

NAME: Babar Ali

SAP ID: 65731

SUBJECT: COAL

LAB10-TASK1:

The screenshot shows a debugger interface with the following components:

- Assembly View:** Displays the assembly code with some instructions highlighted in yellow.
- Registers View:** Shows the CPU registers (AX, BX, CX, DX, CS, IP, SS, SP, BP, SI, DI, DS, ES) and their current values.
- Memory Dump:** Shows the memory dump starting at address 0700:013B, where the instruction at 0713B is highlighted in blue.
- Control Buttons:** Includes buttons for Load, Reload, Step Back, Single Step, Run, and Step Delay ms: 0.

```
01 org 100h
02 .data
03 a db 11001010b
04 .code
05 mov ax,@data
06 mov ds,ax
07 mov bl,a
08 shl bl,1
09 shl bl,2
10 shl bl,3
11 shl bl,2
12 shl bl,3
13
14 ror bl,2
15 ror bl,1
```

Registers View:

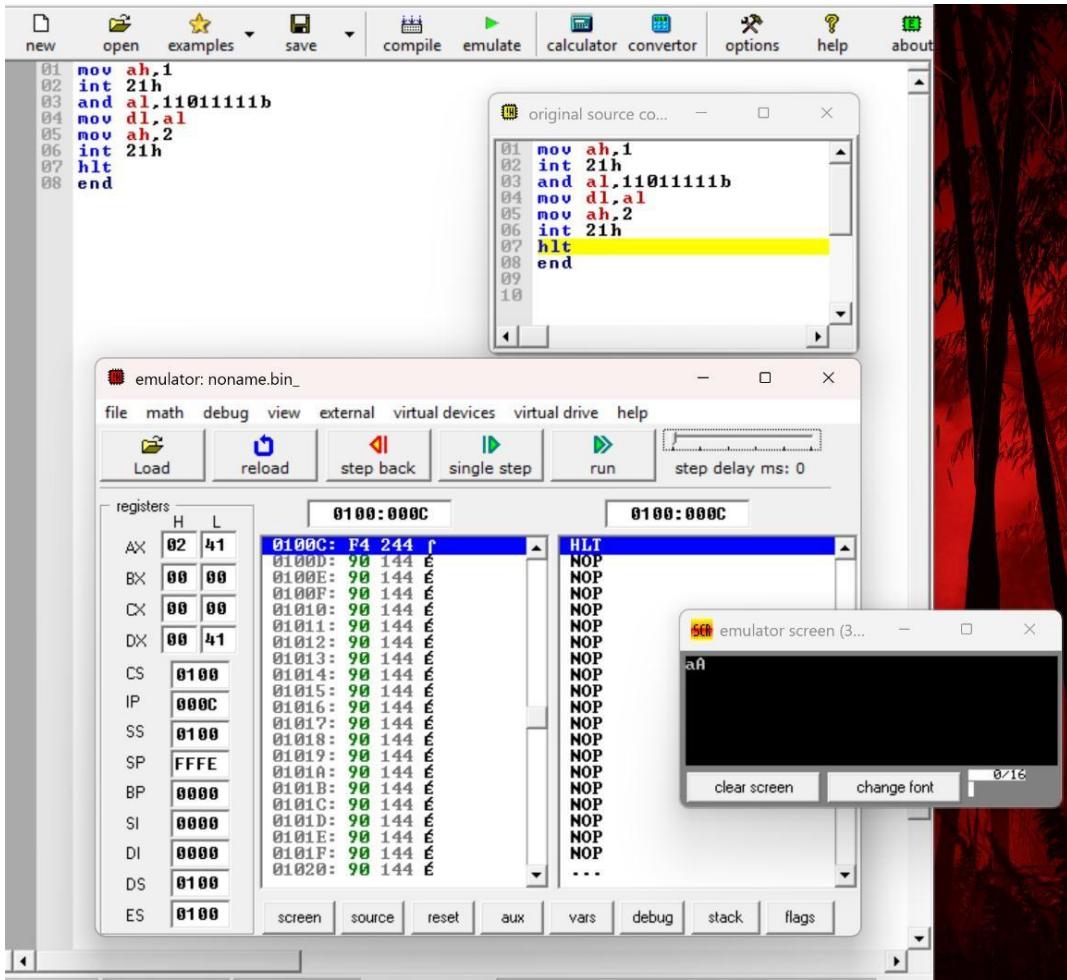
	H	L
AX	07	00
BX	00	00
CX	00	27
DX	00	00
CS	0700	
IP	013B	
SS	0700	
SP	FFFE	
BP	0000	
SI	0000	
DI	0000	
DS	0700	
ES	0700	

