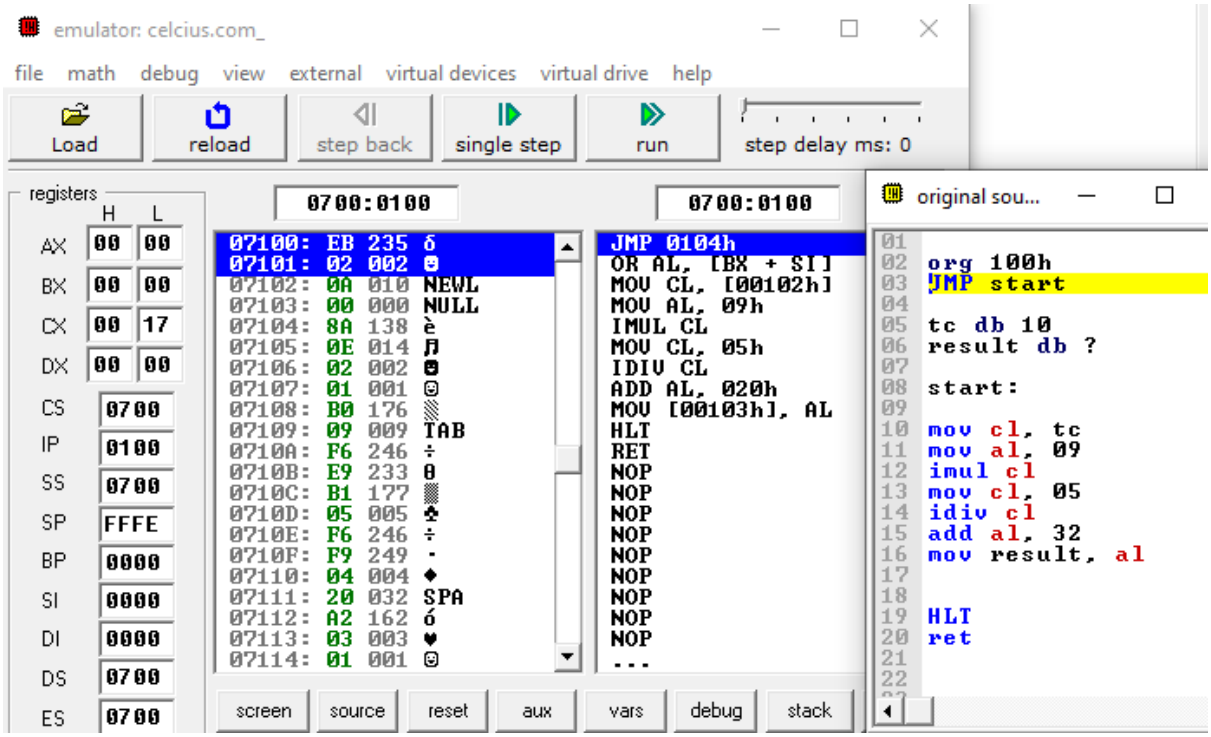
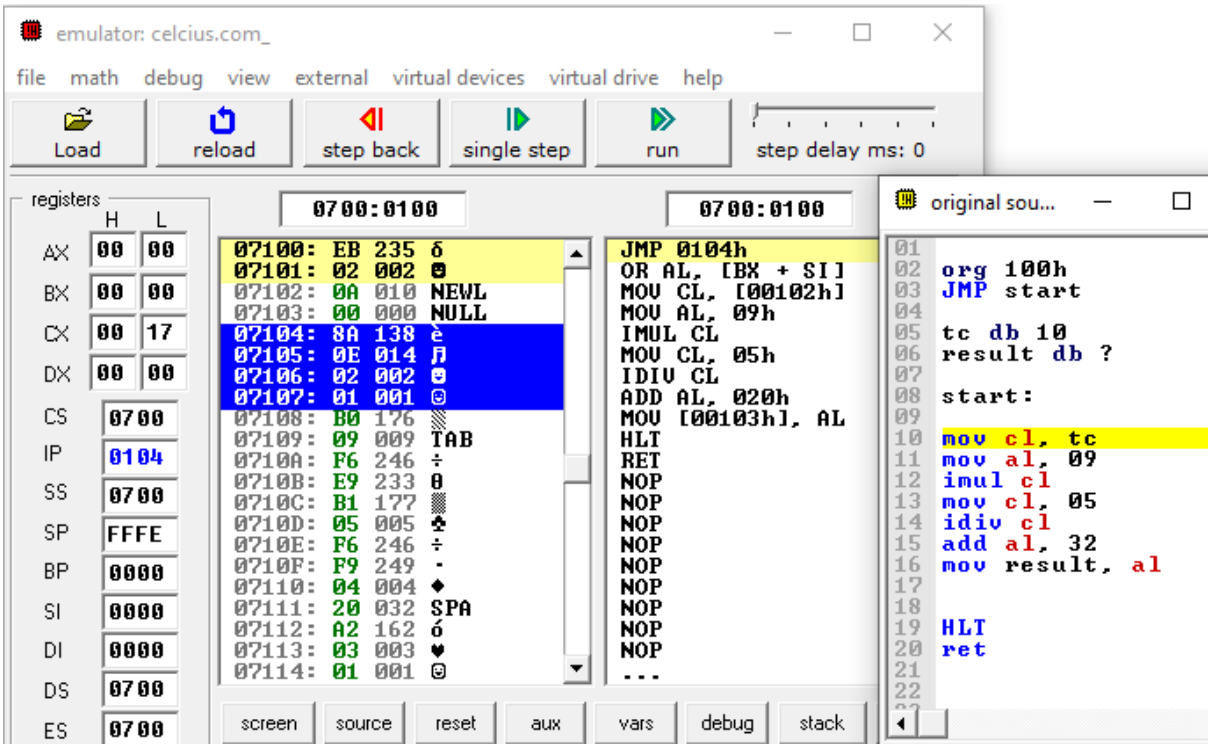
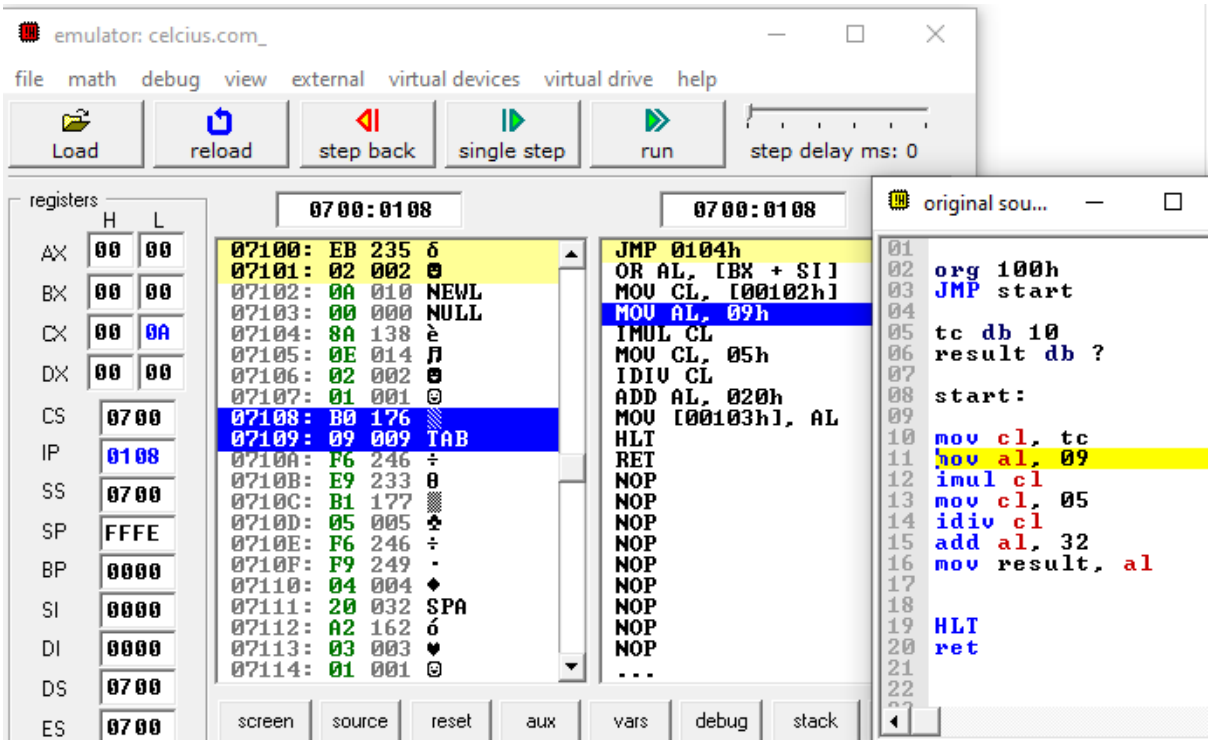
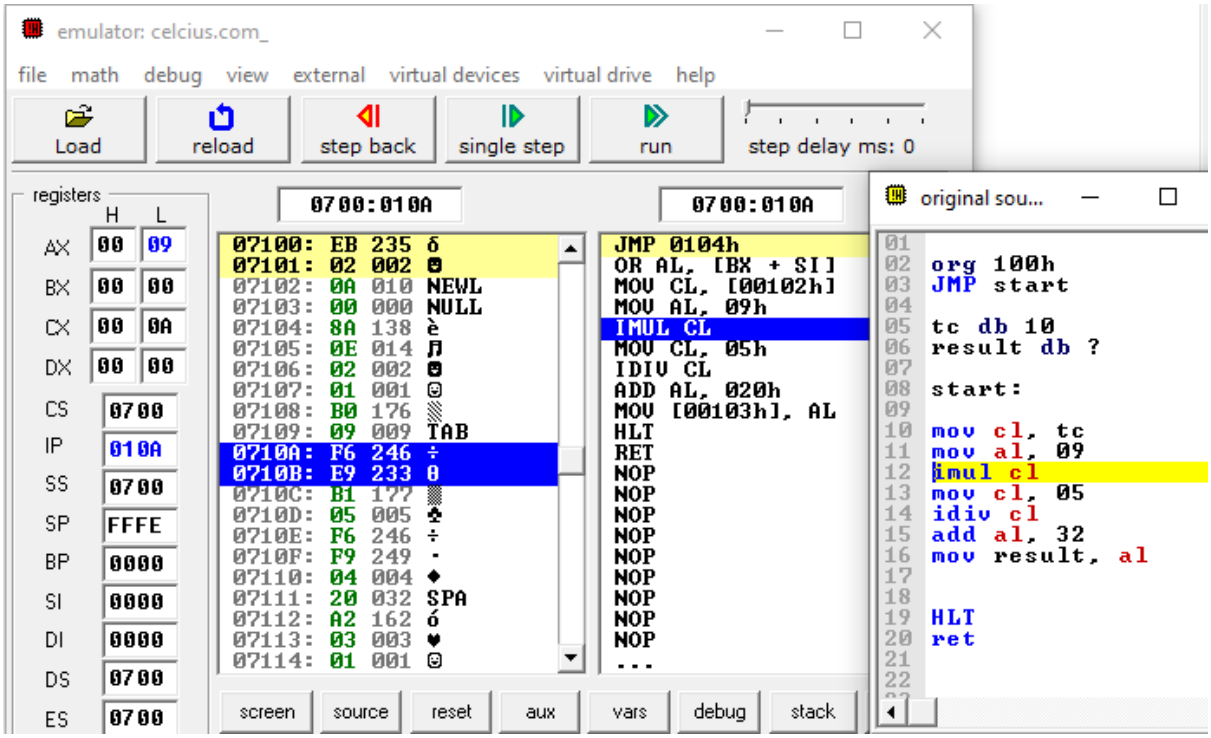
