

Name: Amber Khurshid

Roll No: 22P-9295

Section: BAI-4A

COAL LAB TASK #06

Conditional Jump:

Code:

; a program to add three numbers using byte variables

[org 0x0100]

; initialize stuff

mov ax, 0 ; reset the accumulator

mov bx, 0 ; set the counter

outerloop:

add ax, [num1 + bx]

add bx, 2

cmp bx, 20 ; sets ZF=0 when they are equal

jne outerloop

mov [result], ax

```
mov ax, 0x4c00
```

```
int 0x21
```

; Intel Software Developer Manual - EFLAGS and Instructions (Page 435)

```
num1: dw 10, 20, 30, 40, 50, 10, 20, 30, 40, 50
```

```
result: dw 0
```

1.

```
mov ax, 0
```

Here we are initializing the ax register to value zero.

DOSBox 0.74-3, Cpu speed: 3000 cycles, Fram...

AX 0000	SI 0000	CS 19F5	IP 0103	Stack +0 0000	Flags 7200
BX 0000	DI 0000	DS 19F5		+2 20CD	
CX 0032	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 B80000	MOV	AX,0000	DS:0000	CD 20 FF 9F 00 EA F0 FE
0103 BB0000	MOV	BX,0000	DS:0008	AD DE 1B 05 C5 06 00 00
0106 03871C01	ADD	AX,[011C+BX]	DS:0010	18 01 10 01 18 01 92 01
010A 81C30200	ADD	BX,0002	DS:0018	01 01 01 00 02 FF FF FF
010E 81FB1400	CMP	BX,0014	DS:0020	FF FF FF FF FF FF FF FF
0112 75F2	JNZ	0106	DS:0028	FF FF FF FF EB 19 C0 11
0114 A33001	MOV	[0130],AX	DS:0030	A2 01 14 00 18 00 F5 19
0117 B8004C	MOV	AX,4C00	DS:0038	FF FF FF FF 00 00 00 00
011A CD21	INT	21	DS:0040	05 00 00 00 00 00 00 00
			DS:0048	00 00 00 00 00 00 00 00

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 FFff.
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

2.

`mov bx, 0`

Here we are initializing the bx register to value zero.

DOSBox 0.74-3, Cpu speed: 3000 cycles, Fram...

AX 0000	SI 0000	CS 19F5	IP 0106	Stack +0 0000	Flags 7200
BX 0000	DI 0000	DS 19F5		+2 20CD	
CX 0032	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 >

	0	1	2	3	4	5	6	7
0103 BB0000	MOV	BX,0000						
0106 03871C01	ADD	AX,[011C+BX]						
010A 81C30200	ADD	BX,0002						
010E 81FB1400	CMP	BX,0014						
0112 75F2	JNZ	0106						
0114 A33001	MOV	[0130],AX						
0117 B8004C	MOV	AX,4C00						
011A CD21	INT	21						
011C 0A00	OR	AL,[BX+SI]						

	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	00	02	FF	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

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

3.

```
add ax, [num1 + bx]
```

Here we will add the value present at the address **ax + bx** to **ax** register.

DOSBox 0.74-3, Cpu speed: 3000 cycles, Fram...

AX 000A	SI 0000	CS 19F5	IP 010A	Stack +0 0000	Flags 7204
BX 0000	DI 0000	DS 19F5		+2 20CD	
CX 0032	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 >

0106 03871C01	ADD	AX, [011C+BX]	1	0	1	2	3	4	5	6	7
010A 81C30200	ADD	BX, 0002	DS:0000	CD	20	FF	9F	00	EA	F0	FE
010E 81FB1400	CMP	BX, 0014	DS:0008	AD	DE	1B	05	C5	06	00	00
0112 75F2	JNZ	0106	DS:0010	18	01	10	01	18	01	92	01
0114 A33001	MOV	[0130], AX	DS:0018	01	01	01	00	02	FF	FF	FF
0117 B8004C	MOV	AX, 4C00	DS:0020	FF	FF	FF	FF	FF	FF	FF	FF
011A CD21	INT	21	DS:0028	FF	FF	FF	FF	EB	19	C0	11
011C 0A00	OR	AL, [BX+SI]	DS:0030	A2	01	14	00	18	00	F5	19
011E 1400	ADC	AL, 00	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

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

4.

add bx, 2

The instruction **add bx, 2** increments the value stored in the BX register by 2. This operation is part of a loop where the updated value in BX will be compared with 20 to determine whether the loop should continue running or if a jump is required.

DOSBox 0.74-3, Cpu speed: 3000 cycles, Fram...

AX 000A	SI 0000	CS 19F5	IP 010E	Stack +0 0000	Flags 7200
BX 0002	DI 0000	DS 19F5		+2 20CD	
CX 0032	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 >

010A 81C30200	ADD	BX,0002	1	0	1	2	3	4	5	6	7
010E 81FB1400	CMP	BX,0014	DS:0000	CD	20	FF	9F	00	EA	F0	FE
0112 75F2	JNZ	0106	DS:0008	AD	DE	1B	05	C5	06	00	00
0114 A33001	MOV	[0130],AX	DS:0010	18	01	10	01	18	01	92	01
0117 B8004C	MOV	AX,4C00	DS:0018	01	01	01	00	02	FF	FF	FF
011A CD21	INT	21	DS:0020	FF	FF	FF	FF	FF	FF	FF	FF
011C 0A00	OR	AL,[BX+SI]	DS:0028	FF	FF	FF	FF	EB	19	C0	11
011E 1400	ADC	AL,00	DS:0030	A2	01	14	00	18	00	F5	19
0120 1E	PUSH	DS	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.
.....

1 Step

2ProcStep

3Retrieve

4Help ON

5BRK Menu

6

7 up

8 dn

9 le

10 ri

5.

cmp bx, 20

The **cmp bx, 20** instruction checks whether the value in the BX register is equal to 20. If they are equal, the Zero flag is set to 0.

DOSBox 0.74-3, Cpu speed: 3000 cycles, Fram...

