



**SHAHEED ZULFIKAR ALI BHUTTO**  
INSTITUTE OF SCIENCE AND TECHNOLOGY

# Computer Organization and Assembly Language

## Laboratory Manual

CSCL-2201

Semester 3<sup>rd</sup>

Fall-2022

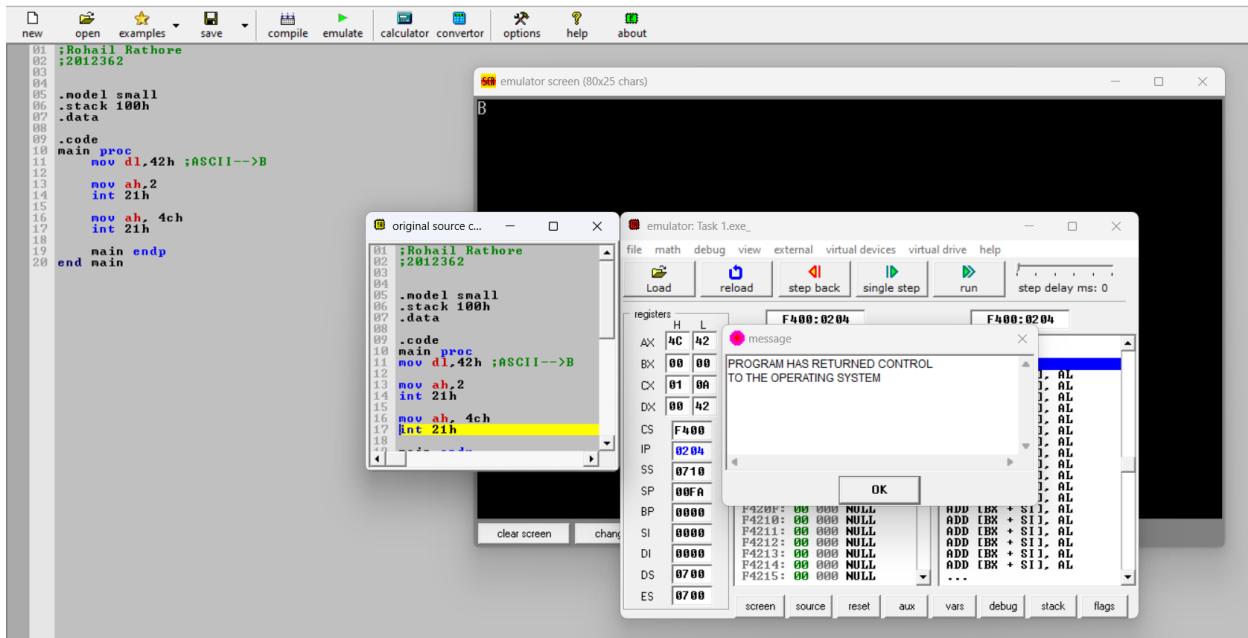
Student's Name: Rohail Rathore

Roll Number: 2012362

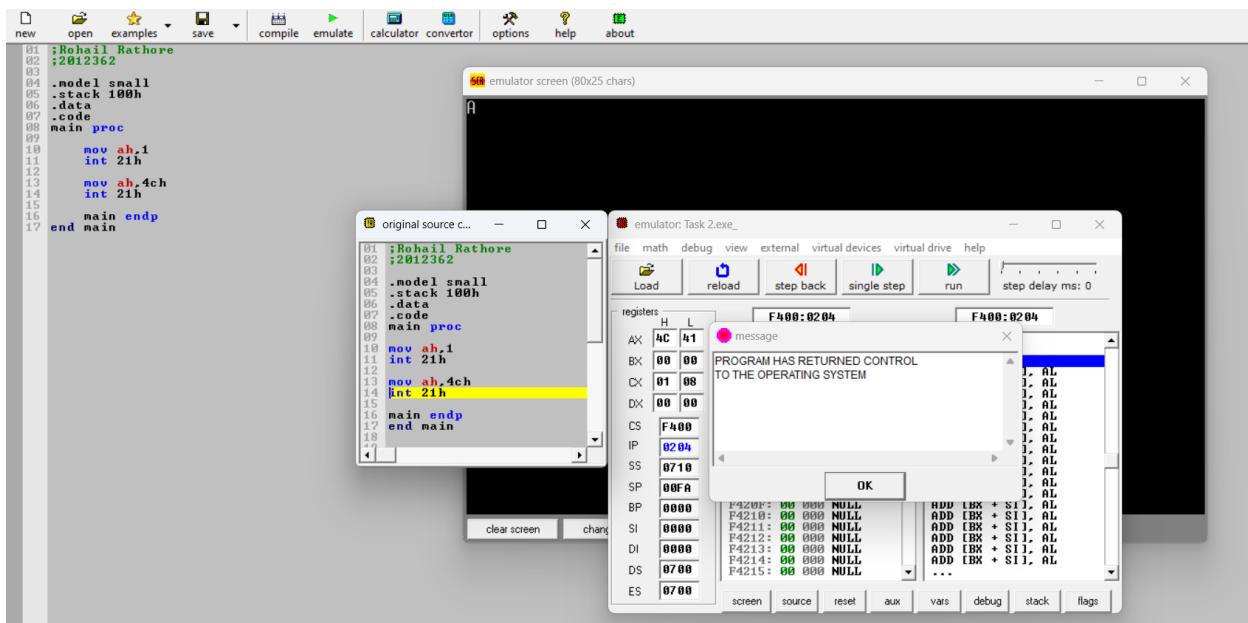
Lab Instructor: Engr. Mubeen Ahmed Khan

Department of Computing Sciences  
Shaheed Zulfiqar Ali Bhutto Institute Of Science And Technology,Karach  
Tel:(021)111-922-478, Fax: (021)35830446, Email: info@szabist.edu.pk

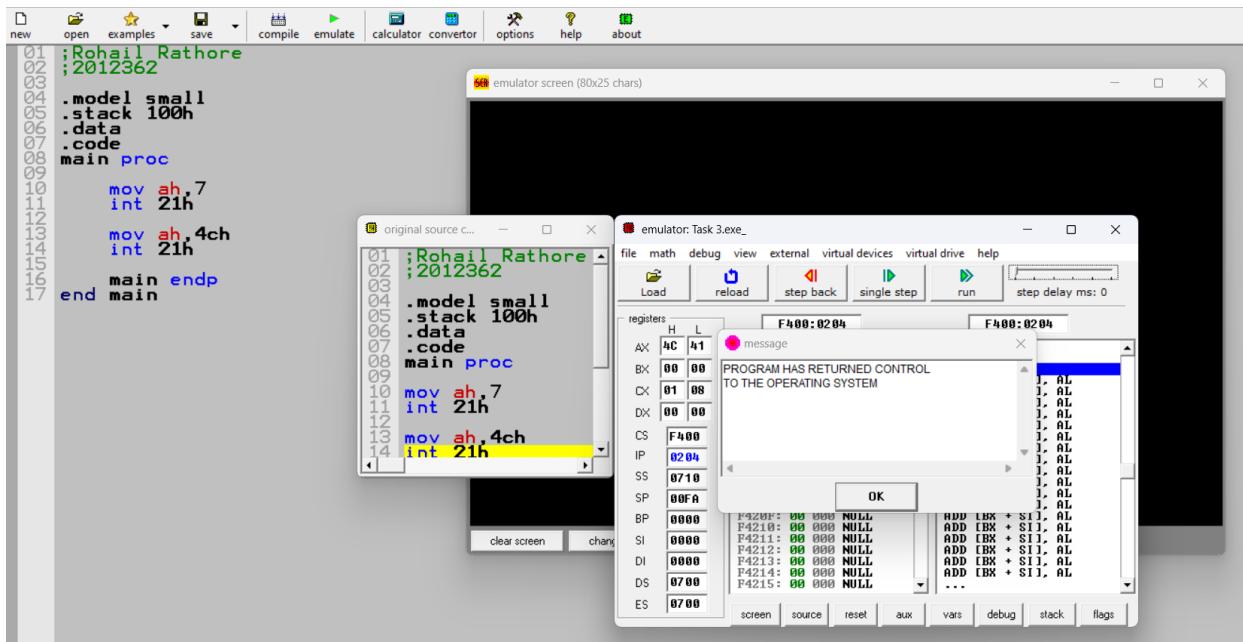
## TASK-1:



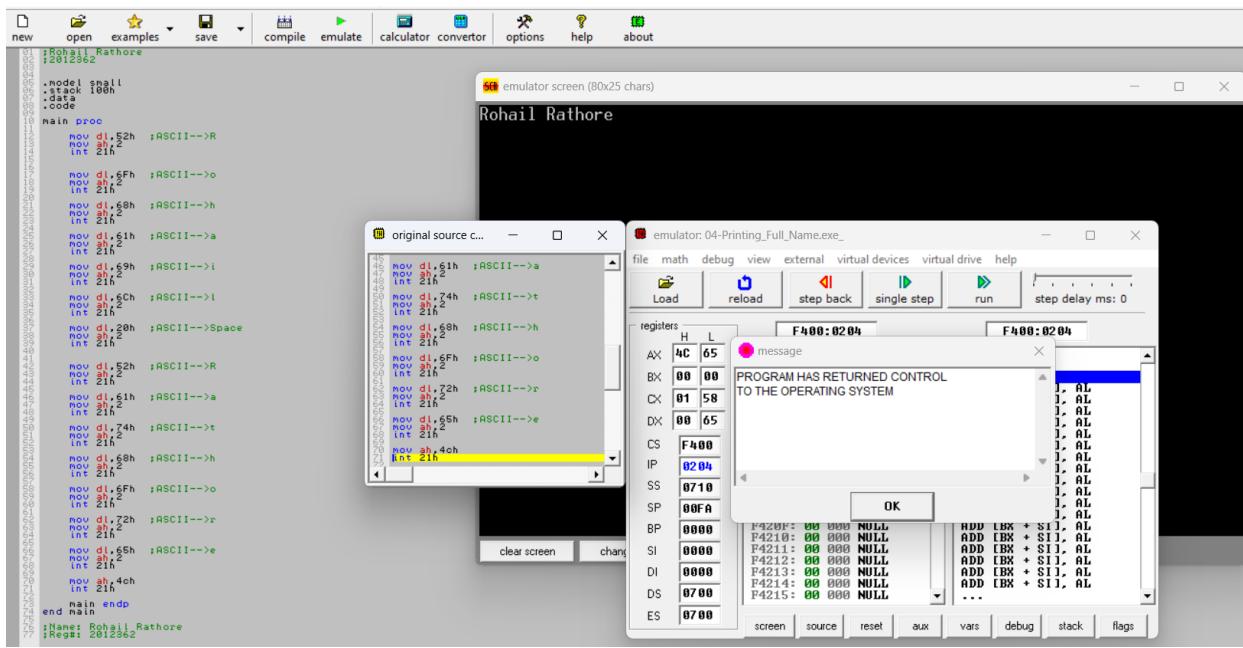
## Task-2:



### Task-3:



### Task-4:



### Task-5:

```

01 Rohail Rathore
02 ;2012362
03
04 .model small
05 .stack 100h
06 .data
07 .code
08 main proc
09     mov ah, 7
10    int 21h
11
12    mov dl, 2Ah
13    mov ah, 2
14    int 21h
15
16    mov ah, 7
17    int 21h
18
19    mov dl, 2Ah
20    mov ah, 2
21    int 21h
22
23    mov ah, 7
24    int 21h
25
26    mov dl, 2Ah
27    mov ah, 2
28    int 21h
29
30    mov ah, 7
31    int 21h
32
33    mov dl, 2Ah
34    mov ah, 2
35    int 21h
36
37    mov ah, 4ch
38    int 21h
39
40 end main

```

## Task-6:

```

01 Rohail Rathore
02 ;2012362
03
04 .model small
05 .stack 100h
06 .data
07
08 .code
09 main proc
10     mov ah, 1           ;User Input
11     int 21h
12     mov cl, al
13
14     mov dl, 10          ;Newline
15     mov ah, 2
16     int 21h
17
18     mov dl, cl          ;Backspace
19     add dl, 32
20     mov ah, 2
21     int 21h
22
23     mov ah, 4ch
24     int 21h
25
26 end main

```

## Task-7:

edit: C:\My Files\Szabist\5th Semester\COAL-LAB\Programs\06-Convertin-U-Lasp

new open examples save compile emulate calculator

```

01 ;Rohail Rathore
02 ;2012362
03
04 .model small
05 .stack 100h
06 .data
07
08
09 .code
10 main proc
11     mov ah,1           ;User Input
12     int 21h
13     mov cl,al
14
15     mov dl,10
16     mov ah,2           ;Newline
17     int 21h
18
19     mov dl,13
20     mov ah,2           ;Backspace
21     int 21h
22
23     mov dl,cl
24     sub dl,32
25     mov ah,2
26     int 21h
27
28     mov ah,4ch
29     int 21h
30
31 main endp
32
33 end main

```

emulator screen (80x25 chars)

original source c... -

F400:0204 F400:0204

message

PROGRAM HAS RETURNED CONTROL  
TO THE OPERATING SYSTEM

OK

Registers:

AX	4C	41
BX	00	00
CX	01	61
DX	00	41
CS	F400	
IP	0204	
SS	0710	
SP	00FA	
BP	0000	
SI	0000	
DI	0000	
DS	0700	
ES	0700	

Memory dump:

```

F420H: 00 0000 NULL ADD LBX + $1, AL
F421H: 00 0000 NULL ADD LBX + $1, AL
F4211: 00 0000 NULL ADD LBX + $1, AL
F4212: 00 0000 NULL ADD LBX + $1, AL
F4213: 00 0000 NULL ADD LBX + $1, AL
F4214: 00 0000 NULL ADD LBX + $1, AL
F4215: 00 0000 NULL ...

```

Buttons: screen, source, reset, aux, vars, debug, stack, flags

## Task-8:

edit: C:\My Files\Szabist\5th Semester\COAL-LAB\Programs\Task 8

new open examples save compile emulate calculator convertor options help about

```

01 ;Rohail Rathore
02 ;2012362
03
04 .model small
05 .stack 100h
06 .data
07
08 .code
09 main proc
10     mov ah,1
11     int 21h
12
13     sub al,48
14     mov ch,al
15
16     mov ah,1
17     int 21h
18
19     sub al,48
20     add ch,al
21
22     mov dl,ch
23     add dl,48
24
25     mov ah,2
26     int 21h
27
28     mov ah,4ch
29     int 21h
30
31 main endp
32
33 end main

```

emulator screen (80x25 chars)

original source c... -

F400:0204 F400:0204

message

PROGRAM HAS RETURNED CONTROL  
TO THE OPERATING SYSTEM

OK

Registers:

AX	4C	35
BX	00	00
CX	05	1D
DX	00	35
CS	F400	
IP	0204	
SS	0710	
SP	00FA	
BP	0000	
SI	0000	
DI	0000	
DS	0700	
ES	0700	

Memory dump:

```

F420H: 00 0000 NULL ADD LBX + $1, AL
F421H: 00 0000 NULL ADD LBX + $1, AL
F4211: 00 0000 NULL ADD LBX + $1, AL
F4212: 00 0000 NULL ADD LBX + $1, AL
F4213: 00 0000 NULL ADD LBX + $1, AL
F4214: 00 0000 NULL ADD LBX + $1, AL
F4215: 00 0000 NULL ...

```

Buttons: screen, source, reset, aux, vars, debug, stack, flags

## Task-9:

The screenshot shows a debugger interface with several windows:

- Assembly Editor:** Shows the assembly code for a program named "Rohail Rathore ;2012362". The code performs arithmetic operations (addition and subtraction) and prints results to the screen.
- Emulator Screen:** Displays the output of the program, showing results like "3+2=5" and "5-3=2".
- Registers Window:** Shows the state of CPU registers (AX, BX, CX, DX, SI, DI, BP, SP, CS, IP, SS, DS, ES) at address F400:0204.
- Memory Dump Window:** Shows memory starting at address F4210, containing multiple entries of "ADD BX + S1, AL".
- Message Box:** A modal dialog box with the message "PROGRAM HAS RETURNED CONTROL TO THE OPERATING SYSTEM" and an "OK" button.

## Task-10:

The screenshot shows a debugger interface with several windows:

- Assembly Editor:** Shows the assembly code for a program named "Rohail Rathore ;2012362". The code performs arithmetic operations (addition and subtraction) and prints results to the screen.
- Emulator Screen:** Displays the output of the program, showing results like "3+2=5" and "5-3=2".
- Registers Window:** Shows the state of CPU registers (AX, BX, CX, DX, SI, DI, BP, SP, CS, IP, SS, DS, ES) at address F400:0204.
- Memory Dump Window:** Shows memory starting at address F4210, containing multiple entries of "ADD BX + S1, AL".
- Message Box:** A modal dialog box with the message "PROGRAM HAS RETURNED CONTROL TO THE OPERATING SYSTEM" and an "OK" button.