1. CODE : Centigrade (Celsius) to Fahrenheit

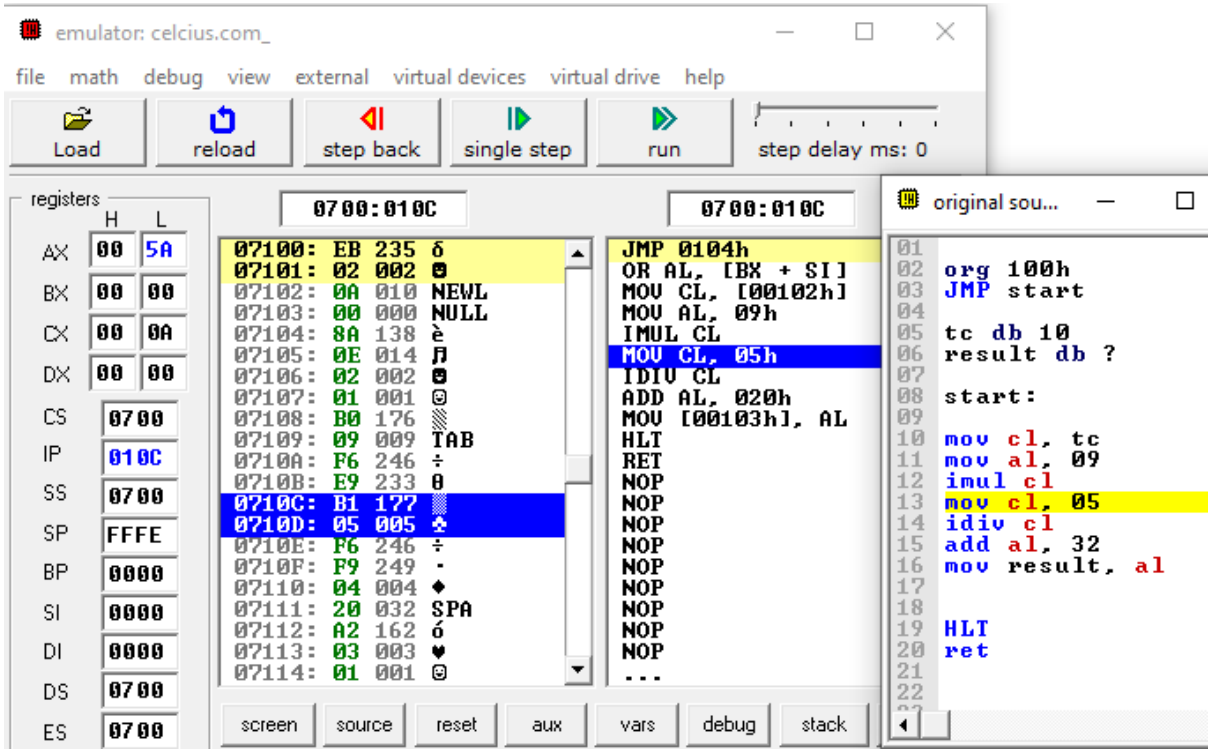
```
org 100h
JMP start

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
```

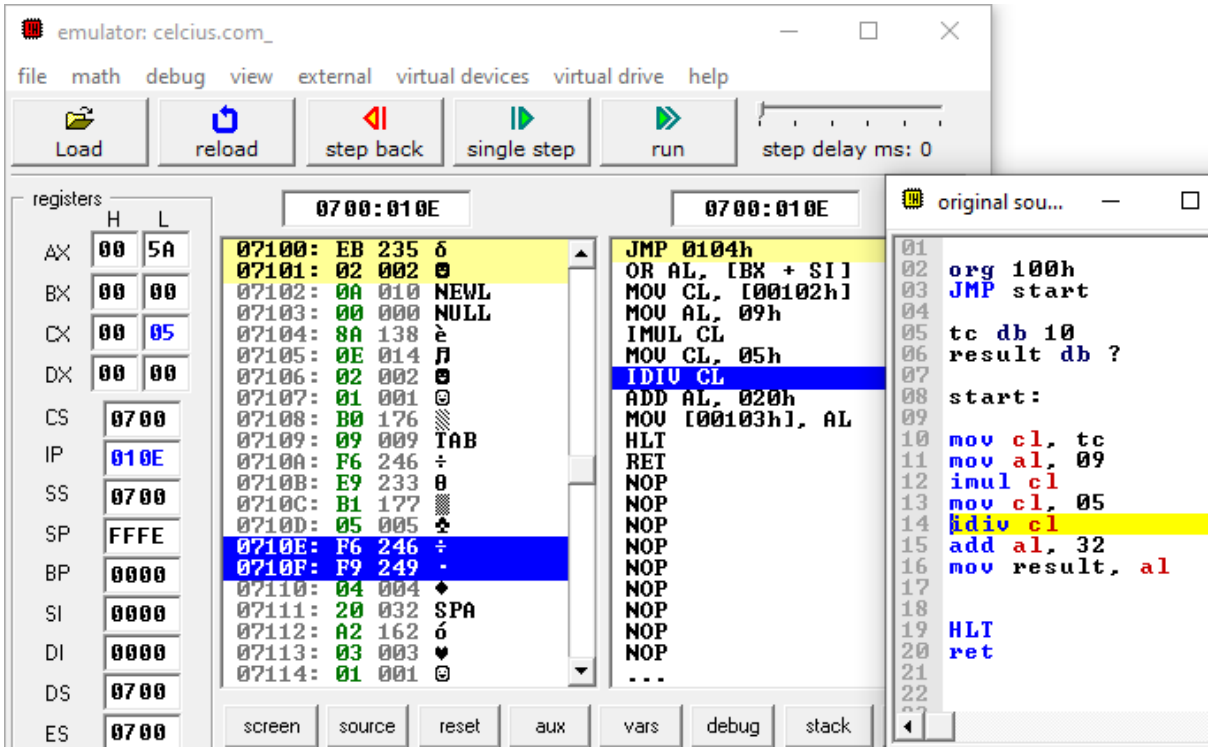