Memory Dump View:

	0700:013B	0700:013B
0712A:	90 144 E	NOP
0712B:	90 144 E	NOP
0712C:	90 144 E	NOP
0712D:	90 144 E	NOP
0712E:	90 144 E	NOP
0712F:	90 144 E	NOP
07130:	90 144 E	NOP
07131:	90 144 E	NOP
07132:	90 144 E	NOP
07133:	90 144 E	NOP
07134:	90 144 E	NOP
07135:	90 144 E	NOP
07136:	90 144 E	NOP
07137:	90 144 E	NOP
07138:	90 144 E	NOP
07139:	90 144 E	NOP
0713A:	90 144 E	NOP
0713B:	F4 244 F	HLT
0713C:	00 000 NULL	ADD [BX + SII], AL
0713D:	00 000 NULL	ADD [BX + SII], AL
0713E:	00 000 NULL	...

Explanation: The code moves data to registers, then performs a series of left shifts on bl by 1, 2, and 3 bits. Finally, it uses XOR to combine bl shifted by 2 with bl shifted by 1.

LAB10-TASK2:



Explanation: It converts the character to uppercase by AND AL with 11011111b to clear the lowercase bit, moves it to DL, and outputs it using INT 21h with AH=2, before halting.

LAB10-TASK3:

The screenshot shows a Windows desktop environment with several open windows. At the top is a menu bar with options like file, edit, bookmarks, assembler, emulator, math, ascii codes, help, new, open, examples, save, compile, emulate, calculator, convertor, options, help, and about. Below the menu is a toolbar with icons for new, open, examples, save, compile, emulate, calculator, convertor, options, help, and about.

The main window is titled "emulator: Babar_65731_L10_T2.bin". It has tabs for file, math, debug, view, external, virtual devices, virtual drive, help, Load, reload, step back, single step, run, and step delay ms: 0. The assembly code window shows the following code:

```
01 mov ah,1  
02 int 21h  
03 mov bl,00100000b  
04 or al,bl  
05 mov dl,al  
06 mov ah,2  
07 int 21h  
08 hlt  
09 end
```

The instruction at address 0100h (mov ah,1) is highlighted in yellow. The registers window shows the following values:

	H	L
AX	02	61
BX	00	20
CX	00	00
DX	00	61
CS	0100	
IP	000E	
SS	0100	
SP	FFFE	
BP	0000	
SI	0000	
DI	0000	
DS	0100	
ES	0100	

The memory dump window shows the memory starting at address 0100h:

	0100:000E	0100:000F
0100E: F4 244 E	0100F: 90 144 E	
01010: 90 144 E	01011: 90 144 E	
01012: 90 144 E	01013: 90 144 E	
01014: 90 144 E	01015: 90 144 E	
01016: 90 144 E	01017: 90 144 E	
01018: 90 144 E	01019: 90 144 E	
0101A: 90 144 E	0101B: 90 144 E	
0101C: 90 144 E	0101D: 90 144 E	
0101E: 90 144 E	0101F: 90 144 E	
01020: 90 144 E	01021: 90 144 E	
01022: 90 144 E	...	

The output window shows the character 'a'.

Explanation: Reads a key press, converts uppercase letters to lowercase, prints the result, then stops.

LAB10-TASK4:

The screenshot shows a debugger interface with the following windows:

- Original source code:** Contains the assembly code for the program.
- Emulator screen:** Displays the output "Babar" followed by "rabaB".
- Registers pane:** Shows CPU register values. AX: 4C 42, BX: 00 00, CX: 00 00, DX: 01 42, CS: F400, IP: 02 04, SS: 0710, SP: 00FA, BP: 0000, SI: 0000, DI: 0000.
- Stack pane:** Shows memory starting at address F400:0204. The stack contains several ADD instructions and an INT 021h instruction.
- Memory dump pane:** Shows memory starting at address F4200. The dump shows the string "Babar" followed by "rabaB".

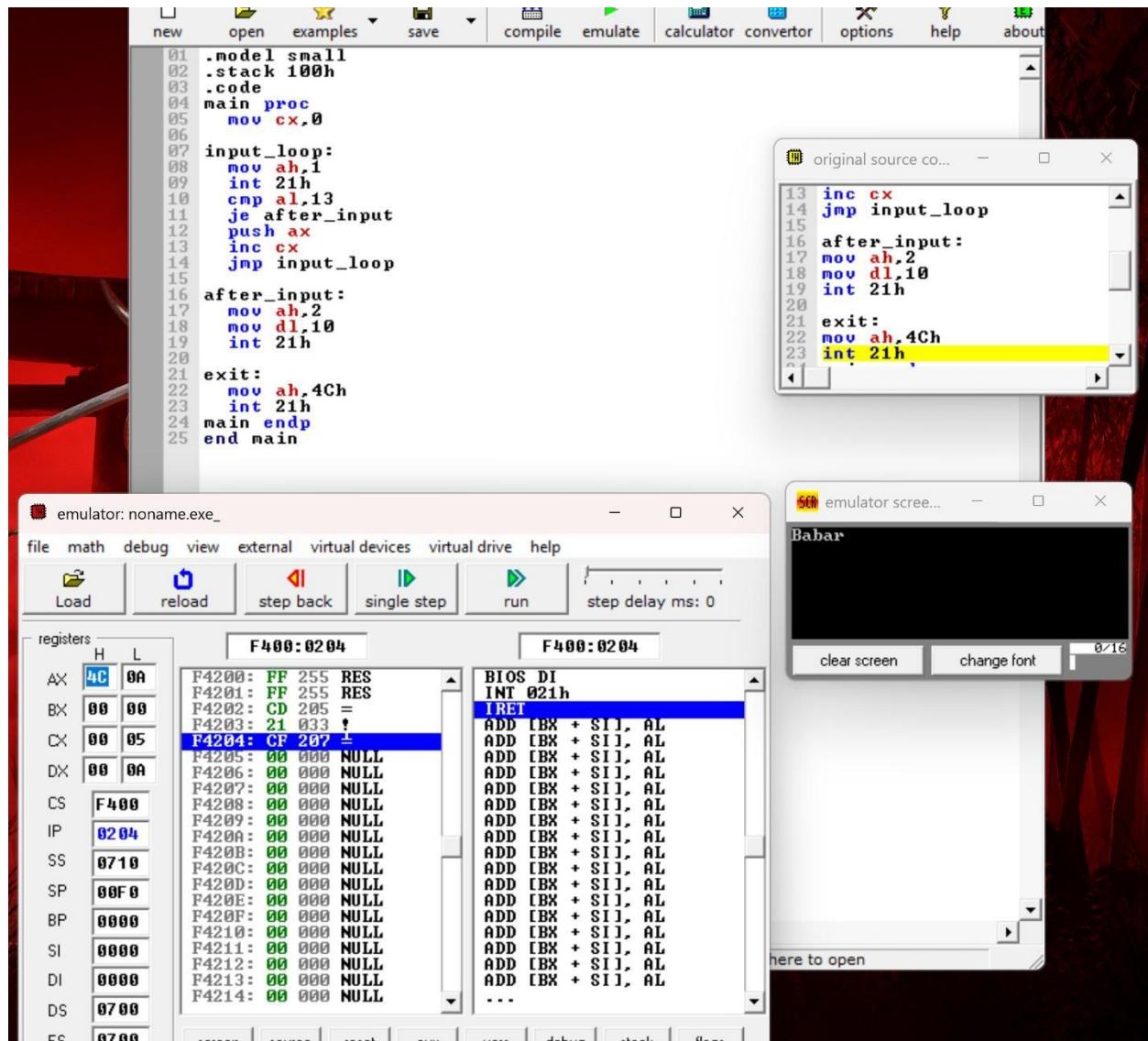
```

01 .model small
02 .stack 100h
03 .code
04 main proc
05 mov cx,0
06 input_loop:
07 mov ah,1
08 int 21h
09 cmp al,13
10 je reverse_start
11 push ax
12 inc cx
13 jmp input_loop
14
15 reverse_start:
16 mov ah,2
17 mov dl,10
18 int 21h
19 cmp cx,0
20 je exit
21
22 reverse_loop:
23 pop dx
24 int 21h
25 loop reverse_loop
26
27 exit:
28 mov ah,4Ch
29 int 21h
30 main endp
31 end main

```

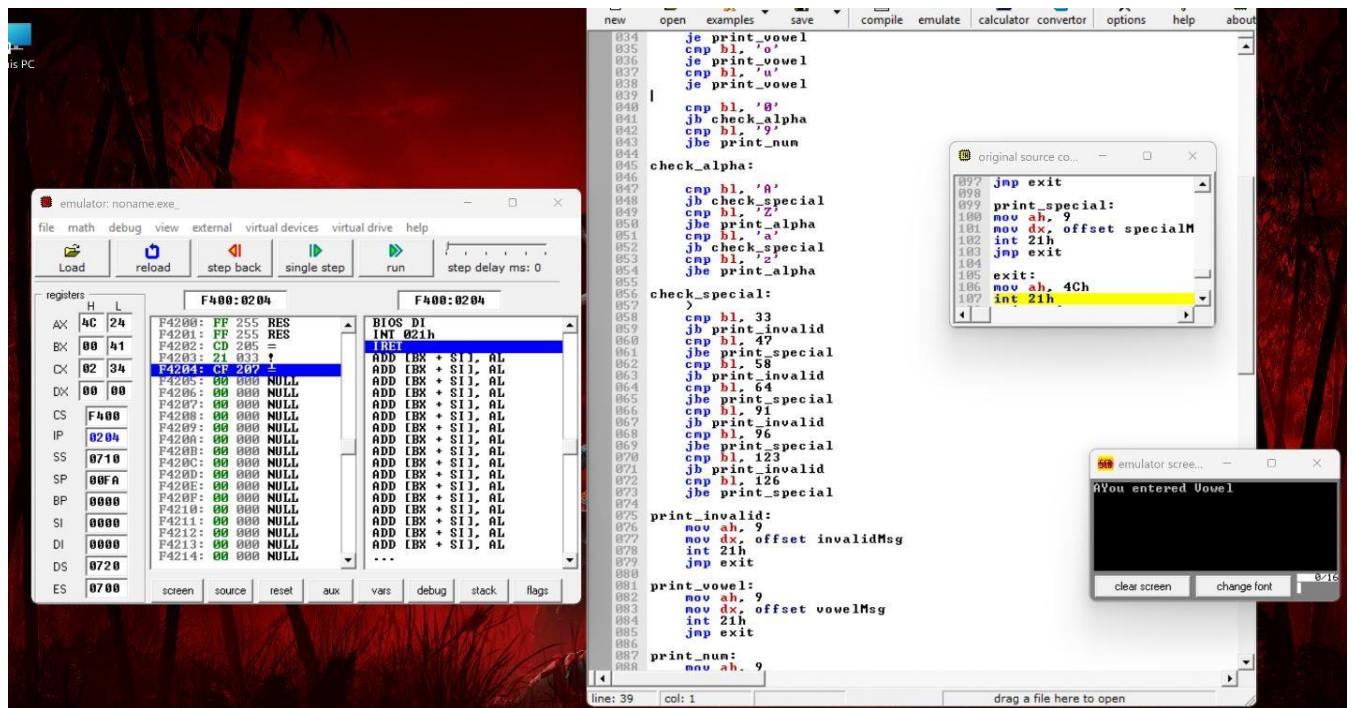
Explanation: This code reverse the name taken from User using Reverse loop.

LAB10-TASK5:



Explanation: This code Forward the characters taken from user using input loop and after input.

LAB10-TASK6:



Explanation:

The program reads a character and classifies it:

- Vowel → prints "vowel"
- Digit → prints "num"
- Letter → prints "alpha"
- Special char → prints "special"
- Else → prints "invalid" Then exits.