## Task-11:

The screenshot shows a debugger interface with several windows:

- Assembly Editor:** Shows the assembly code for a program named "Rohail Rathore\_2012362". The code initializes three variables (num1, num2, num3) to 1, 2, and 3 respectively, adds them together, and prints the result (6) to the screen.
- Emulator Screen:** Displays the character "6" at the top left.
- Registers Window:** Shows CPU register values at address F400:0204. AX contains 36, BX contains 00, CX contains 32, DX contains 36, CS contains F400, IP contains 0204, SS contains 0710, SP contains 00FA, BP contains 0000, SI contains 0000, DI contains 0000, DS contains 0700, and ES contains 0700.
- Message Box:** A modal dialog box displays the message "PROGRAM HAS RETURNED CONTROL TO THE OPERATING SYSTEM".
- Memory Dump:** A window titled "F400:0204" shows memory starting at address F400:0204. It lists multiple entries of "ADD EBX + S1, AL".

## Task-12:

The screenshot shows a debugger interface with several windows:

- Assembly Editor:** Shows the assembly code for a program named "Rohail Rathore\_2012362". The code initializes two variables (al and ah) to 1 and 2 respectively, adds them together, and prints the result (12) to the screen.
- Emulator Screen:** Displays the character "12" at the top left.
- Registers Window:** Shows CPU register values at address F400:0204. AX contains 32, BX contains 00, CX contains 16, DX contains 32, CS contains F400, IP contains 0204, SS contains 0710, SP contains 00FA, BP contains 0000, SI contains 0000, DI contains 0000, DS contains 0700, and ES contains 0700.
- Message Box:** A modal dialog box displays the message "PROGRAM HAS RETURNED CONTROL TO THE OPERATING SYSTEM".
- Memory Dump:** A window titled "F400:0204" shows memory starting at address F400:0204. It lists multiple entries of "ADD EBX + S1, AL".

## Task-13:

The screenshot shows a debugger interface with two windows. The left window displays assembly code:

```

01 ;Rohail Rathore
02 ;2012362
03
04 .model small
05 .stack 100h
06 .data
07 .code
08 main proc
09
10     mov al,2
11     add al,48
12
13     mov dl,al
14     mov ah,2
15     int 21h
16
17     dec al
18
19     mov dl,al
20     mov ah,2
21     int 21h
22
23     mov ah,4ch
24     int 21h
25
26     mov ah,4ch
27     int 21h
28
29 main endp
30 end main

```

The right window shows the emulator screen displaying the message "21". Below it, a message box says "PROGRAM HAS RETURNED CONTROL TO THE OPERATING SYSTEM" with an "OK" button. The registers window shows the following values:

	H	L
AX	AC	31
BX	00	00
CX	01	16
DX	00	31
CS	F400	
IP	0204	
SS	0710	
SP	00FA	
BP	0000	
SI	0000	
DI	0000	
DS	0700	
ES	0700	

## Task-14:

The screenshot shows a debugger interface with two windows. The left window displays assembly code:

```

01 .model small
02 .stack 100h
03 .data
04 msg1 db 'Hello World$'
05
06 .code
07 main proc
08
09     mov ax,@data
10     mov ds,ax
11
12     mov dx,offset msg1
13     mov ah,9
14     int 21h
15
16     mov ah,4ch
17     int 21h
18
19 main endp
20
21 end main

```

The right window shows the emulator screen displaying the message "Hello World". Below it, a message box says "PROGRAM HAS RETURNED CONTROL TO THE OPERATING SYSTEM" with an "OK" button. The registers window shows the following values:

	H	L
AX	AC	24
BX	00	00
CX	01	20
DX	00	00
CS	F400	
IP	0204	
SS	0710	
SP	00FA	
BP	0000	
SI	0000	
DI	0000	
DS	0720	
ES	0700	

## Task-15:

The screenshot shows a debugger interface with several windows:

- Assembly Editor:** Displays the assembly code for a program named "15.exe". The code includes data definitions (.db), stack setup (.stack), and various procedures (main, newline, dx\_offset). A specific instruction at address F400:0204, `mov ah,4ch`, is highlighted.
- Registers Window:** Shows the CPU register state. AX is 4C, BX is 00, CX is 01, DX is 00, CS is F400, IP is 0204, SS is 0710, SP is 00FA, BP is 0000, SI is 0000, DI is 0000, DS is 0720, and ES is 0700.
- Memory Dump Window:** Shows memory starting at address F420F. The memory contains multiple instances of the byte 00, followed by NULL (00 00) and ADD BX + SI, AL (ADD BX + \$1, AL).
- Message Window:** A modal dialog box displays the message "PROGRAM HAS RETURNED CONTROL TO THE OPERATING SYSTEM".

## Task-16:

The screenshot shows a debugger interface with several windows:

- Assembly Editor:** Displays the assembly code for a program named "Task 16.exe". The code includes data definitions (.db), stack setup (.stack), and various procedures (main, newline, loop1). A specific instruction at address F400:0204, `mov ah,4ch`, is highlighted.
- Registers Window:** Shows the CPU register state. AX is 4C, BX is 00, CX is 00, DX is 00, CS is F400, IP is 0204, SS is 0710, SP is 00FA, BP is 0000, SI is 0000, DI is 0000, DS is 0700, and ES is 0700.
- Memory Dump Window:** Shows memory starting at address F420F. The memory contains multiple instances of the byte 00, followed by NULL (00 00) and ADD BX + SI, AL (ADD BX + \$1, AL).
- Message Window:** A modal dialog box displays the message "PROGRAM HAS RETURNED CONTROL TO THE OPERATING SYSTEM".

## Task-17:

The screenshot shows a debugger interface with the following components:

- Assembly Editor:** Displays the assembly code for Task-17. The code initializes CX to 10, loops 10 times, and then prints the value of DL (which is 48) to the screen.
- Registers Window:** Shows the CPU register values at address F400:0204. AX has H: 4C and L: 39. BX, CX, and DX are 00 00. CS is F400, IP is 0204, SS is 0710, SP is 00FA, BP is 0000, SI is 0000, DI is 0000, DS is 0700, and ES is 0700.
- Memory Dump Window:** Shows memory starting at address F4200. It displays a series of ADD instructions where EBX is added to SI, resulting in AL.
- Output Window:** Shows the output "0123456789" on the screen.

## Task-18:

The screenshot shows a debugger interface with the following components:

- Assembly Editor:** Displays the assembly code for Task-18. The code initializes CX to 10, loops 10 times, and then prints the value of DL (which is 57) to the screen.
- Registers Window:** Shows the CPU register values at address F400:0204. AX has H: 4C and L: 38. BX, CX, and DX are 00 00. CS is F400, IP is 0204, SS is 0710, SP is 00FA, BP is 0000, SI is 0000, DI is 0000, DS is 0700, and ES is 0700.
- Memory Dump Window:** Shows memory starting at address F4200. It displays a series of ADD instructions where EBX is added to SI, resulting in AL.
- Output Window:** Shows the output "9876543210" on the screen.

## Task-19:

The screenshot shows a debugger interface with several windows:

- Assembly Editor:** Displays the assembly code for a program named "Task 19.exe". The code includes instructions like `mov cx, 26` and `int 21h`.
- Emulator Screen:** Shows the character set "ABCDEFGHIJKLMNOPQRSTUVWXYZ" displayed on the screen.
- Registers Window:** Shows the state of various CPU registers:
 

	H	L
AX	4C	5A
BX	00	00
CX	00	00
DX	00	5B
CS	F400	
IP	02 04	
SS	0710	
SP	00FA	
BP	0000	
SI	0000	
DI	0000	
DS	0700	
ES	0700	
- Message Window:** Displays a message box with the text "PROGRAM HAS RETURNED CONTROL TO THE OPERATING SYSTEM" and an "OK" button.

## Task-20:

The screenshot shows a debugger interface with several windows:

- Assembly Editor:** Displays the assembly code for a program named "Task 20.exe". The code includes instructions like `mov cx, 26` and `int 21h`.
- Emulator Screen:** Shows the character set "abcdefghijklmnopqrstuvwxyz" displayed on the screen.
- Registers Window:** Shows the state of various CPU registers:
 

	H	L
AX	4C	7A
BX	00	00
CX	00	00
DX	00	7D
CS	F400	
IP	02 04	
SS	0710	
SP	00FA	
BP	0000	
SI	0000	
DI	0000	
DS	0700	
ES	0700	
- Message Window:** Displays a message box with the text "PROGRAM HAS RETURNED CONTROL TO THE OPERATING SYSTEM" and an "OK" button.

