



COAL LABTASK 8

SAJID ISLAM
24P-0745

TASK 1:

Assembly Code:

```
ASM lb8q1.asm
1  [org 0x0100]
2  ✓ jmp start
3  |   num1: db 10111010b ; 8-bit number
4  ✓ start:
5  |   mov al, [num1]
6  |   rol al, 3
7  |   ror al, 3
8  |   rcl al, 3
9  |   rcr al, 3
10 mov ax, 0x4c00
11 int 0x21
```

- **num1** will store the **8-bit** value.
- **rol** function will rotate the number **3-bit** to the left.
- **ror** function will rotate the number **3-bit** to right changing to original number.
- Similarly, **rcl** will rotate the number to **3-bit** left and add carry every time at the **LSB**.
- Then the **rcr** will rotate the number to **3-bit** right and add carry every time at the **MSB**.



COAL LABTASK 8

SAJID ISLAM

24P-0745

Original:

```
DOSBox 0.74, Cpu speed: 3000 cycles, Frameskip 0, Program:...  
AX 00BA SI 0000 CS 19F5 IP 0107 Stack +0 0000 Flags 7200  
BX 0000 DI 0000 DS 19F5 +2 20CD  
CX 0018 BP 0000 ES 19F5 HS 19F5 +4 9FFF OF DF IF SF ZF AF PF CF  
DX 0000 SP FFFE SS 19F5 FS 19F5 +6 EA00 0 0 1 0 0 0 0 0  
  
CMD >  
0104 A00301 MOV AL,[0103]  
0107 C0C003 ROL AL,03  
010A C0C803 ROR AL,03  
010D C0D003 RCL AL,03  
0110 C0D803 RCR AL,03  
0113 B8004C MOV AX,4C00  
0116 CD21 INT 21  
0118 8956E4 MOV [BP-1C1],DX  
011B 8946E6 MOV [BP-1A1],AX  
  
DS:0000 CD 20 FF 9F 00 EA F0 FE  
DS:0008 AD DE 1B 05 C5 06 00 00  
DS:0010 18 01 10 01 18 01 92 01  
DS:0018 01 01 01 00 02 FF FF FF  
DS:0020 FF FF FF FF FF FF FF FF  
DS:0028 FF FF FF FF EB 19 C0 11  
DS:0030 A2 01 14 00 18 00 F5 19  
DS:0038 FF FF FF FF 00 00 00 00  
DS:0040 05 00 00 00 00 00 00 00  
DS:0048 00 00 00 00 00 00 00 00  
  
= f.Ω≡ i|..+...  
.....ff. ....  
δ.L.  
6.....J. ....  
.....
```

ROL:

```
DOSBox 0.74, Cpu speed: 3000 cycles, Frameskip 0, Program:...  
AX 00D5 SI 0000 CS 19F5 IP 010A Stack +0 0000 Flags 7201  
BX 0000 DI 0000 DS 19F5 +2 20CD  
CX 0018 BP 0000 ES 19F5 HS 19F5 +4 9FFF OF DF IF SF ZF AF PF CF  
DX 0000 SP FFFE SS 19F5 FS 19F5 +6 EA00 0 0 1 0 0 0 0 1  
  
CMD >  
0107 C0C003 ROL AL,03  
010A C0C803 ROR AL,03  
010D C0D003 RCL AL,03  
0110 C0D803 RCR AL,03  
0113 B8004C MOV AX,4C00  
0116 CD21 INT 21  
0118 8956E4 MOV [BP-1C1],DX  
011B 8946E6 MOV [BP-1A1],AX  
011E C746F60000 MOV [BP-0A1],0000  
  
DS:0000 CD 20 FF 9F 00 EA F0 FE  
DS:0008 AD DE 1B 05 C5 06 00 00  
DS:0010 18 01 10 01 18 01 92 01  
DS:0018 01 01 01 00 02 FF FF FF  
DS:0020 FF FF FF FF FF FF FF FF  
DS:0028 FF FF FF FF EB 19 C0 11  
DS:0030 A2 01 14 00 18 00 F5 19  
DS:0038 FF FF FF FF 00 00 00 00  
DS:0040 05 00 00 00 00 00 00 00  
DS:0048 00 00 00 00 00 00 00 00  
  
= f.Ω≡ i|..+...  
.....ff. ....  
δ.L.  
6.....J. ....  
.....
```



COAL LABTASK 8

SAJID ISLAM
24P-0745

ROR:

```
DOSBox 0.74, Cpu speed: 3000 cycles, Frameskip 0, Program:...  
AX 00BA SI 0000 CS 19F5 IP 0107 Stack +0 0000 Flags 7200  
BX 0000 DI 0000 DS 19F5 +2 20CD  
CX 0018 BP 0000 ES 19F5 HS 19F5 +4 9FFF OF DF IF SF ZF AF PF CF  
DX 0000 SP FFFE SS 19F5 FS 19F5 +6 EA00 0 0 1 0 0 0 0 0  
  
CMD >  
0104 A00301 MOV AL,[0103]  
0107 C0C003 ROL AL,03  
010A C0C803 ROR AL,03  
010D C0D003 RCL AL,03  
0110 C0D803 RCR AL,03  
0113 B8004C MOV AX,4C00  
0116 CD21 INT 21  
0118 8956E4 MOV [BP-1C],DX  
011B 8946E6 MOV [BP-1A],AX  
  
DS:0000 CD 20 FF 9F 00 EA F0 FE  
DS:0008 AD DE 1B 05 C5 06 00 00  
DS:0010 18 01 10 01 18 01 92 01  
DS:0018 01 01 01 00 02 FF FF FF  
DS:0020 FF FF FF FF FF FF FF FF  
DS:0028 FF FF FF FF EB 19 C0 11  
DS:0030 A2 01 14 00 18 00 F5 19  
DS:0038 FF FF FF FF 00 00 00 00  
DS:0040 05 00 00 00 00 00 00 00  
DS:0048 00 00 00 00 00 00 00 00  
  
= f.Ω= i|..+. ...  
.....ff. ....  
δ.L.  
ó.....J. ....  
.....
```

RCL:

```
DOSBox 0.74, Cpu speed: 3000 cycles, Frameskip 0, Program:...  
AX 00D6 SI 0000 CS 19F5 IP 0110 Stack +0 0000 Flags 7201  
BX 0000 DI 0000 DS 19F5 +2 20CD  
CX 0018 BP 0000 ES 19F5 HS 19F5 +4 9FFF OF DF IF SF ZF AF PF CF  
DX 0000 SP FFFE SS 19F5 FS 19F5 +6 EA00 0 0 1 0 0 0 0 1  
  
CMD >  
010D C0D003 RCL AL,03  
0110 C0D803 RCR AL,03  
0113 B8004C MOV AX,4C00  
0116 CD21 INT 21  
0118 8956E4 MOV [BP-1C],DX  
011B 8946E6 MOV [BP-1A],AX  
011E C746F60000 MOV [BP-0A],0000  
0123 8B46F6 MOV AX,[BP-0A]  
0126 D1E0 SHL AX,1  
  
DS:0000 CD 20 FF 9F 00 EA F0 FE  
DS:0008 AD DE 1B 05 C5 06 00 00  
DS:0010 18 01 10 01 18 01 92 01  
DS:0018 01 01 01 00 02 FF FF FF  
DS:0020 FF FF FF FF FF FF FF FF  
DS:0028 FF FF FF FF EB 19 C0 11  
DS:0030 A2 01 14 00 18 00 F5 19  
DS:0038 FF FF FF FF 00 00 00 00  
DS:0040 05 00 00 00 00 00 00 00  
DS:0048 00 00 00 00 00 00 00 00  
  
= f.Ω= i|..+. ...  
.....ff. ....  
δ.L.  
ó.....J. ....  
.....
```



COAL LABTASK 8

SAJID ISLAM
24P-0745

TASK 2:

Assembly Code:

```
ASM lb8q2.asm
1  [org 0x0100]
2  jmp start
3      multiplicand: db 12 ; 4-bit number
4      multiplier: db 6 ; 4-bit
5      result: db 0 ; 8-bit result
6  start:
7      mov cl, 5 ; how many times we need to run the loop
8      mov al, [multiplicand]
9      mov bl, [ multiplier]
10 bit_checker:
11     shr bl, 1
12     jnc move
13     add [result], bl ; only add if CF IS 1
14 move:
15     shl al, 1
16     dec cl
17     jnz bit_checker
18
19     mov ax, 0x4c00
20     int 0x21
```

- The multiplicand and multiplier are both 4-bit values, producing an 8-bit result.
- We load 5 into CL because one addition is already performed during initialization.
- The multiplicand and multiplier are moved into AL and BL, respectively.
- In the bit_checker loop, the multiplier is rotated right, and the carry flag is checked.
- If the carry flag is set, the multiplicand is added to the result;



COAL LABTASK 8

SAJID ISLAM
24P-0745

- If it is clear, the program jumps to the move loop, where the multiplicand is shifted left before returning to bit_checker for the next iteration.

TASK 3(1):

Assembly Code:

```
ASM lb8q31.asm
1  [org 0x0100]
2  jmp start
3      num1: dw 8AB0h
4      num2: dw 7D70h
5      high: dw 0000h
6  start:
7      mov ax, [num1]
8      add ax, [num2]      ; AX = 8AB0h + 7D70h = 10820h → AX=0820h, CF=1
9      mov word [num1], ax
10     adc word [high], 0  ; Add carry (1) to next word → high = 0001h
11
12     mov ax, 0x4c00
13     int 0x21
14     |
```

The values are stored in num1 and num2.

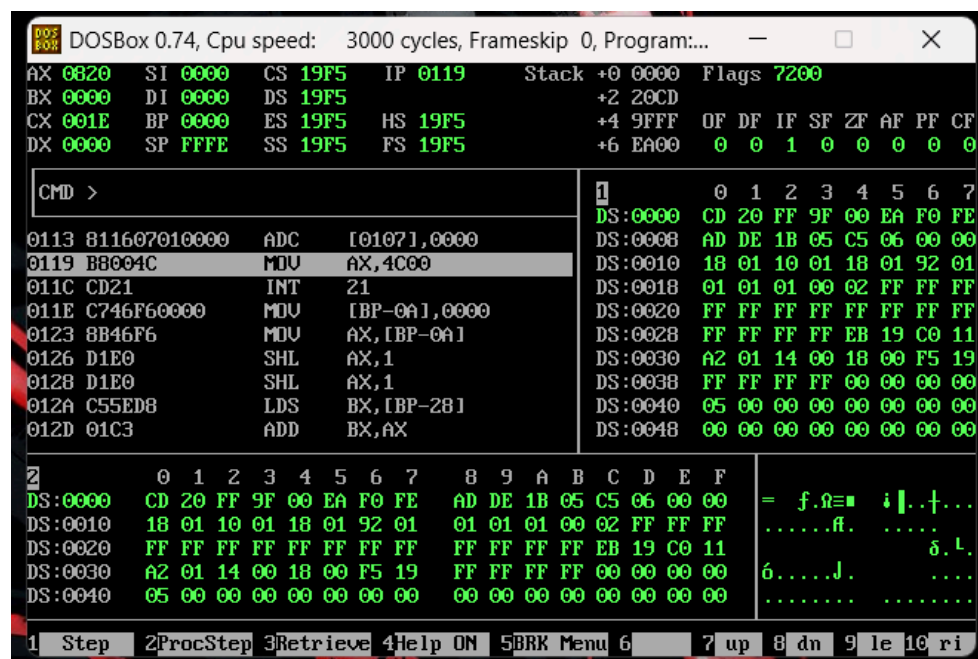
When num1 is loaded into AX and added to num2, an overflow (OF) occurs because the result exceeds the 16-bit limit.

The extra (17th) bit generated from the addition is stored in the Carry Flag (CF).

Afterward, the result in AX is moved back to num1, and the instruction ***adc word [high], 0*** adds the carry to high, effectively performing $high = high + CF$.



SAJID ISLAM
24P-0745





COAL LABTASK 8

SAJID ISLAM
24P-0745

TASK 3(2):

Assembly Code:

```
ASM lb8q32.asm
1  [org 0x0100]
2  jmp start
3      num1: dw 2000h
4      num2: dw 3000h
5      high: dw 0001h
6  start:
7      mov ax, [num1]
8      sub ax, [num2]
9      mov word [num1], ax
10     sbb word [high], 0    ; Subtract with borrow
11
12
13     mov ax, 0x4c00
14     int 0x21
15     |
```

- Values are stored in **num1** and **num2**.
- When **num1** is moved to **ax** and both numbers are subtracted, borrow is taken.
- We move **ax** to **num1** and the **borrow** is subtracted to **high** by using **sbb word [high],0** which means **high = high - borrow**.



COAL LABTASK 8

SAJID ISLAM

24P-0745

```
DOSBox 0.74, Cpu speed: 3000 cycles, Frameskip 0, Program:...  
AX F000 SI 0000 CS 19F5 IP 0110 Stack +0 0000 Flags 72B5  
BX 0000 DI 0000 DS 19F5 +2 20CD  
CX 001E BP 0000 ES 19F5 HS 19F5 +4 9FFF OF DF IF SF ZF AF PF CF  
DX 0000 SP FFFE SS 19F5 FS 19F5 +6 EA00 0 0 1 1 0 0 1 1  
CMD >  
010C 2B060501 SUB AX,[0105]  
0110 A30301 MOV [0103],AX  
0113 811E07010000 SBB [0107],0000  
0119 B8004C MOV AX,4C00  
011C CD21 INT 21  
011E C746F60000 MOV [BP-0A],0000  
0123 8B46F6 MOV AX,[BP-0A]  
0126 D1E0 SHL AX,1  
0128 D1E0 SHL AX,1  
DS:0000 CD 20 FF 9F 00 EA F0 FE  
DS:0008 AD DE 1B 05 C5 06 00 00  
DS:0010 18 01 10 01 18 01 92 01  
DS:0018 01 01 01 00 02 FF FF FF  
DS:0020 FF FF FF FF FF FF FF FF  
DS:0028 FF FF FF FF EB 19 C0 11  
DS:0030 A2 01 14 00 18 00 F5 19  
DS:0038 FF FF FF FF 00 00 00 00  
DS:0040 05 00 00 00 00 00 00 00  
DS:0048 00 00 00 00 00 00 00 00  
1 Step 2ProcStep 3Retrieve 4Help ON 5BRK Menu 6 7 up 8 dn 9 le 10 ri
```

```
DOSBox 0.74, Cpu speed: 3000 cycles, Frameskip 0, Program:...  
AX F000 SI 0000 CS 19F5 IP 0119 Stack +0 0000 Flags 7244  
BX 0000 DI 0000 DS 19F5 +2 20CD  
CX 001E BP 0000 ES 19F5 HS 19F5 +4 9FFF OF DF IF SF ZF AF PF CF  
DX 0000 SP FFFE SS 19F5 FS 19F5 +6 EA00 0 0 1 0 1 0 1 0  
CMD >  
0113 811E07010000 SBB [0107],0000  
0119 B8004C MOV AX,4C00  
011C CD21 INT 21  
011E C746F60000 MOV [BP-0A],0000  
0123 8B46F6 MOV AX,[BP-0A]  
0126 D1E0 SHL AX,1  
0128 D1E0 SHL AX,1  
012A C5ED8 LDS BX,[BP-28]  
012D 01C3 ADD BX,AX  
DS:0000 CD 20 FF 9F 00 EA F0 FE  
DS:0008 AD DE 1B 05 C5 06 00 00  
DS:0010 18 01 10 01 18 01 92 01  
DS:0018 01 01 01 00 02 FF FF FF  
DS:0020 FF FF FF FF FF FF FF FF  
DS:0028 FF FF FF FF EB 19 C0 11  
DS:0030 A2 01 14 00 18 00 F5 19  
DS:0038 FF FF FF FF 00 00 00 00  
DS:0040 05 00 00 00 00 00 00 00  
DS:0048 00 00 00 00 00 00 00 00  
1 Step 2ProcStep 3Retrieve 4Help ON 5BRK Menu 6 7 up 8 dn 9 le 10 ri
```




COAL LABTASK 8

SAJID ISLAM
24P-0745

TASK 4:

Assembly Code:

```
ASM lb8q4.asm
1  [org 0x0100]
2      jmp start
3      val1: db 10101010b
4      val2: db 11001100b
5      result: db 0
6  start:
7      mov al, [val1]
8      mov bl, [val2]
9      and al, bl
10
11     mov [result], al      ; Store AND result
12     mov al, [val1]
13     or al, bl
14
15     mov [result], al      ; Store OR result
16     mov al, [val1]
17     xor al, bl
18
19     mov [result], al      ; Store XOR result
20     mov al, [val1]
21     not al
22
23     mov [result], al      ; Store NOT result
24
25     mov ax, 0x4c00
26     int 0x21
27
```



COAL LABTASK 8

SAJID ISLAM
24P-0745

- val1 and val2 are stored in al and bl.
- When **and al,bl** is run it will check if both are 1 then output 1 else so in our case **10001000** will be stored in the result.
- When **or al,bl** is run it will check if one bit is 1 then output 1 so **10001000** will be stored in the result.
- **xor al,bl** checks if opposite bits then 1 else 0 so the result will store **01100110**
- **not al** will just reverse the bits make 1->0 and 0->1 so the output will store **01010101**

Original:

```
DOSBox 0.74, Cpu speed: 3000 cycles, Frameskip 0, Program:...
AX 00AA SI 0000 CS 19F5 IP 010D Stack +0 0000 Flags 7200
BX 00CC DI 0000 DS 19F5      +2 20CD
CX 002F BP 0000 ES 19F5 HS 19F5  +4 9FFF OF DF IF SF ZF AF PF CF
DX 0000 SP FFFE SS 19F5 FS 19F5  +6 EA00 0 0 1 0 0 0 0 0

CMD >

0109 8A1E0401      MOV     BL,[0104]
010D 20D8          AND     AL,BL
010F A20501      MOV     [0105],AL
0112 A00301      MOV     AL,[0103]
0115 08D8          OR      AL,BL
0117 A20501      MOV     [0105],AL
011A A00301      MOV     AL,[0103]
011D 30D8          XOR     AL,BL
011F A20501      MOV     [0105],AL

1
0 1 2 3 4 5 6 7
DS:0000 CD 20 FF 9F 00 EA F0 FE
DS:0008 AD DE 1B 05 C5 06 00 00
DS:0010 18 01 10 01 18 01 92 01
DS:0018 01 01 01 00 02 FF FF FF
DS:0020 FF FF FF FF FF FF FF FF
DS:0028 FF FF FF FF EB 19 C0 11
DS:0030 A2 01 14 00 18 00 F5 19
DS:0038 FF FF FF FF 00 00 00 00
DS:0040 05 00 00 00 00 00 00 00
DS:0048 00 00 00 00 00 00 00 00

2
0 1 2 3 4 5 6 7 8 9 A B C D E F
DS:0000 CD 20 FF 9F 00 EA F0 FE AD DE 1B 05 C5 06 00 00 = f.~  i |..+...
DS:0010 18 01 10 01 18 01 92 01 01 01 01 00 02 FF FF FF .....f. ....
DS:0020 FF FF FF FF FF FF FF FF FF FF FF FF EB 19 C0 11 .....L.
DS:0030 A2 01 14 00 18 00 F5 19 FF FF FF FF 00 00 00 00 6.....J. ....
DS:0040 05 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....

1 Step 2ProcStep 3Retrieve 4Help ON 5BRK Menu 6 7 up 8 dn 9 le 10 ri
```



COAL LABTASK 8

SAJID ISLAM
24P-0745

AND:

```
DOSBox 0.74, Cpu speed: 3000 cycles, Frameskip 0, Program:...  
AX 0088 SI 0000 CS 19F5 IP 0112 Stack +0 0000 Flags 7284  
BX 00CC DI 0000 DS 19F5 +2 20CD  
CX 002F BP 0000 ES 19F5 HS 19F5 +4 9FFF OF DF IF SF ZF AF PF CF  
DX 0000 SP FFFE SS 19F5 FS 19F5 +6 EA00 0 0 1 1 0 0 1 0  
  
CMD >  
1 0 1 2 3 4 5 6 7  
DS:0000 CD 20 FF 9F 00 EA F0 FE  
DS:0008 AD DE 1B 05 C5 06 00 00  
DS:0010 18 01 10 01 18 01 92 01  
DS:0018 01 01 01 00 02 FF FF FF  
DS:0020 FF FF FF FF FF FF FF FF  
DS:0028 FF FF FF FF EB 19 C0 11  
DS:0030 A2 01 14 00 18 00 F5 19  
DS:0038 FF FF FF FF 00 00 00 00  
DS:0040 05 00 00 00 00 00 00 00  
DS:0048 00 00 00 00 00 00 00 00  
  
2 0 1 2 3 4 5 6 7 8 9 A B C D E F  
DS:0000 CD 20 FF 9F 00 EA F0 FE AD DE 1B 05 C5 06 00 00 = f.Ω≡ i |..†...  
DS:0010 18 01 10 01 18 01 92 01 01 01 01 00 02 FF FF FF .....ff. ....  
DS:0020 FF FF FF FF FF FF FF FF FF FF FF FF FF EB 19 C0 11 δ.L.  
DS:0030 A2 01 14 00 18 00 F5 19 FF FF FF FF 00 00 00 00 6.....J. ....  
DS:0040 05 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....  
  
1 Step 2ProcStep 3Retrieve 4Help ON 5BRK Menu 6 7 up 8 dn 9 le 10 ri
```



COAL LABTASK 8

SAJID ISLAM
24P-0745

OR:

```
DOSBox 0.74, Cpu speed: 3000 cycles, Frameskip 0, Program:...
AX 00EE SI 0000 CS 19F5 IP 0117 Stack +0 0000 Flags 7284
BX 00CC DI 0000 DS 19F5 +2 20CD
CX 002F BP 0000 ES 19F5 HS 19F5 +4 9FFF OF DF IF SF ZF AF PF CF
DX 0000 SP FFFE SS 19F5 FS 19F5 +6 EA00 0 0 1 1 0 0 1 0

CMD >

0115 08D8 OR AL,BL
0117 A20501 MOV [0105],AL
011A A00301 MOV AL,[0103]
011D 30D8 XOR AL,BL
011F A20501 MOV [0105],AL
0122 A00301 MOV AL,[0103]
0125 F6D0 NOT AL
0127 A20501 MOV [0105],AL
012A B8004C MOV AX,4C00

1 0 1 2 3 4 5 6 7
DS:0000 CD 20 FF 9F 00 EA F0 FE
DS:0008 AD DE 1B 05 C5 06 00 00
DS:0010 18 01 10 01 18 01 92 01
DS:0018 01 01 01 00 02 FF FF FF
DS:0020 FF FF FF FF FF FF FF FF
DS:0028 FF FF FF FF EB 19 C0 11
DS:0030 A2 01 14 00 18 00 F5 19
DS:0038 FF FF FF FF 00 00 00 00
DS:0040 05 00 00 00 00 00 00 00
DS:0048 00 00 00 00 00 00 00 00

2 0 1 2 3 4 5 6 7 8 9 A B C D E F
DS:0000 CD 20 FF 9F 00 EA F0 FE AD DE 1B 05 C5 06 00 00 = f.Ω≡ i |..†...
DS:0010 18 01 10 01 18 01 92 01 01 01 01 00 02 FF FF FF .....ff. ....
DS:0020 FF FF FF FF FF FF FF FF FF FF FF FF FF EB 19 C0 11 .....δ.L.
DS:0030 A2 01 14 00 18 00 F5 19 FF FF FF FF 00 00 00 00 ó.....J. ....
DS:0040 05 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....

1 Step 2ProcStep 3Retrieve 4Help ON 5BRK Menu 6 7 up 8 dn 9 le 10 ri
```



COAL LABTASK 8

SAJID ISLAM
24P-0745

XOR:

```
DOSBox 0.74, Cpu speed: 3000 cycles, Frameskip 0, Program:...
AX 0066 SI 0000 CS 19F5 IP 011F Stack +0 0000 Flags 7204
BX 00CC DI 0000 DS 19F5 +2 20CD
CX 002F BP 0000 ES 19F5 HS 19F5 +4 9FFF OF DF IF SF ZF AF PF CF
DX 0000 SP FFFE SS 19F5 FS 19F5 +6 EA00 0 0 1 0 0 0 1 0

CMD >

011D 30D8 XOR AL,BL
011F A20501 MOV [0105],AL
0122 A00301 MOV AL,[0103]
0125 F6D0 NOT AL
0127 A20501 MOV [0105],AL
012A B8004C MOV AX,4C00
012D CD21 INT 21
012F 8B07 MOV AX,[BX]
0131 8B5702 MOV DX,[BX+02]

1 0 1 2 3 4 5 6 7
DS:0000 CD 20 FF 9F 00 EA F0 FE
DS:0008 AD DE 1B 05 C5 06 00 00
DS:0010 18 01 10 01 18 01 92 01
DS:0018 01 01 01 00 02 FF FF FF
DS:0020 FF FF FF FF FF FF FF FF
DS:0028 FF FF FF FF EB 19 C0 11
DS:0030 A2 01 14 00 18 00 F5 19
DS:0038 FF FF FF FF 00 00 00 00
DS:0040 05 00 00 00 00 00 00 00
DS:0048 00 00 00 00 00 00 00 00

2 0 1 2 3 4 5 6 7 8 9 A B C D E F
DS:0000 CD 20 FF 9F 00 EA F0 FE AD DE 1B 05 C5 06 00 00 = f.Ω≡ i |..†...
DS:0010 18 01 10 01 18 01 92 01 01 01 01 00 02 FF FF FF .....ff. ....
DS:0020 FF FF FF FF FF FF FF FF FF FF FF FF EB 19 C0 11 .....δ.ℓ.
DS:0030 A2 01 14 00 18 00 F5 19 FF FF FF FF 00 00 00 00 6.....J. ....
DS:0040 05 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....

1 Step 2ProcStep 3Retrieve 4Help ON 5BRK Menu 6 7 up 8 dn 9 le 10 ri
```



COAL LABTASK 8

SAJID ISLAM
24P-0745

NOT:

```
DOSBox 0.74, Cpu speed: 3000 cycles, Frameskip 0, Program:...
AX 0055 SI 0000 CS 19F5 IP 0127 Stack +0 0000 Flags 7204
BX 00CC DI 0000 DS 19F5      +2 20CD
CX 002F BP 0000 ES 19F5 HS 19F5  +4 9FFF 0F DF IF SF ZF AF PF CF
DX 0000 SP FFFE SS 19F5 FS 19F5  +6 EA00 0 0 1 0 0 0 1 0

CMD >

0125 F6D0      NOT     AL
0127 A20501     MOV     [0105],AL
012A B8004C     MOV     AX,4C00
012D CD21      INT     21
012F 8B07      MOV     AX,[BX]
0131 8B5702     MOV     DX,[BX+02]
0134 85D2      TEST    DX,DX
0136 7504      JNZ     013C
0138 85C0      TEST    AX,AX

1  0 1 2 3 4 5 6 7
DS:0000 CD 20 FF 9F 00 EA F0 FE
DS:0008 AD DE 1B 05 C5 06 00 00
DS:0010 18 01 10 01 18 01 92 01
DS:0018 01 01 01 00 02 FF FF FF
DS:0020 FF FF FF FF FF FF FF FF
DS:0028 FF FF FF FF EB 19 C0 11
DS:0030 A2 01 14 00 18 00 F5 19
DS:0038 FF FF FF FF 00 00 00 00
DS:0040 05 00 00 00 00 00 00 00
DS:0048 00 00 00 00 00 00 00 00

2  0 1 2 3 4 5 6 7 8 9 A B C D E F
DS:0000 CD 20 FF 9F 00 EA F0 FE AD DE 1B 05 C5 06 00 00
DS:0010 18 01 10 01 18 01 92 01 01 01 01 00 02 FF FF FF
DS:0020 FF FF FF FF FF FF FF FF FF FF FF FF EB 19 C0 11
DS:0030 A2 01 14 00 18 00 F5 19 FF FF FF FF 00 00 00 00
DS:0040 05 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

= f.Ω≡  i |..†...
.....ff. ....
δ.L.
ó.....J. ....
.....
```