

1. CODE : Centigrade (Celsius) to Fahrenheit

```
org 100h
JMP start

tc db 10
result db ?

start:

mov cl, tc
mov al, 09
imul cl
mov cl, 05
idiv cl
add al, 32
mov result, al

HLT
ret
```

emulator: celcius.com_

file math debug view external virtual devices virtual drive help

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

registers	H	L
AX	00	00
BX	00	00
CX	00	17
DX	00	00
CS	0700	
IP	0100	
SS	0700	
SP	FFFE	
BP	0000	
SI	0000	
DI	0000	
DS	0700	
ES	0700	

0700:0100

07100:	EB	235	6
07101:	02	002	0
07102:	0A	010	NEWL
07103:	00	000	NULL
07104:	8A	138	è
07105:	0E	014	À
07106:	02	002	0
07107:	01	001	0
07108:	B0	176	À
07109:	09	009	TAB
0710A:	F6	246	÷
0710B:	E9	233	0
0710C:	B1	177	À
0710D:	05	005	5
0710E:	F6	246	÷
0710F:	F9	249	-
07110:	04	004	♦
07111:	20	032	SPA
07112:	A2	162	ó
07113:	03	003	♥
07114:	01	001	0

0700:0100

```
JMP 0104h
OR AL, [BX + SI]
MOV CL, [00102h]
MOV AL, 09h
IMUL CL
MOV CL, 05h
IDIV CL
ADD AL, 020h
MOV [00103h], AL
HLT
RET
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
...
```

original sou...

```
01 org 100h
02 JMP start
03
04
05 tc db 10
06 result db ?
07
08 start:
09
10 mov cl, tc
11 mov al, 09
12 imul cl
13 mov cl, 05
14 idiv cl
15 add al, 32
16 mov result, al
17
18
19 HLT
20 ret
21
22
23
```

screen source reset aux vars debug stack

emulator: celcius.com_

file math debug view external virtual devices virtual drive help

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

registers	H	L
AX	00	00
BX	00	00
CX	00	17
DX	00	00
CS	0700	
IP	0104	
SS	0700	
SP	FFFE	
BP	0000	
SI	0000	
DI	0000	
DS	0700	
ES	0700	

0700:0100	0700:0100
07100: EB 235 6	JMP 0104h
07101: 02 002 0	OR AL, [BX + SI]
07102: 0A 010 NEWL	MOV CL, [00102h]
07103: 00 000 NULL	MOV AL, 09h
07104: 8A 138 2	IMUL CL
07105: 0E 014 7	MOV CL, 05h
07106: 02 002 0	IDIV CL
07107: 01 001 0	ADD AL, 020h
07108: B0 176 7	MOV [00103h], AL
07109: 09 009 TAB	HLT
0710A: F6 246 0	RET
0710B: E9 233 0	NOP
0710C: B1 177 7	NOP
0710D: 05 005 2	NOP
0710E: F6 246 0	NOP
0710F: F9 249 0	NOP
07110: 04 004 2	NOP
07111: 20 032 SPA	NOP
07112: A2 162 6	NOP
07113: 03 003 7	NOP
07114: 01 001 0	...

original sou...

```

01
02 org 100h
03 JMP start
04
05 tc db 10
06 result db ?
07
08 start:
09
10 mov cl, tc
11 mov al, 09
12 imul cl
13 mov cl, 05
14 idiv cl
15 add al, 32
16 mov result, al
17
18
19 HLT
20 ret
21
22

```

screen source reset aux vars debug stack

emulator: celcius.com_

file math debug view external virtual devices virtual drive help

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

registers	H	L
AX	00	00
BX	00	00
CX	00	0A
DX	00	00
CS	0700	
IP	0108	
SS	0700	
SP	FFFE	
BP	0000	
SI	0000	
DI	0000	
DS	0700	
ES	0700	

0700:0100	0700:0100
07100: EB 235 6	JMP 0104h
07101: 02 002 0	OR AL, [BX + SI]
07102: 0A 010 NEWL	MOV CL, [00102h]
07103: 00 000 NULL	MOV AL, 09h
07104: 8A 138 2	IMUL CL
07105: 0E 014 7	MOV CL, 05h
07106: 02 002 0	IDIV CL
07107: 01 001 0	ADD AL, 020h
07108: B0 176 7	MOV [00103h], AL
07109: 09 009 TAB	HLT
0710A: F6 246 0	RET
0710B: E9 233 0	NOP
0710C: B1 177 7	NOP
0710D: 05 005 2	NOP
0710E: F6 246 0	NOP
0710F: F9 249 0	NOP
07110: 04 004 2	NOP
07111: 20 032 SPA	NOP
07112: A2 162 6	NOP
07113: 03 003 7	NOP
07114: 01 001 0	...

original sou...

```

01
02 org 100h
03 JMP start
04
05 tc db 10
06 result db ?
07
08 start:
09
10 mov cl, tc
11 mov al, 09
12 imul cl
13 mov cl, 05
14 idiv cl
15 add al, 32
16 mov result, al
17
18
19 HLT
20 ret
21
22

```

screen source reset aux vars debug stack

emulator: celcius.com_

file math debug view external virtual devices virtual drive help

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

registers	H	L
AX	00	09
BX	00	00
CX	00	0A
DX	00	00
CS	0700	
IP	010A	
SS	0700	
SP	FFFE	
BP	0000	
SI	0000	
DI	0000	
DS	0700	
ES	0700	

0700:010A	0700:010A
07100: EB 235 6	JMP 0104h
07101: 02 002 0	OR AL, [BX + SI]
07102: 0A 010 NEWL	MOV CL, [00102h]
07103: 00 000 NULL	MOV AL, 09h
07104: 8A 138 2	IMUL CL
07105: 0E 014 7	MOV CL, 05h
07106: 02 002 0	IDIV CL
07107: 01 001 0	ADD AL, 020h
07108: B0 176 7	MOV [00103h], AL
07109: 09 009 TAB	HLT
0710A: F6 246 2	RET
0710B: E9 233 0	NOP
0710C: B1 177 7	NOP
0710D: 05 005 2	NOP
0710E: F6 246 2	NOP
0710F: F9 249 2	NOP
07110: 04 004 2	NOP
07111: 20 032 SPA	NOP
07112: A2 162 6	NOP
07113: 03 003 7	NOP
07114: 01 001 0	...

original sou...

```

01 org 100h
02 JMP start
03
04 tc db 10
05 result db ?
06
07 start:
08
09 mov cl, tc
10 mov al, 09
11 imul cl
12
13 mov cl, 05
14 idiv cl
15 add al, 32
16 mov result, al
17
18 HLT
19 ret
20
21
22

```

screen source reset aux vars debug stack

emulator: celcius.com_

file math debug view external virtual devices virtual drive help

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

registers	H	L
AX	00	5A
BX	00	00
CX	00	0A
DX	00	00
CS	0700	
IP	010C	
SS	0700	
SP	FFFE	
BP	0000	
SI	0000	
DI	0000	
DS	0700	
ES	0700	

0700:010C	0700:010C
07100: EB 235 6	JMP 0104h
07101: 02 002 0	OR AL, [BX + SI]
07102: 0A 010 NEWL	MOV CL, [00102h]
07103: 00 000 NULL	MOV AL, 09h
07104: 8A 138 2	IMUL CL
07105: 0E 014 7	MOV CL, 05h
07106: 02 002 0	IDIV CL
07107: 01 001 0	ADD AL, 020h
07108: B0 176 7	MOV [00103h], AL
07109: 09 009 TAB	HLT
0710A: F6 246 2	RET
0710B: E9 233 0	NOP
0710C: B1 177 7	NOP
0710D: 05 005 2	NOP
0710E: F6 246 2	NOP
0710F: F9 249 2	NOP
07110: 04 004 2	NOP
07111: 20 032 SPA	NOP
07112: A2 162 6	NOP
07113: 03 003 7	NOP
07114: 01 001 0	...

original sou...

```

01 org 100h
02 JMP start
03
04 tc db 10
05 result db ?
06
07 start:
08
09 mov cl, tc
10 mov al, 09
11 imul cl
12
13 mov cl, 05
14 idiv cl
15 add al, 32
16 mov result, al
17
18 HLT
19 ret
20
21
22

```

screen source reset aux vars debug stack

emulator: celcius.com_

file math debug view external virtual devices virtual drive help

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

registers

	H	L
AX	00	5A
BX	00	00
CX	00	05
DX	00	00
CS	0700	
IP	010E	
SS	0700	
SP	FFFE	
BP	0000	
SI	0000	
DI	0000	
DS	0700	
ES	0700	

0700:010E

07100:	EB	235	δ
07101:	02	002	⊖
07102:	0A	010	NEWL
07103:	00	000	NULL
07104:	8A	138	è
07105:	0E	014	⌘
07106:	02	002	⊖
07107:	01	001	⊖
07108:	B0	176	⌘
07109:	09	009	IAB
0710A:	F6	246	÷
0710B:	E9	233	⊖
0710C:	B1	177	⌘
0710D:	05	005	⊕
0710E:	F6	246	÷
0710F:	F9	249	-
07110:	04	004	♦
07111:	20	032	SPA
07112:	A2	162	ó
07113:	03	003	♥
07114:	01	001	⊖

0700:010E

```

JMP 0104h
OR AL, [BX + SI]
MOV CL, [00102h]
MOV AL, 09h
IMUL CL
MOV CL, 05h
IDIV CL
ADD AL, 020h
MOV [00103h], AL
HLT
RET
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
...

```

original sou...

```

01
02 org 100h
03 JMP start
04
05 tc db 10
06 result db ?
07
08 start:
09
10 mov cl, tc
11 mov al, 09
12 imul cl
13 mov cl, 05
14 idiv cl
15 add al, 32
16 mov result, al
17
18
19 HLT
20 ret
21
22

```

screen source reset aux vars debug stack

emulator: celcius.com_

file math debug view external virtual devices virtual drive help

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

registers

	H	L
AX	00	12
BX	00	00
CX	00	05
DX	00	00
CS	0700	
IP	0110	
SS	0700	
SP	FFFE	
BP	0000	
SI	0000	
DI	0000	
DS	0700	
ES	0700	

0700:0110

07100:	EB	235	δ
07101:	02	002	⊖
07102:	0A	010	NEWL
07103:	00	000	NULL
07104:	8A	138	è
07105:	0E	014	⌘
07106:	02	002	⊖
07107:	01	001	⊖
07108:	B0	176	⌘
07109:	09	009	IAB
0710A:	F6	246	÷
0710B:	E9	233	⊖
0710C:	B1	177	⌘
0710D:	05	005	⊕
0710E:	F6	246	÷
0710F:	F9	249	-
07110:	04	004	♦
07111:	20	032	SPA
07112:	A2	162	ó
07113:	03	003	♥
07114:	01	001	⊖

0700:0110

```

JMP 0104h
OR AL, [BX + SI]
MOV CL, [00102h]
MOV AL, 09h
IMUL CL
MOV CL, 05h
IDIV CL
ADD AL, 020h
MOV [00103h], AL
HLT
RET
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
...

```

original sou...

```

01
02 org 100h
03 JMP start
04
05 tc db 10
06 result db ?
07
08 start:
09
10 mov cl, tc
11 mov al, 09
12 imul cl
13 mov cl, 05
14 idiv cl
15 add al, 32
16 mov result, al
17
18
19 HLT
20 ret
21
22

```

screen source reset aux vars debug stack

emulator: celcius.com_

file math debug view external virtual devices virtual drive help

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

registers		0700:0112	0700:0112	original sou...
	H L			
AX	00 32	07100: EB 235 6	JMP 0104h	01
BX	00 00	07101: 02 002 0	OR AL, [BX + SI]	02 org 100h
CX	00 05	07102: 0A 010 NEWL	MOV CL, [00102h]	03 JMP start
DX	00 00	07103: 00 000 NULL	MOV AL, 09h	04
CS	0700	07104: 8A 138 2	IMUL CL	05 tc db 10
IP	0112	07105: 0E 014 7	MOV CL, 05h	06 result db ?
SS	0700	07106: 02 002 0	IDIV CL	07
SP	FFFE	07107: 01 001 0	ADD AL, 020h	08 start:
BP	0000	07108: B0 176 8	MOV [00103h], AL	09
SI	0000	07109: 09 009 TAB	HLT	10 mov cl, tc
DI	0000	0710A: F6 246 ÷	RET	11 mov al, 09
DS	0700	0710B: E9 233 0	NOP	12 imul cl
ES	0700	0710C: B1 177 8	NOP	13 mov cl, 05
		0710D: 05 005 2	NOP	14 idiv cl
		0710E: F6 246 ÷	NOP	15 add al, 32
		0710F: F9 249 ·	NOP	16 mov result, al
		07110: 04 004 ♦	NOP	17
		07111: 20 032 SPA	NOP	18
		07112: A2 162 6	NOP	19 HLT
		07113: 03 003 ♥	NOP	20 ret
		07114: 01 001 0	...	21

screen source reset aux vars debug stack

emulator: celcius.com_

file math debug view external virtual devices virtual drive help

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

registers		0700:0115	0700:0100	original sou...
	H L			
AX	00 32	07115: F4 244 7	JMP 0104h	01
BX	00 00	07116: C3 195 7	OR DH, [BP + SI]	02 org 100h
CX	00 05	07117: 90 144 E	MOV CL, [00102h]	03 JMP start
DX	00 00	07118: 90 144 E	MOV AL, 09h	04
CS	0700	07119: 90 144 E	IMUL CL	05 tc db 10
IP	0115	0711A: 90 144 E	MOV CL, 05h	06 result db ?
SS	0700	0711B: 90 144 E	IDIV CL	07
SP	FFFE	0711C: 90 144 E	ADD AL, 020h	08 start:
BP	0000	0711D: 90 144 E	MOV [00103h], AL	09
SI	0000	0711E: 90 144 E	HLT	10 mov cl, tc
DI	0000	0711F: 90 144 E	RET	11 mov al, 09
DS	0700	07120: 90 144 E	NOP	12 imul cl
ES	0700	07121: 90 144 E	NOP	13 mov cl, 05
		07122: 90 144 E	NOP	14 idiv cl
		07123: 90 144 E	NOP	15 add al, 32
		07124: 90 144 E	NOP	16 mov result, al
		07125: 90 144 E	NOP	17
		07126: 90 144 E	NOP	18
		07127: 90 144 E	NOP	19 HLT
		07128: 90 144 E	NOP	20 ret
		07129: 90 144 E	...	21

screen source reset aux vars debug stack

Review Questions

1. What will be the effect if we consider 'mul' instruction instead of 'imul' instruction in the code? Explain with an example.

The **MUL** instruction multiplies unsigned numbers. **IMUL** multiplies signed numbers.

2. Does your code take signed numbers as input?

As my code uses **IMUL** thus we consider signed numbers.

3. Write a similar kind of code for converting temperature from °F to °C.

```
org 100h

tc db 0
result db ?

mov cl, tc
sub cl, 32
mov al, 5
imul cl
mov cl, 9
idiv cl
mov result, al

HLT
ret
```

2. Factorial of number :

```

org 100h

MOV SI, 0200H
MOV CX, 0005
MOV AX, 0001
MOV DX, 0000

L1: MUL CX
    LOOP L1

MOV [SI], AX
MOV [SI+1], DX

HLT
ret

```

registers

	H	L
AX	00	00
BX	00	00
CX	00	17
DX	00	00
CS	0700	
IP	0100	
SS	0700	
SP	FFFE	
BP	0000	
SI	0000	
DI	0000	
DS	0700	
ES	0700	

0700:0100

07100:	BE	190	J
07101:	00	000	NULL
07102:	02	002	0
07103:	B9	185	J
07104:	05	005	0
07105:	00	000	NULL
07106:	B8	184	J
07107:	01	001	0
07108:	00	000	NULL
07109:	BA	186	J
0710A:	00	000	NULL
0710B:	00	000	NULL
0710C:	F7	247	0
0710D:	E1	225	0
0710E:	E2	226	0
0710F:	FC	252	0
07110:	89	137	0
07111:	04	004	0
07112:	89	137	0
07113:	54	084	0
07114:	01	001	0

0700:0100

```

MOV SI, 00200h
MOV CX, 00005h
MOV AX, 00001h
MOV DX, 00000h
MUL CX
LOOP 010Ch
MOV [SI], AX
MOV [SI] + 01h, DX
HLT
RET
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP

```

original sou...

```

01 org 100h
02
03
04 MOV SI, 0200H
05 MOV CX, 0005
06 MOV AX, 0001
07 MOV DX, 0000
08
09 L1: MUL CX
10 LOOP L1
11
12 MOV [SI], AX
13 MOV [SI+1], DX
14
15
16 HLT
17 ret
18
19
20
21
22

```

registers

	H	L
AX	00	00
BX	00	00
CX	00	17
DX	00	00
CS	0700	
IP	0103	
SS	0700	
SP	FFFE	
BP	0000	
SI	0200	
DI	0000	
DS	0700	
ES	0700	

0700:0103

07100:	BE	190	J
07101:	00	000	NULL
07102:	02	002	0
07103:	B9	185	J
07104:	05	005	0
07105:	00	000	NULL
07106:	B8	184	J
07107:	01	001	0
07108:	00	000	NULL
07109:	BA	186	J
0710A:	00	000	NULL
0710B:	00	000	NULL
0710C:	F7	247	0
0710D:	E1	225	0
0710E:	E2	226	0
0710F:	FC	252	0
07110:	89	137	0
07111:	04	004	0
07112:	89	137	0
07113:	54	084	0
07114:	01	001	0

0700:0103

```

MOV SI, 00200h
MOV CX, 00005h
MOV AX, 00001h
MOV DX, 00000h
MUL CX
LOOP 010Ch
MOV [SI], AX
MOV [SI] + 01h, DX
HLT
RET
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP

```

original sou...

```

12 org 100h
13
14 MOV SI, 0200H
15 MOV CX, 0005
16 MOV AX, 0001
17 MOV DX, 0000
18
19 L1: MUL CX
20 LOOP L1
21
22 MOV [SI], AX
23 MOV [SI+1], DX
24
25
26 HLT
27 ret
28
29
30
31
32

```

registers

	H	L
AX	00	00
BX	00	00
CX	00	05
DX	00	00
CS	0700	
IP	0106	
SS	0700	
SP	FFFE	
BP	0000	
SI	0200	
DI	0000	
DS	0700	
ES	0700	

0700:0106

07100:	BE	190	J
07101:	00	000	NULL
07102:	02	002	0
07103:	B9	185	J
07104:	05	005	0
07105:	00	000	NULL
07106:	B8	184	J
07107:	01	001	0
07108:	00	000	NULL
07109:	BA	186	J
0710A:	00	000	NULL
0710B:	00	000	NULL
0710C:	F7	247	0
0710D:	E1	225	0
0710E:	E2	226	0
0710F:	FC	252	0
07110:	89	137	0
07111:	04	004	0
07112:	89	137	0
07113:	54	084	0
07114:	01	001	0

0700:0106

```

MOV SI, 00200h
MOV CX, 00005h
MOV AX, 00001h
MOV DX, 00000h
MUL CX
LOOP 010Ch
MOV [SI], AX
MOV [SI] + 01h, DX
HLT
RET
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP

```

original sou...

```

4 MOV SI, 0200H
5 MOV CX, 0005
6 MOV AX, 0001
7 MOV DX, 0000
8
9 L1: MUL CX
10 LOOP L1
11
12 MOV [SI], AX
13 MOV [SI+1], DX
14
15
16 HLT
17 ret
18
19
20
21
22

```

registers		0700:0109	0700:0109	original sou...
	H L			
AX	00 01	07100: BE 190 J	MOV SI, 00200h	1
BX	00 00	07101: 00 000 NULL	MOV CX, 00005h	2
CX	00 05	07102: 02 002 0	MOV AX, 00001h	3
DX	00 00	07103: B9 185	MOV DX, 00000h	4
CS	0700	07104: 05 005 5	MUL CX	5
IP	0109	07105: 00 000 NULL	LOOP 010Ch	6
SS	0700	07106: B8 184 3	MOV [SI], AX	7
SP	FFFE	07107: 01 001 0	MOV [SI] + 01h, DX	8
BP	0000	07108: 00 000 NULL	HLT	9
SI	0200	07109: BA 186	RET	10
DI	0000	0710A: 00 000 NULL	NOP	11
DS	0700	0710B: 00 000 NULL	NOP	12
ES	0700	0710C: F7 247 7	NOP	13
		0710D: E1 225 0	NOP	14
		0710E: E2 226 1	NOP	15
		0710F: FC 252 "	NOP	16
		07110: 89 137 9	NOP	17
		07111: 04 004 4	NOP	18
		07112: 89 137 9	NOP	19
		07113: 54 084 T	NOP	20
		07114: 01 001 0	...	21

screen source reset aux vars debug stack flags

registers		0700:010C	0700:010C	original sou...
	H L			
AX	00 01	07100: BE 190 J	MOV SI, 00200h	1
BX	00 00	07101: 00 000 NULL	MOV CX, 00005h	2
CX	00 05	07102: 02 002 0	MOV AX, 00001h	3
DX	00 00	07103: B9 185	MOV DX, 00000h	4
CS	0700	07104: 05 005 5	MUL CX	5
IP	010C	07105: 00 000 NULL	LOOP 010Ch	6
SS	0700	07106: B8 184 3	MOV [SI], AX	7
SP	FFFE	07107: 01 001 0	MOV [SI] + 01h, DX	8
BP	0000	07108: 00 000 NULL	HLT	9
SI	0200	07109: BA 186	RET	10
DI	0000	0710A: 00 000 NULL	NOP	11
DS	0700	0710B: 00 000 NULL	NOP	12
ES	0700	0710C: F7 247 7	NOP	13
		0710D: E1 225 0	NOP	14
		0710E: E2 226 1	NOP	15
		0710F: FC 252 "	NOP	16
		07110: 89 137 9	NOP	17
		07111: 04 004 4	NOP	18
		07112: 89 137 9	NOP	19
		07113: 54 084 T	NOP	20
		07114: 01 001 0	...	21

screen source reset aux vars debug stack flags

registers		0700:010E	0700:010E	original sou...
	H L			
AX	00 05	07100: BE 190 J	MOV SI, 00200h	1
BX	00 00	07101: 00 000 NULL	MOV CX, 00005h	2
CX	00 05	07102: 02 002 0	MOV AX, 00001h	3
DX	00 00	07103: B9 185	MOV DX, 00000h	4
CS	0700	07104: 05 005 5	MUL CX	5
IP	010E	07105: 00 000 NULL	LOOP 010Ch	6
SS	0700	07106: B8 184 3	MOV [SI], AX	7
SP	FFFE	07107: 01 001 0	MOV [SI] + 01h, DX	8
BP	0000	07108: 00 000 NULL	HLT	9
SI	0200	07109: BA 186	RET	10
DI	0000	0710A: 00 000 NULL	NOP	11
DS	0700	0710B: 00 000 NULL	NOP	12
ES	0700	0710C: F7 247 7	NOP	13
		0710D: E1 225 0	NOP	14
		0710E: E2 226 1	NOP	15
		0710F: FC 252 "	NOP	16
		07110: 89 137 9	NOP	17
		07111: 04 004 4	NOP	18
		07112: 89 137 9	NOP	19
		07113: 54 084 T	NOP	20
		07114: 01 001 0	...	21

screen source reset aux vars debug stack flags

registers		0700:010E	0700:010E	original sou...
	H L			
AX	00 14	07100: BE 190 J	MOV SI, 00200h	1
BX	00 00	07101: 00 000 NULL	MOV CX, 00005h	2
CX	00 04	07102: 02 002 0	MOV AX, 00001h	3
DX	00 00	07103: B9 185	MOV DX, 00000h	4
CS	0700	07104: 05 005 5	MUL CX	5
IP	010E	07105: 00 000 NULL	LOOP 010Ch	6
SS	0700	07106: B8 184 3	MOV [SI], AX	7
SP	FFFE	07107: 01 001 0	MOV [SI] + 01h, DX	8
BP	0000	07108: 00 000 NULL	HLT	9
SI	0200	07109: BA 186	RET	10
DI	0000	0710A: 00 000 NULL	NOP	11
DS	0700	0710B: 00 000 NULL	NOP	12
ES	0700	0710C: F7 247 7	NOP	13
		0710D: E1 225 0	NOP	14
		0710E: E2 226 1	NOP	15
		0710F: FC 252 "	NOP	16
		07110: 89 137 9	NOP	17
		07111: 04 004 4	NOP	18
		07112: 89 137 9	NOP	19
		07113: 54 084 T	NOP	20
		07114: 01 001 0	...	21

screen source reset aux vars debug stack flags

registers		0700:010E		0700:010E		original sou...
	H	L				
AX	00	3C	07100: BE 190 J	MOV SI, 00200h	1	
BX	00	00	07101: 00 000 NULL	MOV CX, 00005h	2	org 100h
CX	00	03	07102: 02 002 0	MOV AX, 00001h	3	
DX	00	00	07103: B9 185 JI	MOV DX, 00000h	4	MOV SI, 0200H
CS	0700		07104: 05 005 2	MUL CX	5	MOV CX, 0005
IP	010E		07105: 00 000 NULL	LOOP 010Ch	6	MOV AX, 0001
SS	0700		07106: B8 184 J	MOV [SI], AX	7	MOV DX, 0000
SP	FFFE		07107: 01 001 0	MOV [SI] + 01h, DX	8	
BP	0000		07108: 00 000 NULL	HLT	9	L1: MUL CX
SI	0200		07109: BA 186 II	RET	10	LOOP L1
DI	0000		0710A: 00 000 NULL	NOP	1	
DS	0700		0710B: 00 000 NULL	NOP	2	MOV [SI], AX
ES	0700		0710C: F7 247 2	NOP	3	MOV [SI+1], DX
			0710D: E1 225 0	NOP	4	
			0710E: E2 226 I	NOP	5	
			0710F: FC 252 n	NOP	6	
			07110: 89 137 2	NOP	7	HLT
			07111: 04 004 4	NOP	8	ret
			07112: 89 137 2	NOP	9	
			07113: 54 084 T	NOP	10	
			07114: 01 001 0	...	11	

screen source reset aux vars debug stack flags

registers		0700:010E		0700:010E		original sou...
	H	L				
AX	00	78	07100: BE 190 J	MOV SI, 00200h	1	
BX	00	00	07101: 00 000 NULL	MOV CX, 00005h	2	org 100h
CX	00	02	07102: 02 002 0	MOV AX, 00001h	3	
DX	00	00	07103: B9 185 JI	MOV DX, 00000h	4	MOV SI, 0200H
CS	0700		07104: 05 005 2	MUL CX	5	MOV CX, 0005
IP	010E		07105: 00 000 NULL	LOOP 010Ch	6	MOV AX, 0001
SS	0700		07106: B8 184 J	MOV [SI], AX	7	MOV DX, 0000
SP	FFFE		07107: 01 001 0	MOV [SI] + 01h, DX	8	
BP	0000		07108: 00 000 NULL	HLT	9	L1: MUL CX
SI	0200		07109: BA 186 II	RET	10	LOOP L1
DI	0000		0710A: 00 000 NULL	NOP	1	
DS	0700		0710B: 00 000 NULL	NOP	2	MOV [SI], AX
ES	0700		0710C: F7 247 2	NOP	3	MOV [SI+1], DX
			0710D: E1 225 0	NOP	4	
			0710E: E2 226 I	NOP	5	
			0710F: FC 252 n	NOP	6	
			07110: 89 137 2	NOP	7	HLT
			07111: 04 004 4	NOP	8	ret
			07112: 89 137 2	NOP	9	
			07113: 54 084 T	NOP	10	
			07114: 01 001 0	...	11	

screen source reset aux vars debug stack flags

registers		0700:010E		0700:010E		original sou...
	H	L				
AX	00	78	07100: BE 190 J	MOV SI, 00200h	1	
BX	00	00	07101: 00 000 NULL	MOV CX, 00005h	2	org 100h
CX	00	01	07102: 02 002 0	MOV AX, 00001h	3	
DX	00	00	07103: B9 185 JI	MOV DX, 00000h	4	MOV SI, 0200H
CS	0700		07104: 05 005 2	MUL CX	5	MOV CX, 0005
IP	010E		07105: 00 000 NULL	LOOP 010Ch	6	MOV AX, 0001
SS	0700		07106: B8 184 J	MOV [SI], AX	7	MOV DX, 0000
SP	FFFE		07107: 01 001 0	MOV [SI] + 01h, DX	8	
BP	0000		07108: 00 000 NULL	HLT	9	L1: MUL CX
SI	0200		07109: BA 186 II	RET	10	LOOP L1
DI	0000		0710A: 00 000 NULL	NOP	1	
DS	0700		0710B: 00 000 NULL	NOP	2	MOV [SI], AX
ES	0700		0710C: F7 247 2	NOP	3	MOV [SI+1], DX
			0710D: E1 225 0	NOP	4	
			0710E: E2 226 I	NOP	5	
			0710F: FC 252 n	NOP	6	
			07110: 89 137 2	NOP	7	HLT
			07111: 04 004 4	NOP	8	ret
			07112: 89 137 2	NOP	9	
			07113: 54 084 T	NOP	10	
			07114: 01 001 0	...	11	

screen source reset aux vars debug stack flags

Random Access Memory

0700:0200	update				
0700:0200	78	78	00	00	00
0700:0210	00	00	00	00	00
0700:0220	00	00	00	00	00
0700:0230	00	00	00	00	00

Review Questions

1. What will be the effect if we consider the number to be 0 in the given pseudocode?

We will be stuck in infinite code. As CX is already zero and will never be equal to zero but will be negative

2. Why the input number range is selected between 1 to 8? Can we extend the range by altering the ALP suitable for 8086?

Yes, we can extend the range by taking more registers for memory as after 8 factorial the 16 bit memory get exceeded, and the value overflows.

3. Repeat the same problem without using the instructions "LOOP". You can use the instructions like JZ or JNZ or any other registers as per your requirement. Evaluate the value of 7! using your new code.

```
org 100h
```

```
MOV SI, 0200H
MOV CX, 0007
MOV AX, 0001
MOV DX, 0000
```

```
L1: MUL CX
DEC CX
JNZ L1
```

```
MOV [SI], AX
MOV [SI+1], DX
```

```
HLT
ret
```

Random Access Memory	
0700:0200	u
0700:0200	B0 00 00
0700:0210	00 00 00
0700:0220	00 00 00

The screenshot shows a 8086 emulator interface. On the left, the registers window displays the state of various registers: AX (13 B0), BX (00 00), CX (00 00), DX (00 00), CS (0700), IP (0116), SS (0700), SP (FFFE), BP (0000), SI (0200), DI (0000), DS (0700), and ES (0700). The main window is divided into three panes. The top pane shows memory addresses from 07116 to 0712A, with values like F4 244, C3 195, 90 144, etc. The middle pane shows the assembly code being executed, with the instruction `HLT` highlighted. The bottom pane shows the original source code, which matches the pseudocode provided in the text. The code calculates 7! using a loop and stores the result in memory.