## Task-21:

The screenshot shows a debugger interface with the following components:

- Left Panel (Assembly Editor):** Displays assembly code for Task-21. The code includes a main loop that prints the string "ZYXWVUTSRQPONMLKJIHGFE" to the screen. The assembly instructions shown are:
 

```

      11 mov cx,26
      12 loop1:
      13 mov ah,2
      14 int 21h
      15 dec dl
      16 loop loop1
      17 mov ah,4ch
      18 int 21h
      19
      20
      21
      22
      23
      24
      25
      26 end main
      
```
- Middle Panel (Registers View):** Shows the CPU registers in hex format. The AX register contains 4C (AL=1). Other registers like BX, CX, DX, IP, and SS are set to 0000. CS is F400, and SP is 00FA.
 

	H	L
AX	4C	1
BX	00	00
CX	00	00
DX	00	40
CS	F400	
IP	0204	
SS	0710	
SP	00FA	
BP	0000	
SI	0000	
DI	0000	
DS	0700	
ES	0700	
- Right Panel (Memory Dump):** Displays memory starting at address F400:0204. The dump shows a series of null bytes (00) followed by the ASCII string "ZYXWVUTSRQPONMLKJIHGFE". An "OK" button is visible at the bottom right of the dump window.

## Task-22:

The screenshot shows a debugger interface with the following components:

- Left Panel (Assembly Editor):** Displays assembly code for Task-22. The code includes a main loop that prints the string "2468" to the screen. The assembly instructions shown are:
 

```

      08 mov cx,4
      09 mov dx,48
      10 L1:
      11 add dx,2
      12 mov ah,2
      13 int 21h
      14 loop L1
      15 mov ah,4ch
      16 int 21h
      17
      18
      19
      20
      21
      22
      23
      24
      25
      26 end main
      
```
- Middle Panel (Registers View):** Shows the CPU registers in hex format. The AX register contains 4C (AL=38). Other registers like BX, CX, DX, IP, and SS are set to 0000. CS is F400, and SP is 00FA.
 

	H	L
AX	4C	38
BX	00	00
CX	00	00
DX	00	38
CS	F400	
IP	0204	
SS	0710	
SP	00FA	
BP	0000	
SI	0000	
DI	0000	
DS	0700	
ES	0700	
- Right Panel (Memory Dump):** Displays memory starting at address F400:0204. The dump shows a series of null bytes (00) followed by the ASCII string "2468". An "OK" button is visible at the bottom right of the dump window.

## Task-23:

Rohail Rathore  
2012362

```

01:Rohail Rathore
02:2012362

03:model small
04.stack 100h
05.data
06.code

07.main proc
08.
09    mov cx,5
10   OuterLoop:
11.
12   mov bx,cx
13.
14   mov dl,'#'
15   mov ah,2
16   int 21h
17.
18   mov cx,3
19   InnerLoop:
20.
21   mov dl,'*'
22   int 21h
23.
24   loop InnerLoop
25.
26   mov cx,bx
27   loop OuterLoop
28.
29   mov ah,4ch
30   int 21h
31.
32 main endp
33.
34 end main
35.
36

```

emulator screen (80x25 chars)

emulator: Task 23.exe\_

original source c... file math debug view external virtual devices virtual drive help

Load reload step back single step run step delay ms: 0

Registers:

	H	L
AX	4C	2A
BX	00	01
CX	00	00
DX	00	20
CS	F400	
IP	0204	
SS	0710	
SP	00FA	
BP	0000	
SI	0000	
DI	0000	
DS	0700	
ES	0700	

F400:0204 F400:0204

message: PROGRAM HAS RETURNED CONTROL TO THE OPERATING SYSTEM

OK

Screen | Source | Reset | Aux | Vars | Debug | Stack | Flags

## Task-24:

Rohail Rathore  
2012362

```

01:Rohail Rathore
02:2012362

03:model small
04.stack 100h
05.data
06.star dw 1
07.code

08.main proc
09.
10   mov cx,@data
11   mov ds,cx
12.
13   mov cx,5
14   OuterLoop:
15.
16   mov bx,cx
17   mov cx,star
18.
19   InnerLoop:
20.
21   mov dl,'*'
22   mov ah,2
23   int 21h
24.
25   loop InnerLoop
26.
27   inc star
28.
29   mov dl,10
30   mov ah,2
31   int 21h
32.
33   mov dl,13
34   mov ah,2
35   int 21h
36.
37   mov cx,bx
38   loop OuterLoop
39.
40 main endp
41.
42 end main
43.
44

```

emulator screen (80x25 chars)

emulator: Task 24.exe\_

original source c... file math debug view external virtual devices virtual drive help

Load reload step back single step run step delay ms: 0

Registers:

	H	L
AX	4C	0D
BX	00	01
CX	00	00
DX	00	0D
CS	F400	
IP	0204	
SS	0710	
SP	00FA	
BP	0000	
SI	0000	
DI	0000	
DS	0720	
ES	0700	

F400:0204 F400:0204

message: PROGRAM HAS RETURNED CONTROL TO THE OPERATING SYSTEM

OK

Screen | Source | Reset | Aux | Vars | Debug | Stack | Flags

## Task-25:

The screenshot shows a debugger interface with several windows:

- Assembly Window:** Displays the assembly code for Task 25. The code initializes a stack at 100h, defines a data segment with arr1 containing values 1, 2, 3, 4, 5, and implements a loop to print these values using INT 21h. The instruction at address F400:0204 is highlighted.
- Registers Window:** Shows the CPU register state. AX contains 4C, BX contains 00, CX contains 01, DX contains 35, CS contains F400, IP contains 0204, SS contains 0710, SP contains 00FA, BP contains 0000, SI contains 0000, DI contains 0000, DS contains 0720, and ES contains 0700.
- Memory Dump Window:** Shows memory starting at address F4200. It displays a series of ADD IBX + SI, AL instructions, followed by an OK dialog box.
- Output Window:** Shows the output "15" from the program execution.

## Task-26:

The screenshot shows a debugger interface with several windows:

- Assembly Window:** Displays the assembly code for Task 26. It is similar to Task 25 but includes a printLoop label and an additional loop counter CX initialized to 5. The instruction at address F400:0204 is highlighted.
- Registers Window:** Shows the CPU register state. AX contains 4C, BX contains 00, CX contains 05, DX contains 35, CS contains F400, IP contains 0204, SS contains 0710, SP contains 00FA, BP contains 0000, SI contains 0000, DI contains 0000, DS contains 0720, and ES contains 0700.
- Memory Dump Window:** Shows memory starting at address F4200. It displays a series of ADD IBX + SI, AL instructions, followed by an OK dialog box.
- Output Window:** Shows the output "12345" from the program execution.

## Task-27:

**Task-27:**

The screenshot shows a debugger interface with several windows. The main window displays assembly code for a program named 'Task 27.exe'. The code includes a main loop and an exit section. The registers window shows the state of various CPU registers. A message box is displayed, stating 'PROGRAM HAS RETURNED CONTROL TO THE OPERATING SYSTEM' with an 'OK' button.

```

01; Rohail Rathore
02; 2012362
03
04.model small
05.stack 100h
06.data
07
08arr db 20 dup('$')
09
10.code
11main proc
12    mov ax,@data
13    mov ds,ax
14
15    mov bx, offset arr
16    inputLoop:
17        mov ah,1
18        int 21h
19        cmp al,13
20        je exit
21
22        mov [bx],al
23        inc bx
24
25        jmp inputLoop
26
27exit:
28    mov ah,4ch
29    int 21h
30
31    main endp
32
33end main

```

## Task-28:

The screenshot shows a debugger interface with several windows. The main window displays assembly code for a program named 'Task 28.exe'. The code includes a main loop and an exit section. The registers window shows the state of various CPU registers. A message box is displayed, stating 'PROGRAM HAS RETURNED CONTROL TO THE OPERATING SYSTEM' with an 'OK' button.

```

01; Rohail Rathore
02; 2012362
03
04.model small
05.stack 100h
06.data
07
08
09.code
10main proc
11    mov ah,1
12    int 21h
13    push ax
14
15    mov ah,1
16    int 21h
17    push ax
18
19    mov dl,10
20    mov ah,2
21    int 21h
22
23    mov dl,13
24    mov ah,2
25    int 21h
26
27    pop dx
28    mov ah,2
29    int 21h
30
31    pop dx
32    mov ah,2
33    int 21h
34
35    mov ah,4ch
36    int 21h
37
38    main endp
39
40end main

```

## Task-29:

The screenshot shows a debugger interface with the following components:

- Assembly Editor:** On the left, displaying assembly code for a program named "Task 29.exe". The code includes sections for .model small, .stack 100h, .data, .code, main proc, and end main. It contains instructions like mov ax, @data, lea si, msg, mov cx, 11, loop pushLoop, and int 21h.
- Registers Window:** Shows CPU register values at address F400:0204. AX: 4C 48, BX: 00 64, CX: 00 00, DX: 65 48, CS: F400, IP: 0204, SS: 0710, SP: 00FA, BP: 0000, SI: 0000, DI: 0000, DS: 0720, ES: 0700.
- Memory Dump:** A window titled "emulator: Task 29.exe" showing memory starting at address F400:0204. It displays a series of INT 21h calls (F4200-F4215) followed by a stack of AL values (00, 01, 02, 03, 04, 05, 06, 07, 08, 09, 0A, 0B, 0C, 0D, 0E, 0F).
- Output Window:** Shows the string "dlrow olleh" in the emulator screen window.

## Task-30:

The screenshot shows a debugger interface with the following components:

- Assembly Editor:** On the left, displaying assembly code for a program named "Task 30.exe". The code includes sections for .model small, .stack 100h, .data, .code, main proc, input proc, output proc, and end main. It contains instructions like call input, call output, mov ah, 4ch, int 21h, and mov dl, al.
- Registers Window:** Shows CPU register values at address F400:0204. AX: 4C 42, BX: 00 00, CX: 01 16, DX: 00 42, CS: F400, IP: 0204, SS: 0710, SP: 00FA, BP: 0000, SI: 0000, DI: 0000, DS: 0700, ES: 0700.
- Memory Dump:** A window titled "emulator: Task 30.exe" showing memory starting at address F400:0204. It displays a series of INT 21h calls (F4200-F4215) followed by a stack of AL values (00, 01, 02, 03, 04, 05, 06, 07, 08, 09, 0A, 0B, 0C, 0D, 0E, 0F).
- Output Window:** Shows the string "BB" in the emulator screen window.

## Task-31:

The screenshot shows a debugger interface with several windows:

- Assembly Editor:** Shows the assembly code for a program named "Task 31". It includes sections like .model small, .stack 100h, .data, .code, and main proc. The code uses instructions like mov ah, lea dx, call display, and int 21h.
- Registers Window:** Displays CPU registers (AX, BX, CX, DX, CS, IP, SS, SP, BP, SI, DI, DS, ES) with their current values.
- Stack Window:** Shows the stack contents, with a message box overlaid stating "PROGRAM HAS RETURNED CONTROL TO THE OPERATING SYSTEM".
- Memory Dump Window:** Displays memory starting at address F400:0204.
- Output Window:** Shows the emulator screen output: "Name: Rohail Rathore", "Reg No. 2012362", "Institute: SZABIST", and "Email: Rohail@skiff.com".

## Task-32:

The screenshot shows a debugger interface with several windows:

- Assembly Editor:** Shows the assembly code for a program named "Task 32". It includes sections like .model small, .stack 100h, .data, .code, and main proc. The code uses instructions like mov ah, lea dx, call display, and various conditional jumps (je, jne, jz, jnz) based on character comparisons.
- Registers Window:** Displays CPU registers with their current values.
- Stack Window:** Shows the stack contents.
- Memory Dump Window:** Displays memory starting at address F400:0204.
- Output Window:** Shows the emulator screen output: "Activate Windows Go to Settings to activate Windows".

edit: C:\My Files\Szabist\5th Semester\COAL-LAB\LAB-Manual\fwdcoollabtask\Task-32.asm

file edit bookmarks assembler emulator math ascii codes help

emulator screen (80x25 chars)

```

060     displayCount:
061     call newline
062     lea dx,msg2
063     call display
064
065     mov dl,vowels
066     mov ah,2
067     int 21h
068
069     call newline
070     lea dx,msg3
071     call display
072
073     mov dl,consonants
074     mov ah,2
075     int 21h
076
077     mov vowels,'0'
078     mov consonants,'0'
079
080     runAgain:
081     call newline
082     lea dx,msg4
083     call display
084
085     mov ah,1
086     int 21h
087     mov bl,al
088     call newline
089
090     cmp bl,'Y'
091     je input
092
093     cmp bl,'27'
094     je exit
095
096     jmp runAgain
097
098     exit:
099     mov ah,4ch
100     int 21h
101
102     main endp
103
104     display proc
105     mov ah,9
106     int 21h
107     ret
108     display endp
109
110     newline proc
111     mov dl,10
112     mov ah,2
113     int 21h
114
115     mov ah,13
116     mov ah,2
117     int 21h

```

Enter a statement: HELLO  
Total Vowels: 2  
Total Consonants: 3  
Press Y to run again or ESC key to terminate: t  
Press Y to run again or ESC key to terminate: y  
Press Y to run again or ESC key to terminate: y  
Press Y to run again or ESC key to terminate: Y  
Enter a statement: HELLO WORLD  
Total Vowels: 3  
Total Consonants: 7  
Press Y to run again or ESC key to terminate: +

original source c... file math debug view external virtual devices virtual drive help

emulator: Task 32.exe\_

Load message OK

PROGRAM HAS RETURNED CONTROL TO THE OPERATING SYSTEM

AX	4C	0D
BX	00	1B
CX	02	20
DX	00	0D
CS	F400	
IP	0204	
SS	0710	
SP	00FA	
BP	0000	
SI	0000	
DI	0000	
DS	0720	
ES	0700	

screen source reset aux vars debug stack flags

## Task-33:

edit: C:\My Files\Szabist\5th Semester\COAL-LAB\LAB-Manual\fwdcoollabtask\Task-33.asm

file edit bookmarks assembler emulator math ascii codes help

emulator screen (80x25 chars)

```

01 ;Rohail Rathore
02 ;2012362
03
04 .model small
05 .stack 100h
06 .data
07 .code
08
09 main proc
10
11     mov ah,1
12     int 21h
13     or al,00100000b
14
15     mov dl,al
16     mov ah,2
17     int 21h
18
19     mov ah,4ch
20     int 21h
21
22     main endp
23 end main

```

original source c... file math debug view external virtual devices virtual drive help

emulator: Task 33.exe\_

Load message OK

PROGRAM HAS RETURNED CONTROL TO THE OPERATING SYSTEM

AX	4C	61
BX	00	00
CX	01	10
DX	00	61
CS	F400	
IP	0204	
SS	0710	
SP	00FA	
BP	0000	
SI	0000	
DI	0000	
DS	0700	
ES	0700	

P420H: 00 0000 NULL ADD BX + \$11, AL  
F4210: 00 0000 NULL ADD BX + \$11, AL  
F4211: 00 0000 NULL ADD BX + \$11, AL  
F4212: 00 0000 NULL ADD BX + \$11, AL  
F4213: 00 0000 NULL ADD BX + \$11, AL  
F4214: 00 0000 NULL ADD BX + \$11, AL  
F4215: 00 0000 NULL ADD BX + \$11, AL

screen source reset aux vars debug stack flags

## Task-34:

edit: C:\My Files\Szabist\5th Semester\COAL-LAB\LAB-Manual\fwdcoallabtask\Task 34.asm

file edit bookmarks assembler emulator math ascii codes help

bB

```

01 ;Rohail Rathore
02 ;2012362
03
04 .model small
05 .stack 100h
06 .data
07 .code
08
09 main proc
10     mov ah,1
11     int 21h
12     and al,11011111b
13
14     mov dl,al
15     mov ah,2
16     int 21h
17
18     mov ah,4ch
19     int 21h
20
21     main endp
22 end main
23 
```

original source c... emulator screen (80x25 chars)

emulator: Task 34.exe\_

Load reload step back single step run step delay ms: 0

registers F400:0204 F400:0204

AX	0C	42
BX	00	00
CX	01	10
DX	00	42
CS	F400	
IP	0204	
SS	0710	
SP	00FA	
BP	0000	
SI	0000	
DI	0000	
DS	0700	
ES	0700	

message  
PROGRAM HAS RETURNED CONTROL  
TO THE OPERATING SYSTEM

OK

screen source reset aux vars debug stack flags

## Task-35:

edit: C:\My Files\Szabist\5th Semester\COAL-LAB\LAB-Manual\fwdcoallabtask\Task 35.asm