AX 000A	SI 0000	CS 19F5	IP 0112	Stack +0 0000	Flags 7295
BX 0002	DI 0000	DS 19F5		+2 20CD	
CX 0032	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 1 1 1

CMD >

010E 81FB1400	CMP	BX,0014
0112 75F2	JNZ	0105
0114 A33001	MOV	[0130],AX
0117 B8004C	MOV	AX,4C00
011A CD21	INT	21
011C 0A00	OR	AL,[BX+SI]
011E 1400	ADC	AL,00
0120 1E	PUSH	DS
0121 0028	ADD	[BX+SI],CH

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

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 02 FF FF FFff.
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

6.

jne outerloop

The instruction **jne outerloop** examines whether the Zero flag is set to 0. If it's not set to zero, it means that the values being compared are not equal, so the program jumps to the address labeled **outerloop** and continues looping.

DOSBox 0.74-3, Cpu speed: 3000 cycles, Fram...

AX 000A	SI 0000	CS 19F5	IP 0106	Stack +0 0000	Flags 7295
BX 0002	DI 0000	DS 19F5		+2 20CD	
CX 0032	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 1 1 1

CMD >

0112 75F2	JNZ	0106	
0106 03871C01	ADD	AX, [011C+BX]	
010A 81C30200	ADD	BX, 0002	
010E 81FB1400	CMP	BX, 0014	
0112 75F2	JNZ	0106	
0114 A33001	MOV	[0130], AX	
0117 B8004C	MOV	AX, 4C00	
011A CD21	INT	21	
011C 0A00	OR	AL, [BX+SI]	

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

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

7.

Second Iteration:

This is second iteration of the loop. Here the value stored previously in ax register is added to [num1+bx] and we add 2 to the bx register. The value of bx is compared to 20. A jump to the outer loop occurs if the value in bx is not equal to 20.

The Zero Flag is set to zero, indicating BX is not equal to 20.

DOSBox 0.74-3, Cpu speed: 3000 cycles, Fram...

AX 001E	SI 0000	CS 19F5	IP 0106	Stack +0 0000	Flags 7285
BX 0004	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 1 0 0 1 1

CMD >

0112 75F2	JNZ	0106	
0106 03871C01	ADD	AX, [011C+BX]	
010A 81C30200	ADD	BX, 0002	
010E 81FB1400	CMP	BX, 0014	
0112 75F2	JNZ	0106	
0114 A33001	MOV	[0130], AX	
0117 B8004C	MOV	AX, 4C00	
011A CD21	INT	21	
011C 0A00	OR	AL, [BX+SI]	

	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

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

8.

Third Iteration:

Here the value stored previously in ax register is added to [num1+bx] and we add 2 to the bx register. The value of bx is compared to 20. A jump to the outer loop occurs if the value in bx is not equal to 20.

DOSBox 0.74-3, Cpu speed: 3000 cycles, Fram...

AX 003C	SI 0000	CS 19F5	IP 010E	Stack +0 0000	Flags 7204
BX 0006	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 1 0

CMD >

010A 81C30200	ADD	BX,0002	DS:0000	CD 20 FF 9F 00 EA FF FF
010E 81FB1400	CMP	BX,0014	DS:0008	AD DE 1B 05 C5 06 00 00
0112 75F2	JNZ	0106	DS:0010	18 01 10 01 18 01 92 01
0114 A33001	MOV	[0130],AX	DS:0018	01 01 01 00 02 FF FF FF
0117 B8004C	MOV	AX,4C00	DS:0020	FF FF FF FF FF FF FF FF
011A CD21	INT	21	DS:0028	FF FF FF FF EB 19 E6 11
011C 0A00	OR	AL,[BX+SI]	DS:0030	A2 01 14 00 18 00 F5 19
011E 1400	ADC	AL,00	DS:0038	FF FF FF FF 00 00 00 00
0120 1E	PUSH	DS	DS:0040	05 00 00 00 00 00 00 00
			DS:0048	00 00 00 00 00 00 00 00

DS:0000	CD 20 FF 9F 00 EA FF FF	AD DE 1B 05 C5 06 00 00	= f.n i ..+...
DS:0010	18 01 10 01 18 01 92 01	01 01 01 00 02 FF FF FFfl.
DS:0020	FF FF FF FF FF FF FF FF	FF FF FF FF EB 19 E6 11	δ.μ.
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

9.

Fourth Iteration:

AX is set to 60.

BX is set to 6.

The value of num1 (6) is added to 40, resulting in 100, and added to AX.

BX is incremented to 8.

Comparison is made between BX (8) and 20.

The Zero Flag is set to zero, indicating BX is not equal to 20.

DOSBox 0.74-3, Cpu speed: 3000 cycles, Fram...

AX 0064	SI 0000	CS 19F5	IP 010E	Stack +0 0000	Flags 7200
BX 0008	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 >

010A 81C30200	ADD	BX,0002	DS:0000	CD 20 FF 9F 00 EA FF FF
010E 81FB1400	CMP	BX,0014	DS:0008	AD DE 1B 05 C5 06 00 00
0112 75F2	JNZ	0106	DS:0010	18 01 10 01 18 01 92 01
0114 A33001	MOV	[0130],AX	DS:0018	01 01 01 00 02 FF FF FF
0117 B8004C	MOV	AX,4C00	DS:0020	FF FF FF FF FF FF FF FF
011A CD21	INT	21	DS:0028	FF FF FF FF EB 19 E6 11
011C 0A00	OR	AL,[BX+SI]	DS:0030	A2 01 14 00 18 00 F5 19
011E 1400	ADC	AL,00	DS:0038	FF FF FF FF 00 00 00 00
0120 1E	PUSH	DS	DS:0040	05 00 00 00 00 00 00 00
			DS:0048	00 00 00 00 00 00 00 00

DS:0000	CD 20 FF 9F 00 EA FF FF	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 FFff.
DS:0020	FF FF FF FF FF FF FF FF	FF FF FF FF EB 19 E6 11	δ.ρ.
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

10.

Fifth Iteration:

AX is set to 100.

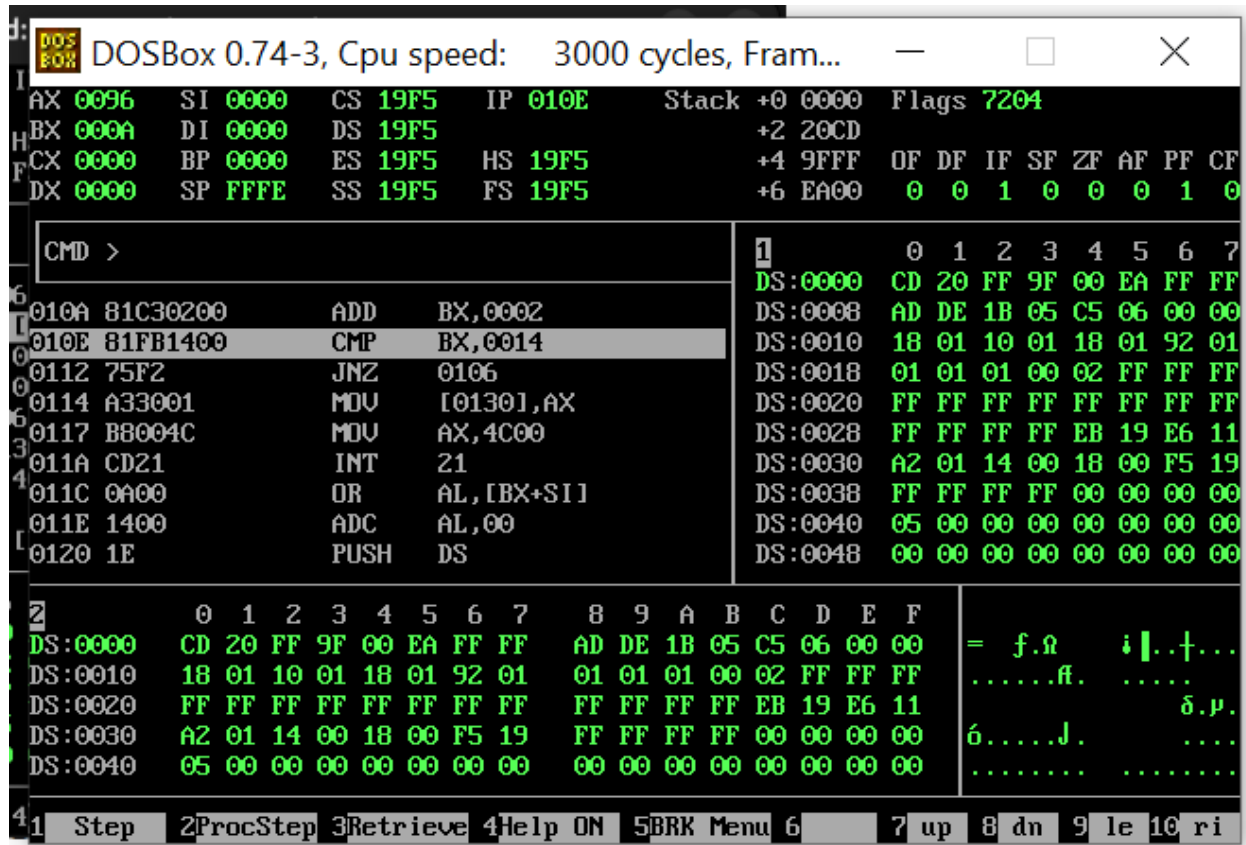
BX is set to 8.

The value of num1 (8) is added to 50, resulting in 150, and added to AX.

BX is incremented to 10.

Comparison is made between BX (10) and 20.

The Zero Flag is set to zero, indicating BX is not equal to 20.



11.

Sixth Iteration:

AX is set to 150.

BX is set to 10.

The value of num1 (10) is added to 10, resulting in 160, and added to AX.

BX is incremented to 12.

Comparison is made between BX (12) and 20.

The Zero Flag is set to zero, indicating BX is not equal to 20.

DOSBox 0.74-3, Cpu speed: 3000 cycles, Fram...

AX 00A0	SI 0000	CS 19F5	IP 010E	Stack +0 0000	Flags 7204
BX 000C	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 1 0

CMD >

010A 81C30200	ADD	BX,0002	DS:0000	CD 20 FF 9F 00 EA FF FF
010E 81FB1400	CMP	BX,0014	DS:0008	AD DE 1B 05 C5 06 00 00
0112 75F2	JNZ	0106	DS:0010	18 01 10 01 18 01 92 01
0114 A33001	MOV	[0130],AX	DS:0018	01 01 01 00 02 FF FF FF
0117 B8004C	MOV	AX,4C00	DS:0020	FF FF FF FF FF FF FF FF
011A CD21	INT	21	DS:0028	FF FF FF FF EB 19 E6 11
011C 0A00	OR	AL,[BX+SI]	DS:0030	A2 01 14 00 18 00 F5 19
011E 1400	ADC	AL,00	DS:0038	FF FF FF FF 00 00 00 00
0120 1E	PUSH	DS	DS:0040	05 00 00 00 00 00 00 00
			DS:0048	00 00 00 00 00 00 00 00

DS:0000	CD 20 FF 9F 00 EA FF FF	AD DE 1B 05 C5 06 00 00	= f.0 i ..+...
DS:0010	18 01 10 01 18 01 92 01	01 01 01 00 02 FF FF FFff.
DS:0020	FF FF FF FF FF FF FF FF	FF FF FF FF EB 19 E6 11δ.p.
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

12.

Seventh Iteration:

AX is set to 160.

BX is set to 12.

The value of num1 (12) is added to 20, resulting in 180, and added to AX.

BX is incremented to 14.

Comparison is made between BX (14) and 20.

The Zero Flag is set to zero, indicating BX is not equal to 20.

DOSBox 0.74-3, Cpu speed: 3000 cycles, Fram...

AX 00B4	SI 0000	CS 19F5	IP 010E	Stack +0 0000	Flags 7200
BX 000E	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

CMD >

010A 81C30200	ADD	BX,0002	DS:0000	CD 20 FF 9F 00 EA FF FF
010E 81FB1400	CMP	BX,0014	DS:0008	AD DE 1B 05 C5 06 00 00
0112 75F2	JNZ	0106	DS:0010	18 01 10 01 18 01 92 01
0114 A33001	MOV	[0130],AX	DS:0018	01 01 01 00 02 FF FF FF
0117 B8004C	MOV	AX,4C00	DS:0020	FF FF FF FF FF FF FF FF
011A CD21	INT	21	DS:0028	FF FF FF FF EB 19 E6 11
011C 0A00	OR	AL,[BX+SI]	DS:0030	A2 01 14 00 18 00 F5 19
011E 1400	ADC	AL,00	DS:0038	FF FF FF FF 00 00 00 00
0120 1E	PUSH	DS	DS:0040	05 00 00 00 00 00 00 00
			DS:0048	00 00 00 00 00 00 00 00

DS:0000	CD 20 FF 9F 00 EA FF FF	AD DE 1B 05 C5 06 00 00	= f.ñ ; ..+...
DS:0010	18 01 10 01 18 01 92 01	01 01 01 00 02 FF FF FFff.
DS:0020	FF FF FF FF FF FF FF FF	FF FF FF FF EB 19 E6 11	δ.p.
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

13.

Eighth Iteration:

AX is set to 180.

BX is set to 14.

The value of num1 (14) is added to 30, resulting in 210, and added to AX.

BX is incremented to 16.

Comparison is made between BX (16) and 20.

The Zero Flag is set to zero, indicating BX is not equal to 20.

DOSBox 0.74-3, Cpu speed: 3000 cycles, Fram...

AX 00D2	SI 0000	CS 19F5	IP 010E	Stack +0 0000	Flags 7210
BX 0010	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 1 0 0

CMD >			0	1	2	3	4	5	6	7		
010A	81C30200	ADD	BX,0002	DS:0000	CD	20	FF	9F	00	EA	FF	FF
010E	81FB1400	CMP	BX,0014	DS:0008	AD	DE	1B	05	C5	06	00	00
0112	75F2	JNZ	0106	DS:0010	18	01	10	01	18	01	92	01
0114	A33001	MOV	[0130],AX	DS:0018	01	01	01	00	02	FF	FF	FF
0117	B8004C	MOV	AX,4C00	DS:0020	FF	FF	FF	FF	FF	FF	FF	FF
011A	CD21	INT	21	DS:0028	FF	FF	FF	FF	EB	19	E6	11
011C	0A00	OR	AL,[BX+SI]	DS:0030	A2	01	14	00	18	00	F5	19
011E	1400	ADC	AL,00	DS:0038	FF	FF	FF	FF	00	00	00	00
0120	1E	PUSH	DS	DS:0040	05	00	00	00	00	00	00	00
				DS:0048	00	00	00	00	00	00	00	00

	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	= f.Ω i ..†...
DS:0010	18	01	10	01	18	01	92	01	01	01	01	00	02	FF	FF	FFff.
DS:0020	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	EB	19	E6	11	δ.μ.
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

19 FF FF FF FF 00 00 00 00 16 J

14.

Ninth Iteration:

AX is set to 210.

BX is set to 16.

The value of num1 (16) is added to 40, resulting in 250, and added to AX.

BX is incremented to 18.

Comparison is made between BX (18) and 20.

The Zero Flag is set to zero, indicating BX is not equal to 20.

The screenshot shows the DOSBox 0.74-3 interface. At the top, the title bar reads "DOSBox 0.74-3, Cpu speed: 3000 cycles, Fram...". Below the title bar, the register and status information is displayed:

Register	Value	Register	Value	Register	Value	Register	Value	Stack	Value	Flags	Value
AX	00FA	SI	0000	CS	19F5	IP	010E	Stack	+0	0000	7204
BX	0012	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 1 0

Below the register information, the command prompt shows "CMD >". The assembly code is displayed in the left pane, and the memory dump is in the right pane.

Assembly Code:

```
010A 81C30200 ADD BX,0002
010E 81FB1400 CMP BX,0014
0112 75F2 JNZ 0106
0114 A33001 MOV [0130],AX
0117 B8004C MOV AX,4C00
011A CD21 INT 21
011C 0A00 OR AL,[BX+SI]
011E 1400 ADC AL,00
0120 1E PUSH DS
```

Memory Dump (DS:0000 to DS:0048):

Address	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	00	02	FF	FF	FF	FF
DS:0020	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

At the bottom, the status bar shows "1 Step 2ProcStep 3Retrieve 4Help ON 5BRK Menu 6 7 up 8 dn 9 le 10 ri".

15.

Tenth Iteration:

AX is set to 250.

BX is set to 18.

The value of num1 (18) is added to 50, resulting in 300, and added to AX.

BX is incremented to 20.

Comparison is made between BX (20) and 20.

The Zero Flag is set to 1, indicating BX is equal to 20, which triggers an exit from the loop.

The screenshot shows the DOSBox 0.74-3 interface. At the top, it displays 'DOSBox 0.74-3, Cpu speed: 3000 cycles, Fram...'. Below this, a status bar shows registers: AX 012C, SI 0000, CS 19F5, IP 010E, Stack +0 0000, Flags 7204; BX 0014, 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 1 0.

The command prompt shows 'CMD >'. Below it, a list of assembly instructions is displayed:

```
010A 81C30200 ADD BX,0002
010E 81FB1400 CMP BX,0014
0112 75F2 JNZ 0106
0114 A33001 MOV [0130],AX
0117 B8004C MOV AX,4C00
011A CD21 INT 21
011C 0A00 OR AL,[BX+SI]
011E 1400 ADC AL,00
0120 1E PUSH DS
```

To the right of the instructions, a memory dump shows the contents of memory locations from DS:0000 to DS:0048. The dump is organized into columns for bytes 0 through 7. The memory dump shows the following values:

Address	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

At the bottom of the screenshot, a status bar shows the following information: 1 Step, 2 ProcStep, 3 Retrieve, 4 Help ON, 5 BRK Menu, 6, 7 up, 8 dn, 9 le, 10 ri.

16.

mov [result], ax

This command will move the value from AX register to the memory address which is labelled as result.

DOSBox 0.74-3, Cpu speed: 3000 cycles, Fram...