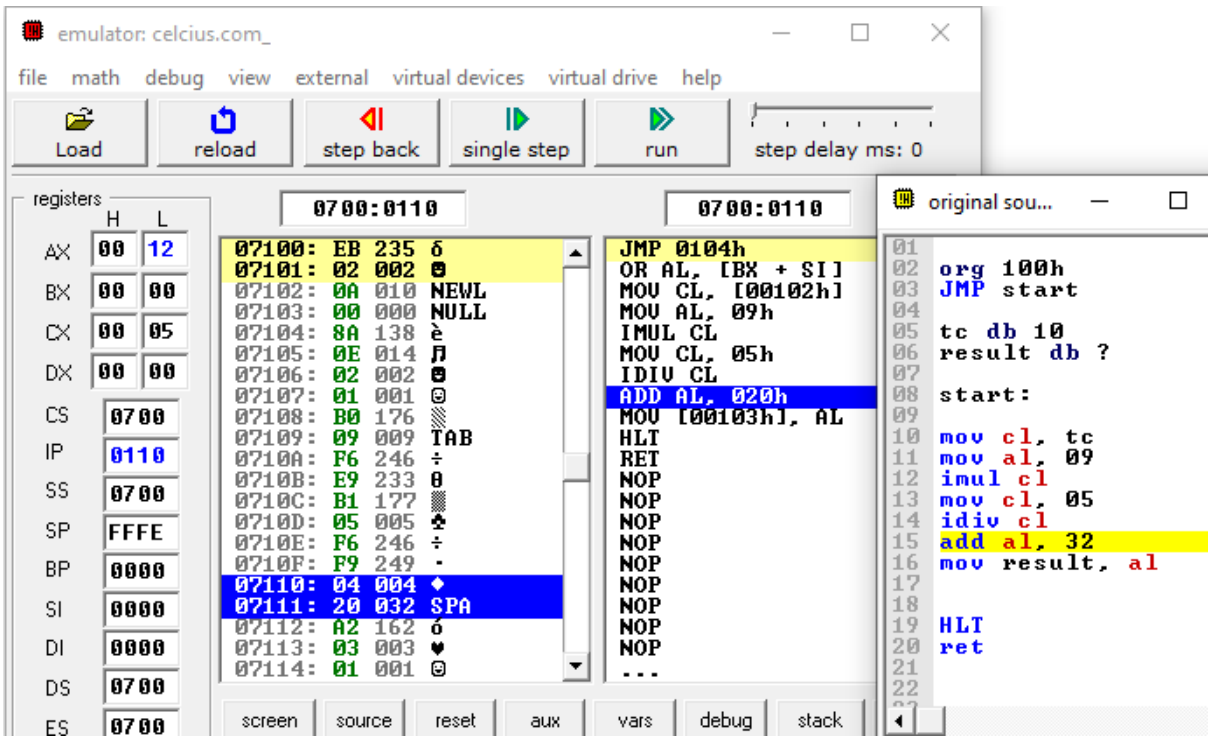

emulator: celcius.com_
file math debug view external virtual devices virtual drive help
Load reload step back single step run step delay ms: 0
registers
H L
AX 00 00
BX 00 00
CX 00 17
DX 00 00
CS 0700
IP 0104
SS 0700
SP FFFE
BP 0000
SI 0000
DI 0000
DS 0700
ES 0700
0700:0100
0700:0100
07100: EB 235 δ
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 TAB
0710A: F6 246 ÷
0710B: E9 233 θ
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 ☺
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
...