file edit bookmarks assembler emulator math ascii codes help

bB

```

01 ;Rohail Rathore
02 ;2012362
03
04 .model small
05 .stack 100h
06 .data
07     count db ?
08     input db 20 dup (?)
09     upper db 10,13, ?'Upper Case: ',?
10     lower db 10,13, ?'Lower Case: ',?
11     toggle db 10,13, ?'Toggle Case: ',?
12
13 .code
14 main proc
15     mov ax,0
16     ds,ax
17     lea si,stat
18     mov ah,1
19     int 21h
20     mov al,ds:[si]
21     mov count,0
22
23 start:
24     mov ah,1
25     int 21h
26     cmp al,'Z'
27     je end1
28     mov ah,2
29     int 21h
30     inc count
31     inc si
32     jmp start
33
34 end1:
35
36     lea dx,upper
37     mov ds,dx
38     lea si,stat
39     mov ah,1
40     int 21h
41     mov al,ds:[si]
42     mov count,0
43
44 L1:
45     mov dl,[si]
46     and dl,11011111b
47     mov ah,2
48     int 21h
49     inc si
50     loop L1
51
52     lea dx,lower
53     mov ds,dx
54     lea si,stat
55     mov ah,1
56     int 21h
57     mov al,ds:[si]
58     mov count,0
59
60 L2:
61     mov dl,[si]
62     or dl,00100000b
63     mov ah,2
64     int 21h
65     inc si
66     loop L2
67
68     lea dx,toggle
69     mov ds,dx
70     lea si,stat
71     mov ah,1
72     int 21h
73     mov al,ds:[si]
74     mov count,0
75
76 L3:
77     mov dl,[si]
78     xor dl,0100000b
79     mov ah,2
80     int 21h
81     inc si
82     loop L3
83
84     mov ah,4ch
85     int 21h
86
87 main endp
88 end main
89 
```

original source c... emulator screen (80x25 chars)

emulator: Task 35.exe\_

Load reload step back single step run step delay ms: 0

registers F400:0204 F400:0204

AX	4C	44
BX	00	00
CX	00	00
DX	00	44
CS	F400	
IP	0204	
SS	0710	
SP	00FA	
BP	0000	
SI	0025	
DI	0000	
DS	0720	
ES	0700	

message  
PROGRAM HAS RETURNED CONTROL  
TO THE OPERATING SYSTEM

OK

screen source reset aux vars debug stack flags

## Task-36:

edit: C:\My Files\Szabist\5th Semester\COAL-LAB\LAB-Manual\wdcoallabtask\Task 36.asm

file edit bookmarks assembler emulator math ascii codes help

new open examples save compile emulate calculator converter options help about

```

;Initial Rethore
.model I386
.data
msg1 db "Enter Password:",0
msg2 db "Correct Password!",0
Inputpassword db 20 dup("$")
count db 0dh,0ah,""
count dd 0
count dw 0
count db 0
count duw 0

code
main proc
    mov ax,@data
    mov ds,ax
    lea dx,msg1
    call print
    lea si,password
lengthCounter:
    cmp al,'1'
    jle L1
    inc count
    JMP lengthCounter
L1:
    mov dl,offset Inputpassword
    input:
        int 21h
        cmp al,13
        je check
        mov [dl],al
        inc dl
        inc count
        call star
        int 21h
        call newline
check:
    mov ah,count
    mov bh,count
    cmp ah,bh
    jne incorrect
    mov ah,count
    mov dl,offset password
    mov di,offset Inputpassword
compare:
    mov al,[di]
    inc di
    cmp al,[dl]
    inc dl
    loop compare
    jne incorrect
    call newline
    lea dx,msg2
    call print
    int 21h
incorrect:
    call newline
    lea dx,msg1
    call print
    int 21h
correct:
    call newline
    lea dx,msg2
    call print
    int 21h
exit:
    mov ah,4ch
    int 21h
main endp
    
```

Activate Windows  
Go to Settings to activate Windows.

edit: C:\My Files\Szabist\5th Semester\COAL-LAB\LAB-Manual\wdcoallabtask\Task 36.asm

file edit bookmarks assembler emulator math ascii codes help

new open examples save compile emulate calculate

emulator screen (80x25 chars)

Enter password: \*\*\*\*\*  
Correct Password

original source

```

    IRO SI
    INC COUNT
    JMP LENGTHCOUNTER
L1:
    MOV DL, OFFSET INPUTPASSWORD
INPUT:
    INT 21H
    CMP AL,13
    JE CHECK
    MOV [DL], AL
    INC DL
    INC COUNT
    JMP INPUT
    CALL NEWLINE
CHECK:
    MOV AH,COUNT
    MOV BH,COUNT
    CMP AH,BH
    JNE INCORRECT
    MOV AH,COUNT
    MOV DL,OFFSET PASSWORD
    MOV DI,OFFSET INPUTPASSWORD
COMPARE:
    MOV AL,[DI]
    INC DI
    CMP AL,[DL]
    INC DL
    LOOP COMPARE
    JNE INCORRECT
    CALL NEWLINE
    CALL PRINT
    INT 21H
INCORRECT:
    CALL NEWLINE
    CALL PRINT
    INT 21H
    CALL EXIT
    MOV AH,4CH
    INT 21H
MAIN ENDP
STAR PROC
    STOSB
    INT 21H
    RET
STAR ENDP
NEWLINE PROC
    MOV AH,10H
    INT 21H
    RET
NEWLINE ENDP
PRINT PROC
    MOV DL,9
    INT 21H
    RET
PRINT ENDP
END MAIN
    
```

emulator: Task 36.exe

File math debug view external virtual devices virtual drive help

Load reload step back single step run step delay ms: 0

Registers F400:0204

AX	0C 24
BX	00 6E
CX	00 00
DX	00 11
CS	F400
IP	0204
SS	0710
SP	00FA
BP	0000
SI	004E
DI	003A
DS	0728
ES	0700

message

PROGRAM HAS RETURNED CONTROL TO THE OPERATING SYSTEM

OK

screen source reset aux vars debug stack flags

Activate Windows  
Go to Settings

## Task-37:

edit: C:\My Files\Szabist\5th Semester\COAL-LAB\LAB-Manual\fwdcoallabtask\Task-37.asm

file edit bookmarks assembler emulator math ascii codes help

new open examples save compile emulate calculate

```

01 .model small
02 .stack 100h
03 .data
04
05 .code
06
07 main proc
08
09     mov al,2
10    mov bl,5
11
12    mul bl
13
14    aam
15    mov ch,ah
16    mov cl,al
17
18    mov dl,ch
19    add dl,48
20    mov ah,2
21    int 21h
22
23    mov dl,cl
24    add dl,48
25    mov ah,2
26    int 21h
27
28    mov ah,4ch
29    int 21h
30
31 main endp
32 end main

```

original source c... F400:0204 F400:0204

registers H L message

AX	AC	30
BX	00	05
CX	01	00
DX	00	30
CS	F400	
IP	0204	
SS	0710	
SP	00FA	
BP	0000	
SI	0000	
DI	0000	
DS	0700	
ES	0700	

F420F: 00 000 NULL ADD [BX + SI], AL  
F4210: 00 000 NULL ADD [BX + SI], AL  
F4211: 00 000 NULL ADD [BX + SI], AL  
F4212: 00 000 NULL ADD [BX + SI], AL  
F4213: 00 000 NULL ADD [BX + SI], AL  
F4214: 00 000 NULL ADD [BX + SI], AL  
F4215: 00 000 NULL ADD [BX + SI], AL

screen source reset aux vars debug stack flags

Activate V

## Task-38:

edit: C:\My Files\Szabist\5th Semester\COAL-LAB\LAB-Manual\fwdcoallabtask\Task-38.asm

file edit bookmarks assembler emulator math ascii codes help

new open examples save compile emulate calculate

2\*5=10

```

01 ;Rohail Rathore
02 ;20B2362
03 .model small
04 .stack 100h
05 .data
06
07 .code
08
09 main proc
10
11    mov ah,1
12    int 21h
13
14    sub al,48
15
16    mov bl,al
17
18    mov dl,'*'
19    int 21h
20
21    mov ah,1
22    int 21h
23
24    sub al,48
25
26    mul bl
27
28    aam
29
30    mov bl,al
31    mov bh,ah
32
33    mov dl,bh
34
35    mov d1,*
36    add d1,dl
37    mov ah,2
38    int 21h
39
40    mov dl,bh
41    add dl,48
42    mov ah,2
43    int 21h
44
45    mov dl,bl
46    add dl,48
47    mov ah,2
48    int 21h
49
50    mov ah,4ch
51    int 21h
52
53 main endp
54 end main

```

original source F400:0204 F400:0204

registers H L message

AX	4C	30
BX	01	00
CX	01	38
DX	00	30
CS	F400	
IP	0204	
SS	0710	
SP	00FA	
BP	0000	
SI	0000	
DI	0000	
DS	0700	
ES	0700	

F420F: 00 000 NULL ADD [BX + SI], AL  
F4210: 00 000 NULL ADD [BX + SI], AL  
F4211: 00 000 NULL ADD [BX + SI], AL  
F4212: 00 000 NULL ADD [BX + SI], AL  
F4213: 00 000 NULL ADD [BX + SI], AL  
F4214: 00 000 NULL ADD [BX + SI], AL  
F4215: 00 000 NULL ADD [BX + SI], AL

screen source reset aux vars debug stack flags

Activate V

## Task-39:

edit: C:\My Files\Szabist\5th Semester\COAL-LAB\LAB-Manual\fwdcoallabtask\Task-20.asm

file edit bookmarks assembler emulator math ascii codes help

new open examples save compile emulate calculate 5.0

```

01;Rohail Rathore
02;2012362
03.model small
04.stack 100h
05.data
06
07 quotient db ?
08 remainder db ?
09
10.code
11.main proc
12
13 mov ax,25
14 mov bl,5
15
16 div bl
17
18 mov quotient,al
19 mov remainder,ah
20
21 mov dl,quotient
22 add dl,48
23 mov ah,2
24 int 21h
25
26 mov dl,'
27 mov ah,2
28 int 21h
29
30 mov dl,remainder
31 add dl,48
32 mov ah,2
33 int 21h
34
35 mov ah,4ch
36 int 21h
37
38 main endp
39