AX 012C	SI 0000	CS 19F5	IP 0117	Stack +0 0000	Flags 7244
BX 0014	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 1 0 1 0

CMD >

0114 A33001	MOV	[0130],AX	1	0	1	2	3	4	5	6	7
0117 B8004C	MOV	AX,4C00	DS:0000	CD	20	FF	9F	00	EA	FF	FF
011A CD21	INT	21	DS:0008	AD	DE	1B	05	C5	06	00	00
011C 0A00	OR	AL,[BX+SI]	DS:0010	18	01	10	01	18	01	92	01
011E 1400	ADC	AL,00	DS:0018	01	01	01	00	02	FF	FF	FF
0120 1E	PUSH	DS	DS:0020	FF	FF	FF	FF	FF	FF	FF	FF
0121 0028	ADD	[BX+SI],CH	DS:0028	FF	FF	FF	FF	EB	19	E6	11
0123 0032	ADD	[BP+SI],DH	DS:0030	A2	01	14	00	18	00	F5	19
0125 000A	ADD	[BP+SI],CL	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	= f.ñ i ..†...
DS:0010	18	01	10	01	18	01	92	01	01	01	00	02	FF	FF	FF	FFff.
DS:0020	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	EB	19	E6	11		δ.μ.
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
2 ProcStep
3 Retrieve
4 Help ON
5 BRK Menu
6
7 up
8 dn
9 le
10 ri

17.

mov ax, 0x4c00

mov ax, 0x4c00 , this moves the value 4c00 into ax register which is a program terminator.


DOSBox 0.74-3, Cpu speed: 3000 cycles, Fram...

AX	4C00	SI	0000	CS	19F5	IP	011A	Stack	+0 0000	Flags	7244
BX	0014	DI	0000	DS	19F5				+2 20CD		
CX	0000	BP	0000	ES	19F5	HS	19F5		+4 9FFF	OF	DF
DX	0000	SP	FFFE	SS	19F5	FS	19F5		+6 EA00	IF	SF
										ZF	AF
										PF	CF
										0	1
										0	1

CMD >

0117	B804C	MOV	AX,4C00
011A	CD21	INT	21
011C	0A00	OR	AL,[BX+SI]
011E	1400	ADC	AL,00
0120	1E	PUSH	DS
0121	0028	ADD	[BX+SI],CH
0123	0032	ADD	[BP+SI],DH
0125	000A	ADD	[BP+SI],CL
0127	0014	ADD	[SI],DL

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

1 Step

2 ProcStep

3 Retrieve

4 Help ON

5 BRK Menu

6

7 up

8 dn

9 le

10 ri

18.

int 0x21

The **INT 21** instruction initiates an interrupt call.

DOSBox 0.74-3, Cpu speed: 3000 cycles, Fram...

AX 4C00 SI 0000 CS F000 IP 14A0 Stack +0 42BD Flags 7044
 BX 0014 DI 0000 DS 19F5 +2 06C5
 CX 0000 BP 0000 ES 19F5 HS 19F5 +4 7044 OF DF IF SF ZF AF PF CF
 DX 0000 SP FFF2 SS 19F5 FS 19F5 +6 011C 0 0 0 0 1 0 1 0

CMD >

Address	Instruction	Comment
011A	CD 21	INT 21
14A0	FB	STI
14A1	FE	DB FE
14A2	38 25	CMP [DI], AH
14A4	00 CF	ADD BH, CL
14A6	CB	RET Far
14A7	51	PUSH CX
14A8	B9 40 01	MOV CX, 0140
14AB	E2 FE	LOOP 14AB

Memory Dump (DS:0000):

Address	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	00	02	FF	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

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

Unconditional Jump:

[org 0x0100]

jmp start ; see next instructions when you haven't yet executed this!

num1: dw 10, 20, 30, 40, 50, 10, 20, 30, 40, 50

result: dw 0

start:

; initialize stuff

mov ax, 0 ; reset the accumulator

mov bx, 0 ; set the counter

outerloop:

add ax, [num1 + bx]

add bx, 2

cmp bx, 20 ; sets ZF=0 when they are equal

jne outerloop

mov [result], ax

mov ax, 0x4c00

int 0x21

1.

jmp start refers to jumping to a specific labeled point within the program, effectively initiating or restarting program execution from that point onwards. The **jmp start** command directs the program to execute from the **start** label within the code.

DOSBox 0.74-3, Cpu speed: 3000 cycles, Fram...

AX 0000	SI 0000	CS 19F5	IP 0119	Stack +0 0000	Flags 7200
BX 0000	DI 0000	DS 19F5		+2 20CD	
CX 0035	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 E91600	JMP	0119
0119 B80000	MOV	AX,0000
011C BB0000	MOV	BX,0000
011F 03870301	ADD	AX,[0103+BX]
0123 81C30200	ADD	BX,0002
0127 81FB1400	CMP	BX,0014
012B 75F2	JNZ	011F
012D A31701	MOV	[0117],AX
0130 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	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

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

2.

`mov ax, 0`

Through this instruction the `ax` register is initialized to zero.

DOSBox 0.74-3, Cpu speed: 3000 cycles, Fram...

AX 0000

SI 0000

CS 19F5

IP 011C

Stack +0 0000

Flags 7200

BX 0000

DI 0000

DS 19F5

+2 20CD

CX 0035

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 >

0119 B80000

MOV

AX,0000

011C BB0000

MOV

BX,0000

011F 03870301

ADD

AX,[0103+BX]

0123 81C30200

ADD

BX,0002

0127 81FB1400

CMP

BX,0014

012B 75F2

JNZ

011F

012D A31701

MOV

[0117],AX

0130 B8004C

MOV

AX,4C00

0133 CD21

INT

21

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≡■ ÷|..†...

.....ff.

δ.L.

ó.....J.

.....

1 Step

2ProcStep

3Retrieve

4Help ON

5BRK Menu

6

7 up

8 dn

9 le

10 ri

3.

mov bx, 0

Through this instruction the **bx** register is initialized to zero.

DOSBox 0.74-3, Cpu speed: 3000 cycles, Fram...

AX 0000	SI 0000	CS 19F5	IP 011F	Stack +0 0000	Flags 7200
BX 0000	DI 0000	DS 19F5		+2 20CD	
CX 0035	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 >

011C BB0000	MOV	BX,0000
011F 03870301	ADD	AX,[0103+BX]
0123 81C30200	ADD	BX,0002
0127 81FB1400	CMP	BX,0014
012B 75F2	JNZ	011F
012D A31701	MOV	[0117],AX
0130 B8004C	MOV	AX,4C00
0133 CD21	INT	21
0135 D2	DB	D2

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

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

4.

add ax, [num1 + bx]

The value at the address num1+bx is added to the ax register through this command.

DOSBox 0.74-3, Cpu speed: 3000 cycles, Fram...

AX 000A	SI 0000	CS 19F5	IP 0123	Stack +0 0000	Flags 7204
BX 0000	DI 0000	DS 19F5		+2 20CD	
CX 0035	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 >

011F 03870301	ADD	AX, [0103+BX]	1	0	1	2	3	4	5	6	7
0123 81C30200	ADD	BX, 0002	DS:0000	CD	20	FF	9F	00	EA	F0	FE
0127 81FB1400	CMP	BX, 0014	DS:0008	AD	DE	1B	05	C5	06	00	00
012B 75F2	JNZ	011F	DS:0010	18	01	10	01	18	01	92	01
012D A31701	MOV	[0117], AX	DS:0018	01	01	01	00	FF	00	01	FF
0130 B8004C	MOV	AX, 4C00	DS:0020	FF	FF	FF	FF	FF	FF	FF	FF
0133 CD21	INT	21	DS:0028	FF	FF	FF	FF	EB	19	C0	11
0135 D2	DB	D2	DS:0030	A2	01	14	00	18	00	F5	19
0136 7504	JNZ	013C	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 |..†...
.....ft.
δ. L.
ó.....J.
.....

1 Step

2 ProcStep

3 Retrieve

4 Help ON

5 BRK Menu

6

7 up

8 dn

9 le

10 ri

5.

add bx, 2

The instruction **add bx, 2** increments the value stored in the BX register by 2. This operation is part of a loop where the updated value in BX will be compared with 20 to determine whether the loop should continue running or if a jump is required.

DOSBox 0.74-3, Cpu speed: 3000 cycles, Fram...

AX 000A	SI 0000	CS 19F5	IP 0127	Stack +0 0000	Flags 7200
BX 0002	DI 0000	DS 19F5		+2 20CD	
CX 0035	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 >

0123 81C30200	ADD	BX,0002
0127 81FB1400	CMP	BX,0014
012B 75F2	JNZ	011F
012D A31701	MOV	[0117],AX
0130 B8004C	MOV	AX,4C00
0133 CD21	INT	21
0135 D2	DB	D2
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	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	FFff.
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
2 ProcStep
3 Retrieve
4 Help ON
5 BRK Menu
6
7 up
8 dn
9 le
10 ri

6.

cmp bx, 20

The **cmp bx, 20** instruction checks whether the value in the BX register is equal to 20. If they are equal, the Zero flag is set to 0.

DOSBox 0.74-3, Cpu speed: 3000 cycles, Fram...

AX 000A	SI 0000	CS 19F5	IP 012B	Stack +0 0000	Flags 7295
BX 0002	DI 0000	DS 19F5		+2 20CD	
CX 0035	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 1 1 1

CMD >

0127 81FB1400	CMP	BX,0014
012B 75F2	JNZ	011F
012D A31701	MOV	[0117],AX
0130 B8004C	MOV	AX,4C00
0133 CD21	INT	21
0135 D2	DB	D2
0136 7504	JNZ	013C
0138 85C0	TEST	AX,AX
013A 741C	JZ	0158

DS:0000	CD 20 FF 9F 00 EA F0 FE	AD DE 1B 05 C5 06 00 00
DS:0008	AD DE 1B 05 C5 06 00 00	01 01 01 00 FF 00 01 FF
DS:0010	18 01 10 01 18 01 92 01	FF FF FF FF FF FF FF FF
DS:0018	01 01 01 00 FF 00 01 FF	FF FF FF FF EB 19 C0 11
DS:0020	FF FF FF FF FF FF FF FF	FF FF FF FF 00 00 00 00
DS:0028	FF FF FF FF EB 19 C0 11	05 00 00 00 00 00 00 00
DS:0030	A2 01 14 00 18 00 F5 19	00 00 00 00 00 00 00 00
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

7.

jne outerloop

The instruction **jne outerloop** examines whether the Zero flag is set to 0. If it's not set to zero, it means that the values being compared are not equal, so the program jumps to the address labeled **outerloop** and continues looping.

DOSBox 0.74-3, Cpu speed: 3000 cycles, Fram...

AX 000A	SI 0000	CS 19F5	IP 011F	Stack +0 0000	Flags 7295
BX 0002	DI 0000	DS 19F5		+2 20CD	
CX 0035	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 1 1 1

CMD >

012B 75F2	JNZ	011F	
011F 03870301	ADD	AX, [0103+BX]	
0123 81C30200	ADD	BX, 0002	
0127 81FB1400	CMP	BX, 0014	
012B 75F2	JNZ	011F	
012D A31701	MOV	[0117], AX	
0130 B8004C	MOV	AX, 4C00	
0133 CD21	INT	21	
0135 D2	DB	D2	

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

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

8.

Iteration 2:

The value of **ax** is updated to **ax + [num1 + bx]**.

The value of **bx** is incremented by 2.

Comparison is made between **bx** and 20.

If **bx** is not equal to 20, a jump to the **outerloop** is executed.

DOSBox 0.74-3, Cpu speed: 3000 cycles, Fram...

AX 001E	SI 0000	CS 19F5	IP 011F	Stack +0 0000	Flags 7285
BX 0004	DI 0000	DS 19F5		+2 20CD	
CX 0035	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 >

012B 75F2	JNZ	011F
011F 03870301	ADD	AX, [0103+BX]
0123 81C30200	ADD	BX, 0002
0127 81FB1400	CMP	BX, 0014
012B 75F2	JNZ	011F
012D A31701	MOV	[0117], AX
0130 B8004C	MOV	AX, 4C00
0133 CD21	INT	21
0135 D2	DB	D2

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 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

9.

Iteration 3:

The value of **ax** is incremented by **[num1 + bx]**.

The value of **bx** is incremented by **2**.

Comparison is made between **bx** and **20**.

If **bx** is not equal to **20**, a jump to the **outerloop** is executed.

DOSBox 0.74-3, Cpu speed: 3000 cycles, Fram...

AX 003C	SI 0000	CS 19F5	IP 011F	Stack +0 0000	Flags 7281
BX 0006	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 1 0 0 0 1

CMD > █

012B 75F2	JNZ	011F
011F 03870301	ADD	AX, [0103+BX]
0123 81C30200	ADD	BX, 0002
0127 81FB1400	CMP	BX, 0014
012B 75F2	JNZ	011F
012D A31701	MOV	[0117], AX
0130 B8004C	MOV	AX, 4C00
0133 CD21	INT	21
0135 D2	DB	D2

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	FF	00	01	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
DS:0000	CD	20	FF	9F	00	EA	FF	FF
DS:0010	18	01	10	01	18	01	92	01
DS:0020	FF	FF	FF	FF	FF	FF	FF	FF
DS:0030	A2	01	14	00	18	00	F5	19
DS:0040	05	00	00	00	00	00	00	00

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

10.

Fourth Iteration:

AX is set to 60.

BX is set to 6.

The value of num1 (6) is added to 40, resulting in 100, and added to AX.

BX is incremented to 8.

Comparison is made between BX (8) and 20.

The Zero Flag is set to zero, indicating BX is not equal to 20.

DOSBox 0.74-3, Cpu speed: 3000 cycles, Fram...

AX 0064	SI 0000	CS 19F5	IP 011F	Stack +0 0000	Flags 7281
BX 0008	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 1 0 0 0 1

CMD > █

012B 75F2	JNZ	011F	
011F 03870301	ADD	AX, [0103+BX]	
0123 81C30200	ADD	BX, 0002	
0127 81FB1400	CMP	BX, 0014	
012B 75F2	JNZ	011F	
012D A31701	MOV	[0117], AX	
0130 B8004C	MOV	AX, 4C00	
0133 CD21	INT	21	
0135 D2	DB	D2	

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	FF	00	01	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	FF	00	01	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

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

11.

Fifth Iteration:

AX is set to 100.

BX is set to 8.

The value of num1 (8) is added to 50, resulting in 150, and added to AX.

BX is incremented to 10.

Comparison is made between BX (10) and 20.

The Zero Flag is set to zero, indicating BX is not equal to 20.

DOSBox 0.74-3, Cpu speed: 3000 cycles, Fram...

AX 0096	SI 0000	CS 19F5	IP 011F	Stack +0 0000	Flags 7285
BX 000A	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 1 0 0 1 1

CMD > 000A

012B 75F2	JNZ	011F
011F 03870301	ADD	AX, [0103+BX]
0123 81C30200	ADD	BX, 0002
0127 81FB1400	CMP	BX, 0014
012B 75F2	JNZ	011F
012D A31701	MOV	[0117], AX
0130 B8004C	MOV	AX, 4C00
0133 CD21	INT	21
0135 D2	DB	D2

DS:0000	CD 20 FF 9F 00 EA FF FF	AD DE 1B 05 C5 06 00 00	0 1 2 3 4 5 6 7
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 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	

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

12.

Sixth Iteration:

AX is set to 150.

BX is set to 10.

The value of num1 (10) is added to 10, resulting in 160, and added to AX.

BX is incremented to 12.

Comparison is made between BX (12) and 20.

The Zero Flag is set to zero, indicating BX is not equal to 20.

DOSBox 0.74-3, Cpu speed: 3000 cycles, Fram...

AX 00A0	SI 0000	CS 19F5	IP 011F	Stack +0 0000	Flags 7281
BX 000C	DI 0000	DS 19F5		+2 20CD	
CX 0000	BP 0000	ES 19F5	HS 19F5	+4 9FFF	OF DF IF SF ZF AF PF C
DX 0000	SP FFFE	SS 19F5	FS 19F5	+6 EA00	0 0 1 1 0 0 0

CMD > █

012B 75F2	JNZ	011F
011F 03870301	ADD	AX, [0103+BX]
0123 81C30200	ADD	BX, 0002
0127 81FB1400	CMP	BX, 0014
012B 75F2	JNZ	011F
012D A31701	MOV	[0117], AX
0130 B8004C	MOV	AX, 4C00
0133 CD21	INT	21
0135 D2	DB	D2

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	FF	00	01	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	FF	00	01	FF
DS:0020	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

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

13.

Seventh Iteration:

- AX is set to 160.
- BX is set to 12.
- The value of num1 (12) is added to 20, resulting in 180, and added to AX.
- BX is incremented to 14.
- Comparison is made between BX (14) and 20.
- The Zero Flag is set to zero, indicating BX is not equal to 20.

DOSBox 0.74-3, Cpu speed: 3000 cycles, Fram...

AX 00B4	SI 0000	CS 19F5	IP 011F	Stack +0 0000	Flags 7285
BX 000E	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 1 0 0 1 1

CMD >

012B 75F2	JNZ	011F
011F 03870301	ADD	AX, [0103+BX]
0123 81C30200	ADD	BX, 0002
0127 81FB1400	CMP	BX, 0014
012B 75F2	JNZ	011F
012D A31701	MOV	[0117], AX
0130 B8004C	MOV	AX, 4C00
0133 CD21	INT	21
0135 D2	DB	D2

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 FF 00 01 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	= f.Ω i . . † . . .
DS:0010	18	01	10	01	18	01	92	01	01	01	01	00	FF	00	01	FFft.
DS:0020	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	EB	19	E6	11	δ.μ.
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

14.

Eighth Iteration:

AX is set to 180.

BX is set to 14.

The value of num1 (14) is added to 30, resulting in 210, and added to AX.

BX is incremented to 16.

Comparison is made between BX (16) and 20.

The Zero Flag is set to zero, indicating BX is not equal to 20.

DOSBox 0.74-3, Cpu speed: 3000 cycles, Fram...

AX 00D2 SI 0000 CS 19F5 IP 011F Stack +0 0000 Flags 7295
 BX 0010 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 1 0 1 1 1

CMD > 0028

012B 75F2	JNZ	011F
011F 03870301	ADD	AX, [0103+BX]
0123 81C30200	ADD	BX, 0002
0127 81FB1400	CMP	BX, 0014
012B 75F2	JNZ	011F
012D A31701	MOV	[0117], AX
0130 B8004C	MOV	AX, 4C00
0133 CD21	INT	21
0135 D2	DB	D2

DS:0000	CD 20 FF 9F 00 EA FF FF	AD DE 1B 05 C5 06 00 00	0 1 2 3 4 5 6 7
DS:0008	AD DE 1B 05 C5 06 00 00	01 01 01 00 FF 00 01 FF	
DS:0010	18 01 10 01 18 01 92 01	FF FF FF FF FF FF FF FF	
DS:0018	01 01 01 00 FF 00 01 FF	FF FF FF FF EB 19 E6 11	
DS:0020	FF FF FF FF FF FF FF FF	FF FF FF FF 00 00 00 00	
DS:0028	FF FF FF FF EB 19 E6 11	00 00 00 00 00 00 00 00	
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

15.

Ninth Iteration:

AX is set to 210.

BX is set to 16.

The value of num1 (16) is added to 40, resulting in 250, and added to AX.

BX is incremented to 18.

Comparison is made between BX (18) and 20.

The Zero Flag is set to zero, indicating BX is not equal to 20.

DOSBox 0.74-3, Cpu speed: 3000 cycles, Fram...

AX 00FA	SI 0000	CS 19F5	IP 011F	Stack +0 0000	Flags 7291
BX 0012	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 1 0 1 0 1

CMD > 0032

012B 75F2	JNZ	011F
011F 03870301	ADD	AX, [0103+BX]
0123 81C30200	ADD	BX, 0002
0127 81FB1400	CMP	BX, 0014
012B 75F2	JNZ	011F
012D A31701	MOV	[0117], AX
0130 B8004C	MOV	AX, 4C00
0133 CD21	INT	21
0135 D2	DB	D2

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	FF	00	01	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	FF	00	01	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

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

16.

Tenth Iteration:

AX is set to 250.

BX is set to 18.

The value of num1 (18) is added to 50, resulting in 300, and added to AX.

BX is incremented to 20.

Comparison is made between BX (20) and 20.

The Zero Flag is set, indicating BX is equal to 20, which triggers an exit from the loop.

DOSBox 0.74-3, Cpu speed: 3000 cycles, Fram...

AX 00FA	SI 0000	CS 19F5	IP 011F	Stack +0 0000	Flags 7291
BX 0012	DI 0000	DS 19F5		+2 20CD	
CX 0035	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 1 0 1

CMD >

0032

012B 75F2	JNZ	011F
011F 03870301	ADD	AX, [0103+BX]
0123 81C30200	ADD	BX, 0002
0127 81FB1400	CMP	BX, 0014
012B 75F2	JNZ	011F
012D A31701	MOV	[0117], AX
0130 B8004C	MOV	AX, 4C00
0133 CD21	INT	21
0135 D2	DB	D2

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.
ó.....J.
.....

1

Step

2ProcStep

3Retrieve

4Help ON

5BRK Menu

6

7 up

8 dn

9 le

10 ri

17.

mov [result], ax

This instruction moves the value stored in AX register to the memory address labelled as result.

DOSBox 0.74-3, Cpu speed: 3000 cycles, Fram...

AX 012C

SI 0000

CS 19F5

IP 0130

Stack +0 0000

Flags 7244

BX 0014

DI 0000

DS 19F5

+2 20CD

CX 0035

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 >

012D A31701

MOV

[0117],AX

0130 B8004C

MOV

AX,4C00

0133 CD21

INT

21

0135 D2

DB

D2

0136 7504

JNZ

013C

0138 85C0

TEST

AX,AX

013A 741C

JZ

0158

013C C746DC0000

MOV

[BP-24],0000

0141 8E5EFC

MOV

DS,[BP-04]

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

= 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

18.

mov ax, 0x4c00

This moves the value 4c00 into ax register which is a program terminator.

DOSBox 0.74-3, Cpu speed: 3000 cycles, Fram...

AX 4C00	SI 0000	CS F000	IP 14A0	Stack +0 42BD	Flags 7044
BX 0014	DI 0000	DS 19F5		+2 06C5	
CX 0035	BP 0000	ES 19F5	HS 19F5	+4 7044	OF DF IF SF ZF AF PF CF
DX 0000	SP FFF2	SS 19F5	FS 19F5	+6 0135	0 0 0 0 1 0 1 0

CMD >

0133 CD21	INT 21	
14A0 FB	STI	
14A1 FE	DB FE	
14A2 3825	CMP [DI],AH	
14A4 00CF	ADD BH,CL	
14A6 CB	RET Far	
14A7 51	PUSH CX	
14A8 B94001	MOV CX,0140	
14AB E2FE	LOOP 14AB	

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 |..†...
.....ft.
δ. L.
ó.....J.
.....

1 Step

2ProcStep

3Retrieve

4Help ON

5BRK Menu

6

7 up

8 dn


9 le

10 ri

20.

int 0x21

The INT 21 instruction initiates an interrupt call.


DOSBox 0.74-3, Cpu speed: 3000 cycles, Fram...
—
☐
×

AX 4C00	SI 0000	CS 19F5	IP 0133	Stack +0 0000	Flags 7244
BX 0014	DI 0000	DS 19F5		+2 20CD	
CX 0035	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 >|

0130 B8004C	MOV	AX,4C00
0133 CD21	INT	21
0135 D2	DB	D2
0136 7504	JNZ	013C
0138 85C0	TEST	AX,AX
013A 741C	JZ	0158
013C C746DC0000	MOV	[BP-24],0000
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	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
DS:0000	CD	20	FF	9F	00	EA	F0	FE
DS:0010	18	01	10	01	18	01	92	01
DS:0020	FF	FF	FF	FF	FF	FF	FF	FF
DS:0030	A2	01	14	00	18	00	F5	19
DS:0040	05	00	00	00	00	00	00	00

= f.Ω=■ i|.+. .
ff.
 δ.L.
 6.....J.

1 Step

2ProcStep

3Retrieve

4Help ON

5BRK Menu

6

7 up

8 dn

9 le

10 ri