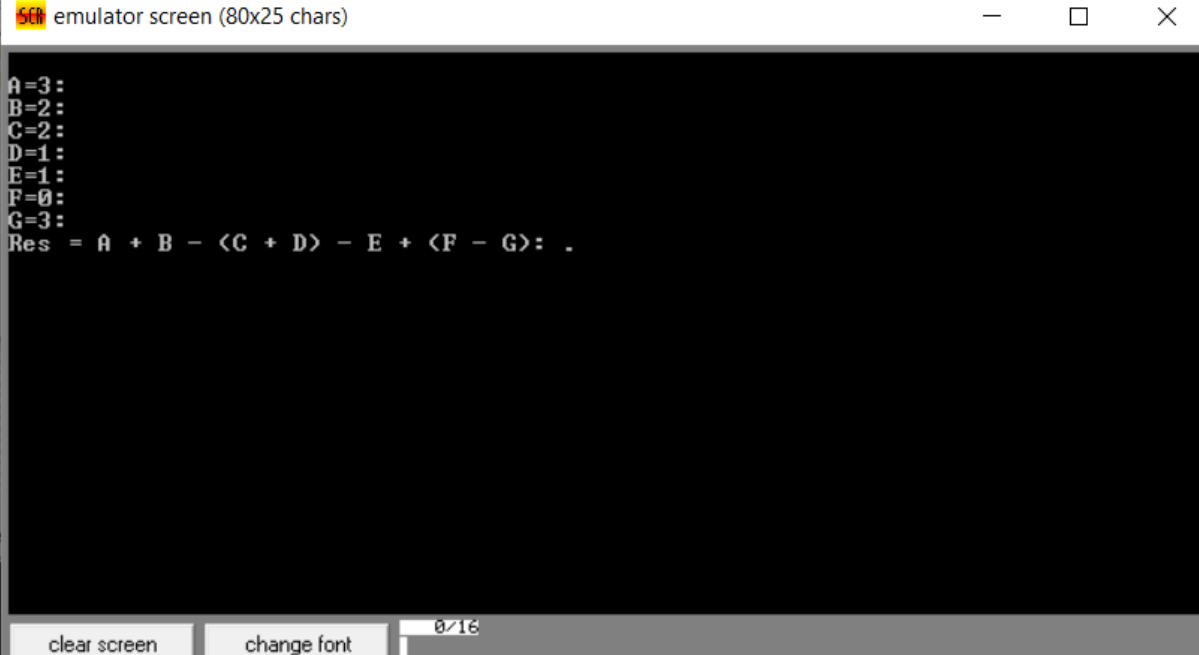
```
;display result  
lea dx, resmsg  
mov ah, 9  
int 21h
```

```
mov dl, Res  
add dl, 48  
mov ah, 02  
int 21h
```

```
mov ah, 4ch  
int 21h
```

```
main endp  
end main
```


OUTPUT:



The screenshot shows a window titled "sc emulator screen (80x25 chars)". The window contains a black area with white text. The text displays variable assignments for A, B, C, D, E, F, and G, followed by a calculation for Res. The calculation uses angle brackets for grouping. At the bottom of the window, there is a grey bar with two buttons: "clear screen" and "change font", and a small display showing "0/16".

```
A=3:
B=2:
C=2:
D=1:
E=1:
F=0:
G=3:
Res = A + B - <C + D> - E + <F - G>: .
```

clear screen change font 0/16

TASK 4

CODE:

.model data

.data

A db 4

B db 1

C db 3

D db 2

E db 4

F db 2

G db 3

Res db ?

msg1 db 0ah,0dh, "A=4:","\$"

msg2 db 0ah,0dh, "B=1:","\$"

msg3 db 0ah,0dh, "C=3:","\$"

msg4 db 0ah,0dh, "D=2:","\$"

msg5 db 0ah,0dh, "E=4:","\$"

msg6 db 0ah,0dh, "F=2:","\$"

msg7 db 0ah,0dh, "G=3:","\$"

resmsg db 0ah,0dh, "Res:","\$"

.code

mov ax, @data

mov ds, ax

; A-B

lea dx, msg1

mov ah, 9

int 21h

lea dx, msg2

mov ah, 9

int 21h

mov al, A

mov bl, B

sub al, bl

mov dl, al

add dl, 48

mov ah, 2

int 21h

; -(C+D)

lea dx, msg3

mov ah, 9

int 21h

lea dx, msg4

mov ah, 9

int 21h

mov al, C
mov bl, D
add al, bl
neg al
mov cl, al

mov al, cl
mov bl, 1
add al, bl
mov dl, al
add dl, 48
mov ah, 2
int 21h

; +E
lea dx, msg5
mov ah, 9
int 21h

mov al, E
add dl, al
mov ah, 2
int 21h

; -(F-G)
lea dx, msg6
mov ah, 9
int 21h

lea dx, msg7
mov ah, 9
int 21h

mov al, F
mov bl, G
sub al, bl
neg al
mov cl, al

mov al, cl
mov bl, 1
sub al, bl
add dl, al
mov ah, 2
int 21h

; Store result in Res
lea dx, resmsg
mov ah, 9
int 21h

```
mov al, dl
sub al, 48
mov Res, al
```

```
mov ah, 4ch
int 21h
```

OUTPUT:

