



National University
Of Computer and Emerging Sciences

EE2003 – Computer Organization and Assembly Language (Fall 2023) 10 Oct 2023

Assignment: 03, Marks: 10, Due Date: 20 minutes, CLO: 3 Teacher: Dr Muhammad Usman Abbasi

Name: Ahmed – Mustafa Roll No. 22P-9060 Section: BS-CS-3A

1. Make separate subroutines for add, subtract, multiply and divide and then perform all these operations between two numbers of your choice using these subroutines and passing them the numbers as parameters on the stack. Also store the results for each of the operation in the variables shown in the starter code.

Answer:

Code:

```

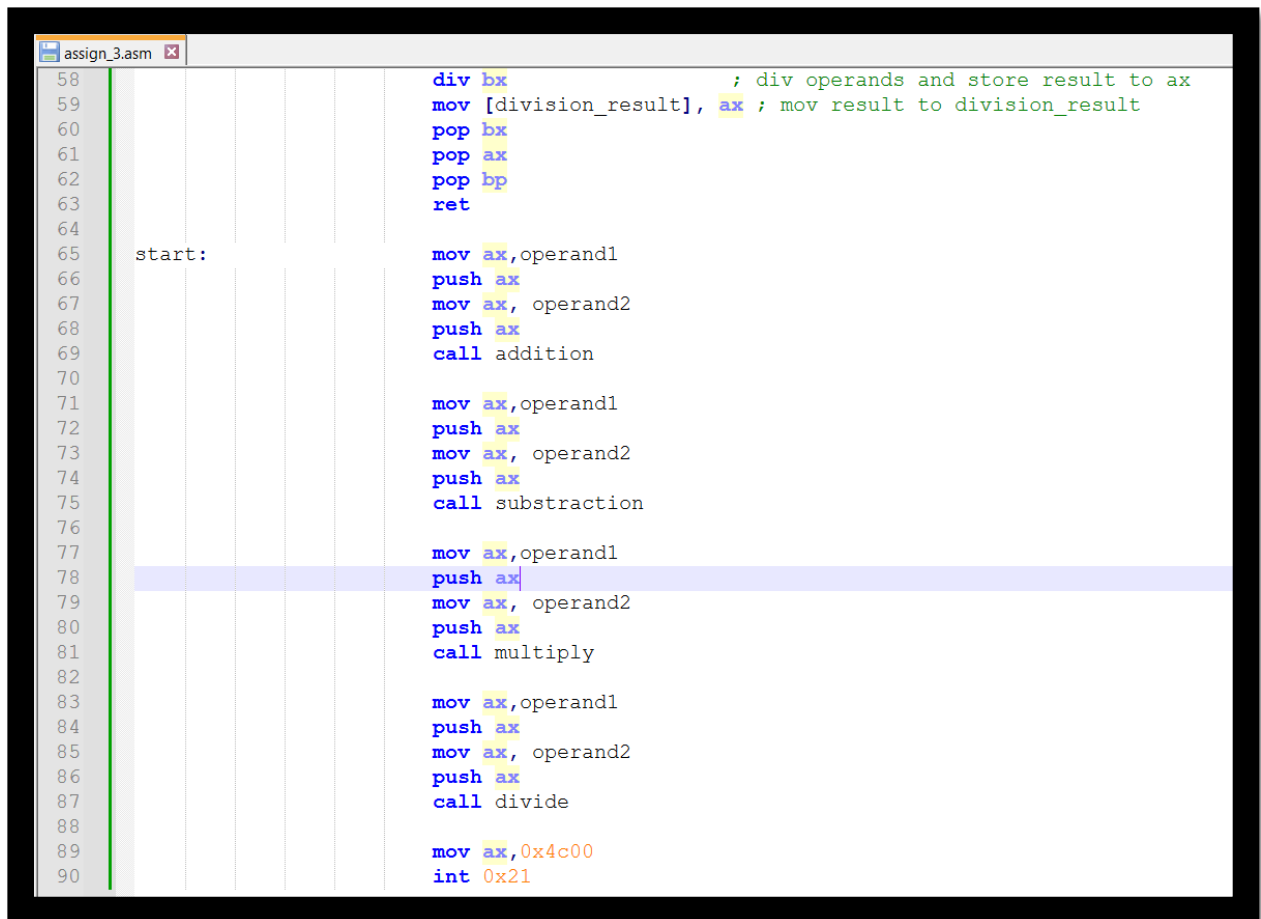
37 ; multiply subroutine
38 multiply:
39     push bp                ; save old value of base pointer
40     mov bp,sp              ; mov base pointer to stack pointer
41     push ax                ; save old value of ax register
42     push bx                ; save old value of bx register
43     mov ax,[bp+6]          ; mov operand1 to ax from stack
44     mov bx,[bp+4]          ; mov operand2 to bx from stack
45     imul ax,bx             ; mul both operands and store result to ax
46     mov [multiplication_result], ax ;mov result to multiplication_result
47     pop bx
48     pop ax
49     pop bp
50     ret
51 ; divide subroutine
52 divide:
53     push bp                ; save old value of base pointer
54     mov bp,sp              ; mov base pointer to stack pointer
55     push ax                ; save old value of ax register
56     push bx                ; save old value of bx register
57     mov ax,[bp+6]          ; mov operand1 to ax from stack
58     mov bx,[bp+4]          ; mov operand2 to bx from stack
59     xor dx,dx              ; clear dx
60     div bx                 ; div operands and store result to ax
61     mov [division_result], ax ; mov result to division_result
62     pop bx
63     pop ax
64     pop bp
65     ret
66 start:
67     mov ax,operand1
68     push ax
69     mov ax, operand2
70     push ax
71     call addition
72     mov ax,operand1
73     push ax

```

```

1  [org 0x0100]
2
3  jmp start ; jump to start label
4
5  operand1: dw 20
6  operand2: dw 12
7  sum_result: dw 0
8  subtraction_result: dw 0
9  multiplication_result: dw 0
10 division_result: dw 0
11 ; addition subroutine
12 addition:      push bp          ; save old value of base pointer
13                mov bp,sp        ; mov base pointer to stack pointer
14                push ax          ; save old value of ax register
15                push bx          ; save old value of bx register
16                mov ax,[bp+6]     ; mov operand1 to ax from stack
17                mov bx,[bp+4]     ; mov operand2 to bx from stack
18                add ax,bx         ; add both operands and store result to ax
19                mov [sum_result],ax ; mov result to sum_result
20                pop bx
21                pop ax
22                pop bp
23                ret              ; return back using instruction pointer
24 ; subtraction subroutine
25 subtraction:   push bp          ; save old value of base pointer
26                mov bp,sp        ; mov base pointer to stack pointer
27                push ax          ; save old value of ax register
28                push bx          ; save old value of bx register
29                mov ax,[bp+6]     ; mov operand1 to ax from stack
30                mov bx,[bp+4]     ; mov operand2 to bx from stack
31                sub ax,bx         ; sub both operands and store result to ax
32                mov [subtraction_result],ax ; mov result to sub_result
33                pop bx
34                pop ax
35                pop bp
36                ret

```



```
58      div bx          ; div operands and store result to ax
59      mov [division_result], ax ; mov result to division_result
60      pop bx
61      pop ax
62      pop bp
63      ret
64
65      start:          mov ax,operand1
66                      push ax
67                      mov ax, operand2
68                      push ax
69                      call addition
70
71                      mov ax,operand1
72                      push ax
73                      mov ax, operand2
74                      push ax
75                      call subtraction
76
77                      mov ax,operand1
78                      push ax
79                      mov ax, operand2
80                      push ax
81                      call multiply
82
83                      mov ax,operand1
84                      push ax
85                      mov ax, operand2
86                      push ax
87                      call divide
88
89                      mov ax,0x4c00
90                      int 0x21
```

Explanation:

This assembly code defines a program to perform arithmetic operations on operand1 and operand2 using the stack for subroutine calls. It initializes operands and result variables, with four subroutines (addition, subtraction, multiply, and divide) following a pattern of stack preservation, operation execution, and result storage.

In the start section, the program calls each arithmetic subroutine sequentially, utilizing the stack for parameter passing. The DOS interrupt int 0x21 is employed for program termination. The stack serves as a crucial mechanism for managing data and control flow during subroutine execution.

Screen-Shot of Debugger:

DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Progra... — □ ×

AX 0000	SI 0000	CS 19F5	IP 0100	Stack +0 0000	Flags 7202
BX 0000	DI 0000	DS 19F5		+2 20CD	
CX 0093	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 >

0100 E95F00	JMP	0162
0103 1400	ADC	AL,00
0105 0C00	OR	AL,00
0107 0000	ADD	[BX+SI],AL
0109 0000	ADD	[BX+SI],AL
010B 0000	ADD	[BX+SI],AL
010D 0000	ADD	[BX+SI],AL
010F 55	PUSH	BP

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 FF 00 01 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	FF	00	01	FFf.
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	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

DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Progra... — □ ×

AX 0000	SI 0000	CS 19F5	IP 0162	Stack +0 0000	Flags 7200
BX 0000	DI 0000	DS 19F5		+2 20CD	
CX 0093	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

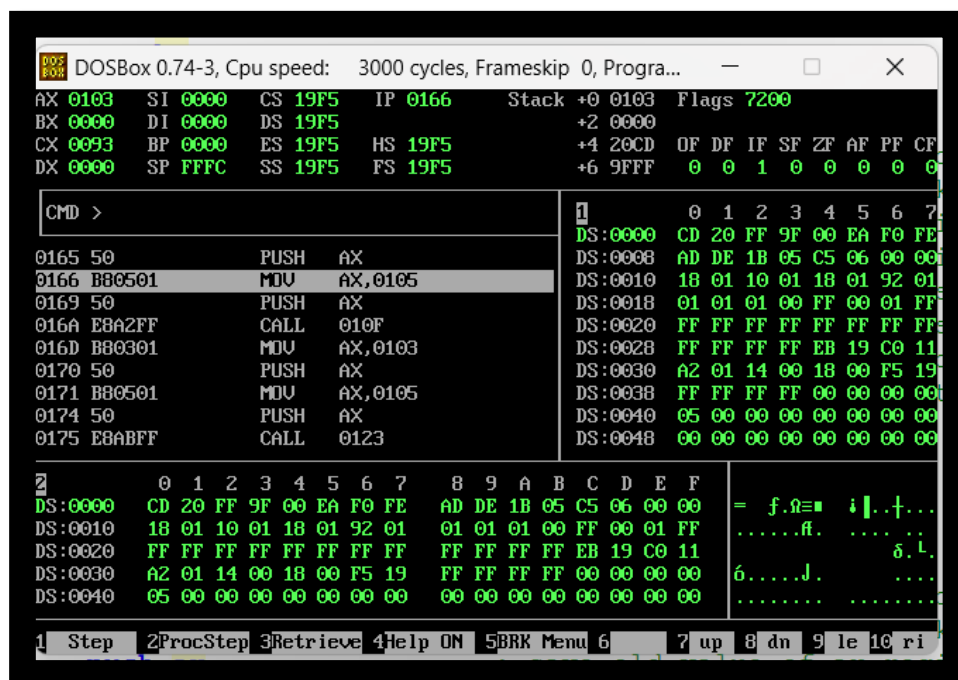
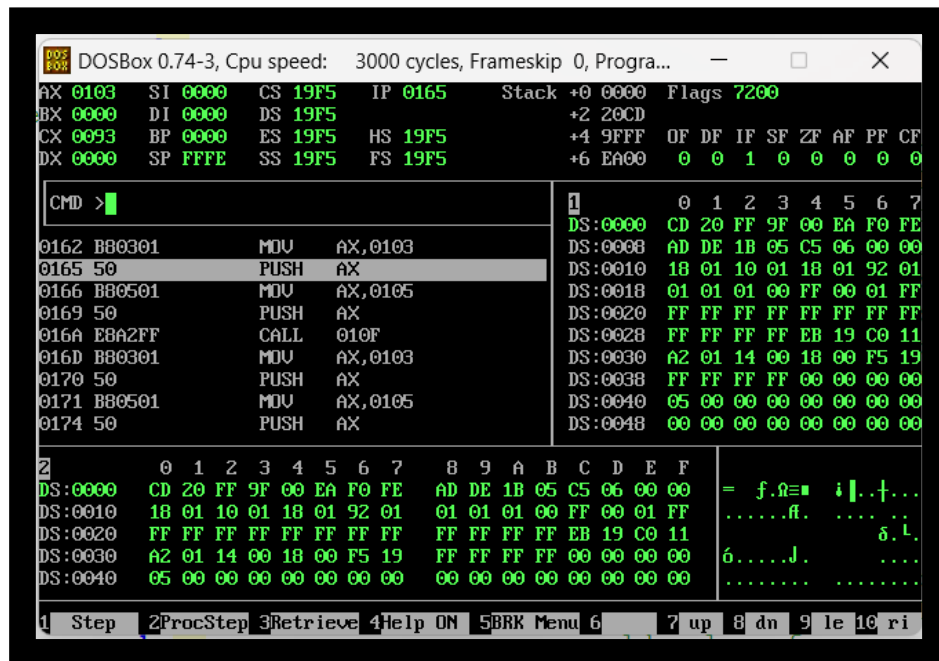
CMD >

0100 E95F00	JMP	0162
0162 B80301	MOV	AX,0103
0165 50	PUSH	AX
0166 B80501	MOV	AX,0105
0169 50	PUSH	AX
016A E8A2FF	CALL	010F
016D B80301	MOV	AX,0103
0170 50	PUSH	AX
0171 B80501	MOV	AX,0105

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 FF 00 01 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	FF	00	01	FFf.
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	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



```
DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Progra...
AX 0105 SI 0000 CS 19F5 IP 0169 Stack +0 0103 Flags 7200
BX 0000 DI 0000 DS 19F5 +2 0000
CX 0093 BP 0000 ES 19F5 HS 19F5 +4 20CD OF DF IF SF ZF AF PF CF
DX 0000 SP FFFC SS 19F5 FS 19F5 +6 9FFF 0 0 1 0 0 0 0 0

CMD >

3166 B80501 MOV AX,0105
3169 50 PUSH AX
316A E8A2FF CALL 010F
316D B80301 MOV AX,0103
3170 50 PUSH AX
3171 B80501 MOV AX,0105
3174 50 PUSH AX
3175 E8ABFF CALL 0123
3178 B80301 MOV AX,0103

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 FF 00 01 FF .....f. ....
DS:0020 FF FF FF FF FF FF FF FF FF FF FF FF EB 19 C0 11 .....δ.
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 .....

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

```
DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Progra...
AX 0105 SI 0000 CS 19F5 IP 016A Stack +0 0105 Flags 7200
BX 0000 DI 0000 DS 19F5 +2 0103
CX 0093 BP 0000 ES 19F5 HS 19F5 +4 0000 OF DF IF SF ZF AF PF CF
DX 0000 SP FFFA SS 19F5 FS 19F5 +6 20CD 0 0 1 0 0 0 0 0

CMD >

0169 50 PUSH AX
016A E8A2FF CALL 010F
016D B80301 MOV AX,0103
0170 50 PUSH AX
0171 B80501 MOV AX,0105
0174 50 PUSH AX
0175 E8ABFF CALL 0123
0178 B80301 MOV AX,0103
017B 50 PUSH AX

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 FF 00 01 FF .....f. ....
DS:0020 FF FF FF FF FF FF FF FF FF FF FF FF EB 19 C0 11 .....δ.
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 .....

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

DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Progra...

AX 0103	SI 0000	CS 19F5	IP 0171	Stack +0 0103	Flags 7200
BX 0000	DI 0000	DS 19F5		+2 0105	
CX 0093	BP 0000	ES 19F5	HS 19F5	+4 0103	OF DF IF SF ZF AF PF CF
DX 0000	SP FFF8	SS 19F5	FS 19F5	+6 0000	0 0 1 0 0 0 0 0

CMD >

0170 50	PUSH	AX
0171 B80501	MOV	AX,0105
0174 50	PUSH	AX
0175 E8ABFF	CALL	0123
0178 B80301	MOV	AX,0103
017B 50	PUSH	AX
017C B80501	MOV	AX,0105
017F 50	PUSH	AX
0180 E8B4FF	CALL	0137

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	FF	00	01	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	FF	00	01	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.
6.....J.
.....

1 Step

2 ProcStep

3 Retrieve

4 Help ON

5 BRK Menu

6

7 up

8 dn

9 le

10 ri

DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Progra...

AX 0105	SI 0000	CS 19F5	IP 0174	Stack +0 0103	Flags 7200
BX 0000	DI 0000	DS 19F5		+2 0105	
CX 0093	BP 0000	ES 19F5	HS 19F5	+4 0103	OF DF IF SF ZF AF PF CF
DX 0000	SP FFF8	SS 19F5	FS 19F5	+6 0000	0 0 1 0 0 0 0 0

CMD >

0171 B80501	MOV	AX,0105
0174 50	PUSH	AX
0175 E8ABFF	CALL	0123
0178 B80301	MOV	AX,0103
017B 50	PUSH	AX
017C B80501	MOV	AX,0105
017F 50	PUSH	AX
0180 E8B4FF	CALL	0137
0183 B80301	MOV	AX,0103

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	FF	00	01	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	FF	00	01	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.
6.....J.
.....

1 Step

2 ProcStep

3 Retrieve

4 Help ON

5 BRK Menu

6

7 up

8 dn

9 le

10 ri

DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Progra... — □ ×

AX 0105	SI 0000	CS 19F5	IP 0175	Stack +0 0105	Flags 7200
BX 0000	DI 0000	DS 19F5		+2 0103	
CX 0093	BP 0000	ES 19F5	HS 19F5	+4 0105	OF DF IF SF ZF AF PF CF
DX 0000	SP FFF6	SS 19F5	FS 19F5	+6 0103	0 0 1 0 0 0 0 0

CMD >

0174 50	PUSH	AX
0175 E8ABFF	CALL	0123
0178 B80301	MOV	AX,0103
017B 50	PUSH	AX
017C B80501	MOV	AX,0105
017F 50	PUSH	AX
0180 E8B4FF	CALL	0137
0183 B80301	MOV	AX,0103
0186 50	PUSH	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	FF	00	01	FF
DS:0020	FF	FF	FF	FF	FF	FF	FF	FF
DS:0028	FF	FF	FF	FF	FF	EB	19	C0
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	FF	00	01	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 |..†...
.....f.
δ.L.
6.....J.
.....

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

DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Progra... — □ ×

AX 0105	SI 0000	CS 19F5	IP 0178	Stack +0 0105	Flags 7291
BX 0000	DI 0000	DS 19F5		+2 0103	
CX 0093	BP 0000	ES 19F5	HS 19F5	+4 0105	OF DF IF SF ZF AF PF CF
DX 0000	SP FFF6	SS 19F5	FS 19F5	+6 0103	0 0 1 1 0 1 0 1

CMD >

0175 E8ABFF	CALL	0123
0178 B80301	MOV	AX,0103
017B 50	PUSH	AX
017C B80501	MOV	AX,0105
017F 50	PUSH	AX
0180 E8B4FF	CALL	0137
0183 B80301	MOV	AX,0103
0186 50	PUSH	AX
0187 B80501	MOV	AX,0105

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	FF	00	01	FF
DS:0020	FF	FF	FF	FF	FF	FF	FF	FF
DS:0028	FF	FF	FF	FF	FF	EB	19	C0
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	FF	00	01	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 |..†...
.....f.
δ.L.
6.....J.
.....

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

DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Progra...

AX	0103	SI	0000	CS	19F5	IP	017B	Stack	+0 0105	Flags	7291
BX	0000	DI	0000	DS	19F5				+2 0103		
CX	0093	BP	0000	ES	19F5	HS	19F5		+4 0105	OF DF IF SF ZF AF PF CF	
DX	0000	SP	FFF6	SS	19F5	FS	19F5		+6 0103	0 0 1 1 0 1 0 1	

CMD >

0178 B80301	MOV	AX,0103
017B 50	PUSH	AX
017C B80501	MOV	AX,0105
017F 50	PUSH	AX
0180 E8B4FF	CALL	0137
0183 B80301	MOV	AX,0103
0186 50	PUSH	AX
0187 B80501	MOV	AX,0105
018A 50	PUSH	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 FF 00 01 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	FF	00	01	FFf.
DS:0020	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	EB	19	C0	11δ.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

DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Progra...

AX	0103	SI	0000	CS	19F5	IP	017C	Stack	+0 0103	Flags	7291
BX	0000	DI	0000	DS	19F5				+2 0105		
CX	0093	BP	0000	ES	19F5	HS	19F5		+4 0103	OF DF IF SF ZF AF PF CF	
DX	0000	SP	FFF4	SS	19F5	FS	19F5		+6 0105	0 0 1 1 0 1 0 1	

CMD >

017B 50	PUSH	AX
017C B80501	MOV	AX,0105
017F 50	PUSH	AX
0180 E8B4FF	CALL	0137
0183 B80301	MOV	AX,0103
0186 50	PUSH	AX
0187 B80501	MOV	AX,0105
018A 50	PUSH	AX
018B E8BEFF	CALL	014C

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 FF 00 01 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	FF	00	01	FFf.
DS:0020	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	EB	19	C0	11δ.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

DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Progra...

AX 0105	SI 0000	CS 19F5	IP 0183	Stack +0 0105	Flags 7A91
BX 0000	DI 0000	DS 19F5		+2 0103	
CX 0093	BP 0000	ES 19F5	HS 19F5	+4 0105	OF DF IF SF ZF AF PF CF
DX 0000	SP FFF2	SS 19F5	FS 19F5	+6 0103	1 0 1 1 0 1 0 1

CMD >

0180 E8B4FF	CALL	0137
0183 B80301	MOV	AX,0103
0186 50	PUSH	AX
0187 B80501	MOV	AX,0105
018A 50	PUSH	AX
018B E8BEFF	CALL	014C
018E B8004C	MOV	AX,4C00
0191 CD21	INT	21
0193 0485	ADD	AL,85

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	FF	00	01	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

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	FF	00	01	FFf.	
DS:0020	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	EB	19	C0	11δ. L.
DS:0030	A2	01	14	00	18	00	F5	19	FF	FF	FF	FF	00	00	00	00	00	6.....J.
DS:0040	05	00	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

DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Progra...

AX 0103	SI 0000	CS 19F5	IP 0186	Stack +0 0105	Flags 7A91
BX 0000	DI 0000	DS 19F5		+2 0103	
CX 0093	BP 0000	ES 19F5	HS 19F5	+4 0105	OF DF IF SF ZF AF PF CF
DX 0000	SP FFF2	SS 19F5	FS 19F5	+6 0103	1 0 1 1 0 1 0 1

CMD >

0183 B80301	MOV	AX,0103
0186 50	PUSH	AX
0187 B80501	MOV	AX,0105
018A 50	PUSH	AX
018B E8BEFF	CALL	014C
018E B8004C	MOV	AX,4C00
0191 CD21	INT	21
0193 0485	ADD	AL,85
0195 F67434	DIV	B/ISI+341

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	FF	00	01	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

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	FF	00	01	FFf.	
DS:0020	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	EB	19	C0	11δ. L.
DS:0030	A2	01	14	00	18	00	F5	19	FF	FF	FF	FF	00	00	00	00	00	6.....J.
DS:0040	05	00	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

DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Progra...

AX 0103 SI 0000 CS 19F5 IP 0187 Stack +0 0103 Flags 7A91
BX 0000 DI 0000 DS 19F5 +2 0105
CX 0093 BP 0000 ES 19F5 HS 19F5 +4 0103 OF DF IF SF ZF AF PF CF
DX 0000 SP FFF0 SS 19F5 FS 19F5 +6 0105 1 0 1 1 0 1 0 1

CMD >

0186 50 PUSH AX
0187 B80501 MOV AX,0105
018A 50 PUSH AX
018B EB00 CALL 014C
018E B8004C MOV AX,4C00
0191 CD21 INT 21
0193 0485 ADD AL,85
0195 F67434 DIV B:[SI+34]
0198 8E5EFE MOV DS,[BP-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 FF 00 01 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 FF 00 01 FFf.
DS:0020 FF FF FF FF FF FF FF FF FF FF FF FF EB 19 C0 11δ.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

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

DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Progra...

AX 0105 SI 0000 CS 19F5 IP 018A Stack +0 0103 Flags 7A91
BX 0000 DI 0000 DS 19F5 +2 0105
CX 0093 BP 0000 ES 19F5 HS 19F5 +4 0103 OF DF IF SF ZF AF PF CF
DX 0000 SP FFF0 SS 19F5 FS 19F5 +6 0105 1 0 1 1 0 1 0 1

CMD >

0187 B80501 MOV AX,0105
018A 50 PUSH AX
018B EB00 CALL 014C
018E B8004C MOV AX,4C00
0191 CD21 INT 21
0193 0485 ADD AL,85
0195 F67434 DIV B:[SI+34]
0198 8E5EFE MOV DS,[BP-02]
019B 837C0C0B CMP [SI+0C],000B

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 FF 00 01 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 FF 00 01 FFf.
DS:0020 FF FF FF FF FF FF FF FF FF FF FF FF EB 19 C0 11δ.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

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

DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Progra...

AX 0105	SI 0000	CS 19F5	IP 018B	Stack +0 0105	Flags 7A91
BX 0000	DI 0000	DS 19F5		+2 0103	
CX 0093	BP 0000	ES 19F5	HS 19F5	+4 0105	OF DF IF SF ZF AF PF CF
DX 0000	SP FFEE	SS 19F5	FS 19F5	+6 0103	1 0 1 1 0 1 0

CMD >		0	1	2	3	4	5	6	7
018A 50	PUSH	AX							
018B E8BEFF	CALL	014C							
018E B8004C	MOV	AX,4C00							
0191 CD21	INT	21							
0193 0485	ADD	AL,85							
0195 F67434	DIV	B/[SI+34]							
0198 8E5EFE	MOV	DS,[BP-02]							
019B 837C0C08	CMP	[SI+0C],0008							
019F 752B	JNZ	01CC							

	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	FF	00	01	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

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

DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Progra...

AX 0105	SI 0000	CS 19F5	IP 018E	Stack +0 0105	Flags 7244
BX 0000	DI 0000	DS 19F5		+2 0103	
CX 0093	BP 0000	ES 19F5	HS 19F5	+4 0105	OF DF IF SF ZF AF PF CF
DX 0103	SP FFEE	SS 19F5	FS 19F5	+6 0103	0 0 1 0 1 0 1 0

CMD >		0	1	2	3	4	5	6	7
018B E8BEFF	CALL	014C							
018E B8004C	MOV	AX,4C00							
0191 CD21	INT	21							
0193 0485	ADD	AL,85							
0195 F67434	DIV	B/[SI+34]							
0198 8E5EFE	MOV	DS,[BP-02]							
019B 837C0C08	CMP	[SI+0C],0008							
019F 752B	JNZ	01CC							
01A1 8B4404	MOV	AX,[SI+04]							

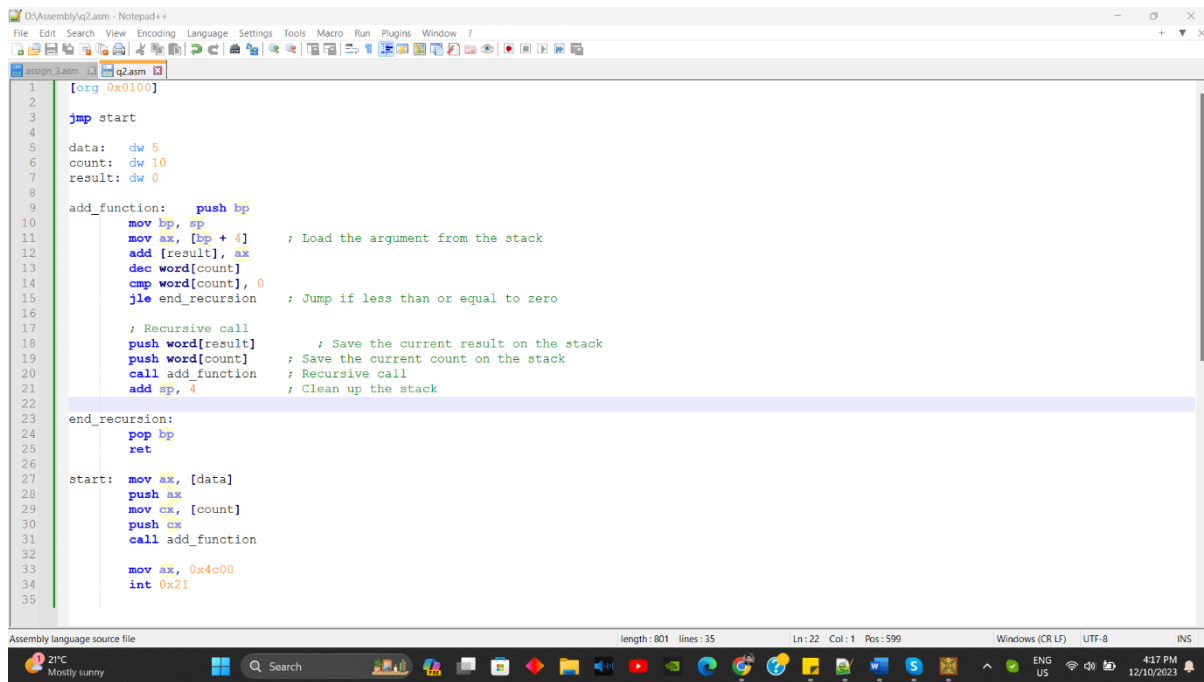
	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
DS:0000	CD	20	FF	9F	00	EA	F0	FE	AD	DE	1B	05	C5	06	00	00
DS:0010	18	01	10	01	18	01	92	01	01	01	01	00	FF	00	01	FF
DS:0020	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	EB	19	C0	11
DS:0030	A2	01	14	00	18	00	F5	19	FF	FF	FF	FF	00	00	00	00
DS:0040	05	00	00	00	00	00	00	00	00	00	00	00	00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

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

2. Perform recursion in assembly language using subroutines of your choice.

Answer:

Code:




```
1  [org 0x0100]
2
3  jmp start
4
5  data:  dw 5
6  count: dw 10
7  result: dw 0
8
9  add_function:  push bp
10               mov bp, sp
11               mov ax, [bp + 4] ; Load the argument from the stack
12               add [result], ax
13               dec word[count]
14               cmp word[count], 0
15               jle end_recursion ; Jump if less than or equal to zero
16
17               ; Recursive call
18               push word[result] ; Save the current result on the stack
19               push word[count] ; Save the current count on the stack
20               call add_function ; Recursive call
21               add sp, 4 ; Clean up the stack
22
23  end_recursion:
24               pop bp
25               ret
26
27  start:  mov ax, [data]
28         push ax
29         mov cx, [count]
30         push cx
31         call add_function
32
33         mov ax, 0x4c00
34         int 0x21
35
```

Explanation:

The provided assembly code is a concise implementation of recursive addition. It defines a function called `add function` that takes an argument from the stack, adds it to the result variable, decrements the count variable, and recursively calls itself until count becomes less than or equal to zero. The recursion is managed by saving and restoring the state (result and count) on the stack. The start section initializes data and count, pushes them onto the stack, and calls the add function. Finally, the program prints the result and exits gracefully. This code exemplifies a modular and clear approach to recursive addition in assembly language.

Screen-Shot of Debugger:


DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Progra...

AX 0000	SI 0000	CS 19F5	IP 0130	Stack +0 0000	Flags 7200
BX 0000	DI 0000	DS 19F5		+2 20CD	
CX 0000	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 >


00FE 0000	ADD	[BX+SI],AL
0100 E92D00	JMP	0130
0103 050000	ADD	AX,0000
0106 0037	ADD	[BX],DH
0108 005589	ADD	[DI-77],DL
010B E58B	IN	AX,[8B]
010D 46	INC	SI
010E 0401	ADD	AL,01
0110 06	PUSH	ES

1	0	1	2	3	4	5	6	7
DS:0000	CD	20	FF	9F	00	EA	FF	FF
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	E6	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	FF	FF	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	E6	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.ñ i |..+...
.....ff.
.....δ.μ.
6.....J.
.....

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


DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Progra...

AX 0000	SI 0000	CS 19F5	IP 0130	Stack +0 0000	Flags 7200
BX 0000	DI 0000	DS 19F5		+2 20CD	
CX 0000	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 >

0100 E92D00	JMP	0130
0130 A10301	MOV	AX,[0103]
0133 50	PUSH	AX
0134 8B0E0501	MOV	CX,[0105]
0138 51	PUSH	CX
0139 E8CDFF	CALL	0109
013C B8004C	MOV	AX,4C00
013F CD21	INT	21
0141 BE5EFC	MOV	DS,[BP-04]

1	0	1	2	3	4	5	6	7
DS:0000	CD	20	FF	9F	00	EA	FF	FF
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	E6	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	FF	FF	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	E6	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.ñ i |..+...
.....ff.
.....δ.μ.
6.....J.
.....

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

DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Progra...

AX 0005

SI 0000

CS 19F5

IP 0133

Stack +0 0000

Flags 7200

BX 0000

DI 0000

DS 19F5

+2 20CD

CX 0000

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 >

0130 A10301

MOV AX,[0103]

0133 50

PUSH AX

0134 8B0E0501

MOV CX,[0105]

0138 51

PUSH CX

0139 EBCDFF

CALL 0109

013C B8004C

MOV AX,4C00

013F CD21

INT 21

0141 8E5EFC

MOV DS,[BP-04]

0144 837D0E00

CMP [DI+0E],0000

1

0 1 2 3 4 5 6 7

DS:0000

CD 20 FF 9F 00 EA FF FF

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 E6 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 FF FF 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 E6 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.ñ i | . + ...

.....ff.

.....δ.μ.

6.....J.

.....

1 Step

2ProcStep

3Retrieve

4Help ON

5BRK Menu

6

7 up

8 dn

9 le

10 ri

DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Progra...

AX 0005

SI 0000

CS 19F5

IP 0134

Stack +0 0005

Flags 7200

BX 0000

DI 0000

DS 19F5

+2 0000

CX 0000

BP 0000

ES 19F5

HS 19F5

+4 20CD

OF DF IF SF ZF AF PF CF

DX 0000

SP FFFC

SS 19F5

FS 19F5

+6 9FFF

0 0 1 0 0 0 0 0

CMD >

0133 50

PUSH AX

0134 8B0E0501

MOV CX,[0105]

0138 51

PUSH CX

0139 EBCDFF

CALL 0109

013C B8004C

MOV AX,4C00

013F CD21

INT 21

0141 8E5EFC

MOV DS,[BP-04]

0144 837D0E00

CMP [DI+0E],0000

0148 7409

JZ 0153

1

0 1 2 3 4 5 6 7

DS:0000

CD 20 FF 9F 00 EA FF FF

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 E6 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 FF FF 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 E6 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.ñ i | . + ...

.....ff.

.....δ.μ.

6.....J.

.....

1 Step

2ProcStep

3Retrieve

4Help ON

5BRK Menu

6

7 up

8 dn

9 le

10 ri

DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Progra...

AX 0005	SI 0000	CS 19F5	IP 0138	Stack +0 0005	Flags 7200
BX 0000	DI 0000	DS 19F5		+2 0000	
CX 0000	BP 0000	ES 19F5	HS 19F5	+4 20CD	OF DF IF SF ZF AF PF CF
DX 0000	SP FFFC	SS 19F5	FS 19F5	+6 9FFF	0 0 1 0 0 0 0 0

CMD >

0134 8B0E0501 MOV CX,[0105]
0138 51 PUSH CX
0139 E8CDDFF CALL 0109
013C B8004C MOV AX,4C00
013F CD21 INT 21
0141 8E5EFC MOV DS,[BP-04]
0144 837D0E00 CMP [DI+0E],0000
0148 7409 JZ 0153
014A 8B46F2 MOV AX,[BP-0E]

1

	0	1	2	3	4	5	6	7
DS:0000	CD	20	FF	9F	00	EA	FF	FF
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	E6	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	FF	FF	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	E6	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.ñ ¡|..†...
.....ñ.
 δ.µ.
6.....J.
.....
.....

1 Step

2 ProcStep

3 Retrieve

4 Help ON

5 BRK Menu

6

7 up

8 dn

9 le

10 ri

DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Progra...

AX 0005	SI 0000	CS 19F5	IP 0139	Stack +0 0000	Flags 7200
BX 0000	DI 0000	DS 19F5		+2 0005	
CX 0000	BP 0000	ES 19F5	HS 19F5	+4 0000	OF DF IF SF ZF AF PF CF
DX 0000	SP FFFA	SS 19F5	FS 19F5	+6 20CD	0 0 1 0 0 0 0 0

CMD >

0138 51 PUSH CX
0139 E8CDDFF CALL 0109
013C B8004C MOV AX,4C00
013F CD21 INT 21
0141 8E5EFC MOV DS,[BP-04]
0144 837D0E00 CMP [DI+0E],0000
0148 7409 JZ 0153
014A 8B46F2 MOV AX,[BP-0E]
014D 48 DEC AX

1

	0	1	2	3	4	5	6	7
DS:0000	CD	20	FF	9F	00	EA	FF	FF
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	E6	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	FF	FF	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	E6	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.ñ ¡|..†...
.....ñ.
 δ.µ.
6.....J.
.....
.....

1 Step

2 ProcStep

3 Retrieve

4 Help ON

5 BRK Menu

6

7 up

8 dn

9 le

10 ri

DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Progra...

AX 0000	SI 0000	CS 19F5	IP 013C	Stack +0 0000	Flags 7284
BX 0000	DI 0000	DS 19F5		+2 0005	
CX 0000	BP 0000	ES 19F5	HS 19F5	+4 0000	OF DF IF SF ZF AF PF CF
DX 0000	SP FFFA	SS 19F5	FS 19F5	+6 20CD	0 0 1 1 0 0 1 0

CMD >	1	0	1	2	3	4	5	6	7		
0139 E8CDDF	CALL	0109	DS:0000	CD	20	FF	9F	00	EA	FF	FF
013C B8004C	MOV	AX,4C00	DS:0008	AD	DE	1B	05	C5	06	00	00
013F CD21	INT	21	DS:0010	18	01	10	01	18	01	92	01
0141 8E5EFC	MOV	DS,[BP-04]	DS:0018	01	01	01	00	02	FF	FF	FF
0144 837D0E00	CMP	[DI+0E1],0000	DS:0020	FF	FF	FF	FF	FF	FF	FF	FF
0148 7409	JZ	0153	DS:0028	FF	FF	FF	FF	EB	19	E6	11
014A 8B46F2	MOV	AX,[BP-0E]	DS:0030	A2	01	14	00	18	00	F5	19
014D 48	DEC	AX	DS:0038	FF	FF	FF	FF	00	00	00	00
014E 3B46F6	CMP	AX,[BP-0A]	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	FF	FF	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	E6	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.ñ ì |.+. ...
.....ff.
.....δ.μ.
6.....J.
.....

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

DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Progra...

AX 4C00	SI 0000	CS 19F5	IP 013F	Stack +0 0000	Flags 7284
BX 0000	DI 0000	DS 19F5		+2 0005	
CX 0000	BP 0000	ES 19F5	HS 19F5	+4 0000	OF DF IF SF ZF AF PF CF
DX 0000	SP FFFA	SS 19F5	FS 19F5	+6 20CD	0 0 1 1 0 0 1 0

CMD >	1	0	1	2	3	4	5	6	7		
013C B8004C	MOV	AX,4C00	DS:0000	CD	20	FF	9F	00	EA	FF	FF
013F CD21	INT	21	DS:0008	AD	DE	1B	05	C5	06	00	00
0141 8E5EFC	MOV	DS,[BP-04]	DS:0010	18	01	10	01	18	01	92	01
0144 837D0E00	CMP	[DI+0E1],0000	DS:0018	01	01	01	00	02	FF	FF	FF
0148 7409	JZ	0153	DS:0020	FF	FF	FF	FF	FF	FF	FF	FF
014A 8B46F2	MOV	AX,[BP-0E]	DS:0028	FF	FF	FF	FF	EB	19	E6	11
014D 48	DEC	AX	DS:0030	A2	01	14	00	18	00	F5	19
014E 3B46F6	CMP	AX,[BP-0A]	DS:0038	FF	FF	FF	FF	00	00	00	00
0151 7E08	JNG	015B	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	FF	FF	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	E6	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.ñ ì |.+. ...
.....ff.
.....δ.μ.
6.....J.
.....

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