```

original source c... file math debug view external virtual devices virtual drive help

Load reload step back single step run step delay ms: 0

registers H L F400:0204 F400:0204

AX	4C	30
BX	00	05
CX	01	3E
DX	00	30
CS	F400	
IP	0204	
SS	0710	
SP	00FA	
BP	0000	
SI	0000	
DI	0000	
DS	0700	
ES	0700	

message PROGRAM HAS RETURNED CONTROL TO THE OPERATING SYSTEM OK

F42010: 00 0000 NULL ADD EBX + \$11, AL  
F4210: 00 0000 NULL ADD EBX + \$11, AL  
F4211: 00 0000 NULL ADD EBX + \$11, AL  
F4212: 00 0000 NULL ADD EBX + \$11, AL  
F4213: 00 0000 NULL ADD EBX + \$11, AL  
F4214: 00 0000 NULL ADD EBX + \$11, AL  
F4215: 00 0000 NULL ADD EBX + \$11, AL

screen source reset aux vars debug stack flags

## Task-40:

edit: C:\My Files\Szabist\5th Semester\COAL-LAB\LAB-Manual\fwdcoallabtask\Task-40.asm

file edit bookmarks assembler emulator math ascii codes help

new open examples save compile emulate calculate 8240

```

01;Rohail Rathore
02;2012362
03.model small
04.stack 100h
05.data
06
07
08.main proc
09
10 mov ah,1
11 int 21h
12
13 mov cl,al
14 sub cl,48
15
16 mov ah,1
17 int 21h
18
19 mov bl,al
20 sub bl,48
21
22 mov al,cl
23
24 mov ah,00h
25 div bl
26
27 mov cx,00h
28 mov cx,ax
29
30 mov dl,c1 ;Quotient
31 add dl,48
32 mov ah,2
33 int 21h
34
35 mov d1,ch ;Remainder
36 add d1,48
37 mov ah,2
38 int 21h
39
40 mov ah,4ch
41 int 21h
42
43 main endp
44
45

```

original source c... file math debug view external virtual devices virtual drive help

Load reload step back single step run step delay ms: 0

registers H L F400:0204 F400:0204

AX	4C	30
BX	00	02
CX	00	04
DX	00	30
CS	F400	
IP	0204	
SS	0710	
SP	00FA	
BP	0000	
SI	0000	
DI	0000	
DS	0700	
ES	0700	

message PROGRAM HAS RETURNED CONTROL TO THE OPERATING SYSTEM OK

F42010: 00 0000 NULL ADD EBX + \$11, AL  
F4210: 00 0000 NULL ADD EBX + \$11, AL  
F4211: 00 0000 NULL ADD EBX + \$11, AL  
F4212: 00 0000 NULL ADD EBX + \$11, AL  
F4213: 00 0000 NULL ADD EBX + \$11, AL  
F4214: 00 0000 NULL ADD EBX + \$11, AL  
F4215: 00 0000 NULL ADD EBX + \$11, AL

screen source reset aux vars debug stack flags

## Task-41:

edit: C:\My Files\Szabist\5th Semester\COAL-LAB\LAB-Manual\fwdcoallabtask\Task 41.asm

file edit bookmarks assembler emulator math ascii codes help

```

001;Rohail Rathore
002:2812362
003.nodel small
004.stack 100h
005.data
006
007 msg db "Enter a digit: $"
008 count db 01
009 num db ?
010
011 .code
012 main proc
013     mov ax,0data
014     mov ds,ax
015
016     mov cx,10
017
018     lea si,msg
019     mov ah,9
020     int 21h
021
022     mov ah,1
023     int 21h
024     sub al,48
025
026     mov num,al
027     call newline
028     call newline
029
030     loop1:
031
032     mov al,num
033
034     mov dl,al
035     add dl,48
036     mov ah,2
037     int 21h
038
039     mov dl,' '
040     mov ah,2
041     int 21h
042
043     mov al,count
044
045     aaa
046     mov bl,al
047     mov bh,ah
048
049     mov dl,bh
050     add dl,48
051     mov ah,2
052     int 21h
053
054     mov dl,al
055     add dl,48
056     mov ah,2
057     int 21h
058

```

Activate Windows  
Go to Settings to activate Wind

edit: C:\My Files\Szabist\5th Semester\COAL-LAB\LAB-Manual\fwdcoallabtask\Task 41.asm

file edit bookmarks assembler emulator math ascii codes help

emulator screen (80x25 chars)

```

049 mov dl,0bh
050 add dl,48
051 mov ah,2
052 int 21h
053
054 mov dl,bl
055 add dl,48
056 mov ah,2
057 int 21h
058
059 mov dl,' '
060 mov ah,2
061 int 21h
062
063 mov al,num
064
065 nul count
066
067 aaa
068
069 mov bl,al
070 mov ah,ah
071
072 mov dl,bh
073 add dl,48
074 mov ah,2
075 int 21h
076
077 mov dl,bl
078 add dl,48
079 mov ah,2
080 int 21h
081
082 call newline
083 inc count
084
085 loop loop1
086
087 mov ah,4ch
088 int 21h
089
090 main endp
091
092 newline proc
093     inc count
094     mov dl,10
095     mov ah,2
096     int 21h
097
098     mov dl,13
099     mov ah,2
100    int 21h
101
102     ret
103     newline endp
104
105 end main

```

original source c... file math debug view external virtual devices virtual drive help

Load step back single step run step delay ms: 0

registers H L F400:0204 F400:0204

AX	0C 0D
BX	08 00
CX	00 00
DX	00 00
CS	F400
IP	0204
SS	0710
SP	00FA
BP	0000
SI	0000
DI	0000
DS	0720
ES	0700

PROGRAM HAS RETURNED CONTROL TO THE OPERATING SYSTEM

OK

screen source reset aux vars debug stack flags

line: 2 col: 9 drag a file here to open

Activate Windows  
Go to Settings to activate Wind

## Task-42:

edit: C:\My Files\Szabist\5th Semester\COAL-LAB\LAB-Manual\fwdcoallabtask\Task 42.asm

File edit bookmarks assembler emulator math ascii codes help

```

01 ;Babail Rathore
02 ;2812362
03 print macro string
04 lea dx,[string]
05 mov ah,9
06 int 21h
07 endm
08
09 newline macro
10 mov ah,2
11 mov dl,13
12 int 21h
13
14 mov dl,10
15 mov ah,2
16 int 21h
17 endm
18
19 display macro num1
20 mov dl,num1
21 mov ah,2
22 int 21h
23 endm
24
25 numberDisplay macro var
26    mov ah,1
27    int 21h
28    add dl,48
29    mov ah,2
30    int 21h
31 endm
32
33 .model small
34 .stack 100h
35 .data
36 msg db 'Enter a Number: $'
37 count db 1
38 ten db ?
39 unit db ?
40 num db ?
41
42 .code
43
44 main proc
45    mov ax, @data
46    mov ds, ax
47
48    print msg
49
50    mov ah, 1
51    int 21h
52    sub al, 48
53    mov num, al
54
55    mov cx, 10
56    loop:
57    newline

```

Activate Windows  
Go to Settings to activate Windows

edit: C:\My Files\Szabist\5th Semester\COAL-LAB\LAB-Manual\fwdcoallabtask\Task 42.asm

File edit bookmarks assembler emulator math ascii codes help

emulator screen (80x25 chars)

```

31 Enter a Number: 4
32 4*01=04
33
34 .model small
35 .stack 100h
36 .data
37 msg db 'Enter a Number: $'
38 count db 1
39 ten db ?
40 unit db ?
41
42 .code
43
44 main proc
45    mov ax, @data
46    mov ds, ax
47
48    print msg
49
50    mov ah, 1
51    int 21h
52    sub al, 48
53    mov num, al
54
55    mov cx, 10
56    loop:
57
58    newline
59    numberDisplay num
60    display '%'
61
62    mov al, count
63    aaa
64    mov ten, ah
65    mov unit, al
66
67    numberDisplay ten
68    numberDisplay unit
69    display '%'
70
71    mov al, num
72    mov cl, count
73    aaa
74    mov ten, ah
75    mov unit, al
76
77    numberDisplay ten
78    numberDisplay unit
79    display '%'
80    inc count
81    loop loop1
82
83    mov ah, 4ch
84    int 21h
85
86    main endp
87 end main

```

original source c... -

emulator: Task 42.exe...

file math debug view external virtual devices virtual drive help

Load Reload step back single step run step delay ms: 0

registers H L

AX	4C	30
BX	00	00
CX	00	00
DX	00	30
CS	F400	
IP	0204	
SS	0710	
SP	00FA	
BP	0000	
SI	0000	
DI	0000	
DS	0720	
ES	0700	

F400:0204 F400:0204

message

PROGRAM HAS RETURNED CONTROL TO THE OPERATING SYSTEM

OK

F420F: 00 0000 NULL ADD BX + SI, AL
F4210: 00 0000 NULL ADD BX + SI, AL
F4211: 00 0000 NULL ADD BX + SI, AL
F4212: 00 0000 NULL ADD BX + SI, AL
F4213: 00 0000 NULL ADD BX + SI, AL
F4214: 00 0000 NULL ADD BX + SI, AL
F4215: 00 0000 NULL ...

screen source reset aux vars debug stack flags

## Task-43:

edit: C:\My Files\Szabist\5th Semester\COAL-LAB\LAB-Manual\fwdcoallabtask\Task 43.asm

emulator screen (80x25 chars)

```

new open examples save compile emulate calculate even
01 .model small
02 .stack 100h
03 .data
04 even db 10,13,'even$'
05 odd db 10,13,'odd$'
06 .code
07 main proc
08
09 mov ax,0data
10 mov ds,ax
11
12 mov ah,1
int 21h
13
14 mov dl,al
15 mov bl,2
16
17 cmp ah,0
18 jne oddPrint
19
20 lea dx,even
21 mov ah,9
22 int 21h
23
24 jmp exit
25
26 oddPrint:
27 lea dx,odd
28 mov ah,9
29 int 21h
30
31 exit:
32 mov ah,4ch
33 int 21h
34
35 main endp
36 end main
37
38

```

original source c... emulator: Task 43.exe

file math debug view external virtual devices virtual drive help

Load step back single step run step delay ms: 0

registers F400:0204 F400:0204

AX	4C	24
BX	00	02
CX	01	38
DX	00	00
CS	F400	
IP	0204	
SS	0710	
SP	00FA	
BP	0000	
SI	0000	
DI	0000	
DS	0720	
ES	0700	

message PROGRAM HAS RETURNED CONTROL TO THE OPERATING SYSTEM

OK

screen source reset aux vars debug stack flags

## Task-44:

edit: C:\My Files\Szabist\5th Semester\COAL-LAB\LAB-Manual\fwdcoallabtask\Task 44.asm

emulator screen (80x25 chars)

```

new open examples save compile emulate calculate Enter a Statement: Hello Worlds Total Characters: 11
01 Rohail Rathore
02 ;2812362
03 .model small
04 .stack 100h
05 .data
06
07 msg1 db 'Enter a Statement: $'
08 msg2 db 10,13,'Total Characters: $'
09 total db 0
10
11 .code
12 main proc
13 mov ax,0data
14 mov ds,ax
15
16 lea dx,msg1
17 mov ah,9
18 int 21h
19
20 counter:
21 mov ah,1
int 21h
22
23 cmp al,13
je exit
24
25 cmp al,' '
je counter
26 incCount:
27 inc total
28 jmp counter
29
30 incCount:
31 inc total
32 jmp counter
33
34 exit:
35 lea dx,msg2
36 mov ah,9
37 int 21h
38
39 mov al,total
40 aaa
41
42 mov dh,al
43 add ah,48
44 mov dl,ah
45 mov ah,2
46
47 int 21h
48
49 mov al,dh
50 add al,48
51 mov dl,al
52 mov ah,2
53 int 21h
54
55 mov ah,4ch
56 int 21h
57
58 main endp
59 end main
60
61

```

original source emulator: Task 44.exe

file math debug view external virtual devices virtual drive help

Load step back single step run step delay ms: 0

registers F400:0204 F400:0204

AX	4C	31
BX	00	00
CX	01	75
DX	01	31
CS	F400	
IP	0204	
SS	0710	
SP	00FA	
BP	0000	
SI	0000	
DI	0000	
DS	0720	
ES	0700	

message PROGRAM HAS RETURNED CONTROL TO THE OPERATING SYSTEM

OK

screen source reset aux vars debug stack flags

## Task-45:

edit: C:\My Files\Szabist\5th Semester\COAL-LAB\LAB-Manual\fwdcoallabtask\Task 45.asm

```

001 ;Rohail Rathore
002 :2012362
003 newline macro
004    db ' '
005    mov ah,2
006    int 21h
007
008    mov ah,10
009    mov ah,2
010    int 21h
011    newline endm
012
013 print macro str
014    lea dx,str
015    mov ah,9
016    int 21h
017    print endm
018
019 .model small
020 .stack 100h
021 .data
022    msg db 'Press any Key: $'
023    upp db 'user pressed Upper Case letter$'
024    low db 'user pressed Lower Case letter$'
025    num db 'user pressed Number$'
026    sym db 'user pressed Symbol$'
027    again db 'Press Y to run again or ESC to terminate: $'
028
029 .code
030
031 main proc
032     mov ax,0data
033     mov ds,ax
034
035 start:
036     newline
037     print msg
038
039     mov ah,1
040     int 21h
041
042     cmp al,'A'
043     jb symbol ;j1=jb-->Jump if less than
044     cmp al,57
045     jle number ;jle-->Jump Less than equal to
046     cmp al,65
047     jg symbol
048     cmp al,91
049     jl upper
050     cmp al,37
051     jl symbol
052     cmp al,122
053     jl lower
054     cmp al,126
055     jle symbol
056
057 lower:
058     newline
059     print low
060     jmp endi
061
062 upper:
063     newline
064     print upp
065     jmp endi
066
067 number:
068     newline
069     print num
070     jmp endi
071
072 symbol:
073     newline
074     print sym
075     jmp endi
076
077 end1:
078     newline
079     print again
080     mov ah,1
081     int 21h
082
083     cnp al,'y'
084     exit:
085     mov ah,4ch
086     int 21h
087
088     cnp al,'Y'
089     exit:
090     cmp al,27
091     je exit
092     jmp end1
093
094 exit:
095     mov ah,4ch
096     int 21h
097
098 main endp
099 end main
100

```

Activate Windows  
Go to Settings to activate Windows.

edit: C:\My Files\Szabist\5th Semester\COAL-LAB\LAB-Manual\fwdcoallabtask\Task 45.asm

emulator screen (80x25 chars)

```

043 jb symbol ;j1=jb-->Jump if
044 cmp al,57
045 jle number ;jle-->Jump Less
046 cmp al,65
047 jl symbol
048 cmp al,91
049 jl upper
050 cmp al,37
051 jl symbol
052 cmp al,122
053 jl lower
054 cmp al,126
055 jle symbol
056
057 lower:
058 newline
059 print low
060 jmp endi
061
062 upper:
063 newline
064 print upp
065 jmp endi
066
067 number:
068 newline
069 print num
070 jmp endi
071
072 symbol:
073 newline
074 print sym
075 jmp endi
076
077 end1:
078 newline
079 print again
080 mov ah,1
081 int 21h
082
083 cnp al,'y'
084 exit:
085 mov ah,4ch
086 int 21h
087
088 cnp al,'Y'
089 exit:
090 cmp al,27
091 je exit
092 jmp end1
093
094 exit:
095 mov ah,4ch
096 int 21h
097
098 main endp
099 end main
100

```

emulator: Task 45.exe...

original source c...

F400:0204 F400:0204

message

PROGRAM HAS RETURNED CONTROL TO THE OPERATING SYSTEM

OK

	H	L
AX	4C	1B
BX	00	00
CX	02	6B
DX	00	78
CS	F400	
IP	0204	
SS	0710	
SP	00FA	
BP	0000	
SI	0000	
DI	0000	
DS	0720	
ES	0700	

F420F: 00 00 NULL ADD BX + \$1, AL  
F4210: 00 00 NULL ADD BX + \$1, AL  
F4211: 00 00 NULL ADD BX + \$1, AL  
F4212: 00 00 NULL ADD BX + \$1, AL  
F4213: 00 00 NULL ADD BX + \$1, AL  
F4214: 00 00 NULL ADD BX + \$1, AL  
F4215: 00 00 NULL ...

screen source reset aux vars debug stack flags

Activate Go to