screen source reset aux vars debug stack
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

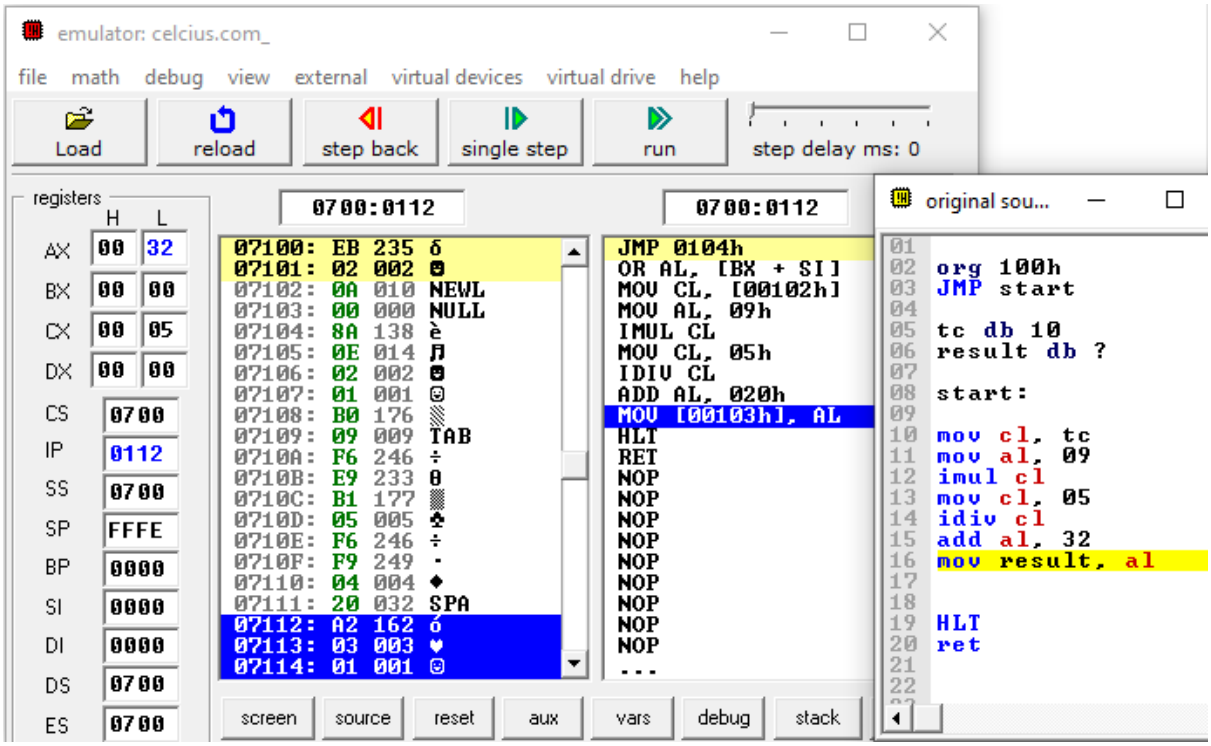

emulator: celcius.com_
file math debug view external virtual devices virtual drive help
Load reload step back single step run step delay ms: 0
registers
H L
AX 00 00
BX 00 00
CX 00 0A
DX 00 00
CS 0700
IP 0108
SS 0700
SP FFFE
BP 0000
SI 0000
DI 0000
DS 0700
ES 0700
0700:0108
0700:0108
07100: EB 235 δ
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 TAB
0710A: F6 246 ÷
0710B: E9 233 θ
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 ☺
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
...
screen source reset aux vars debug stack
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

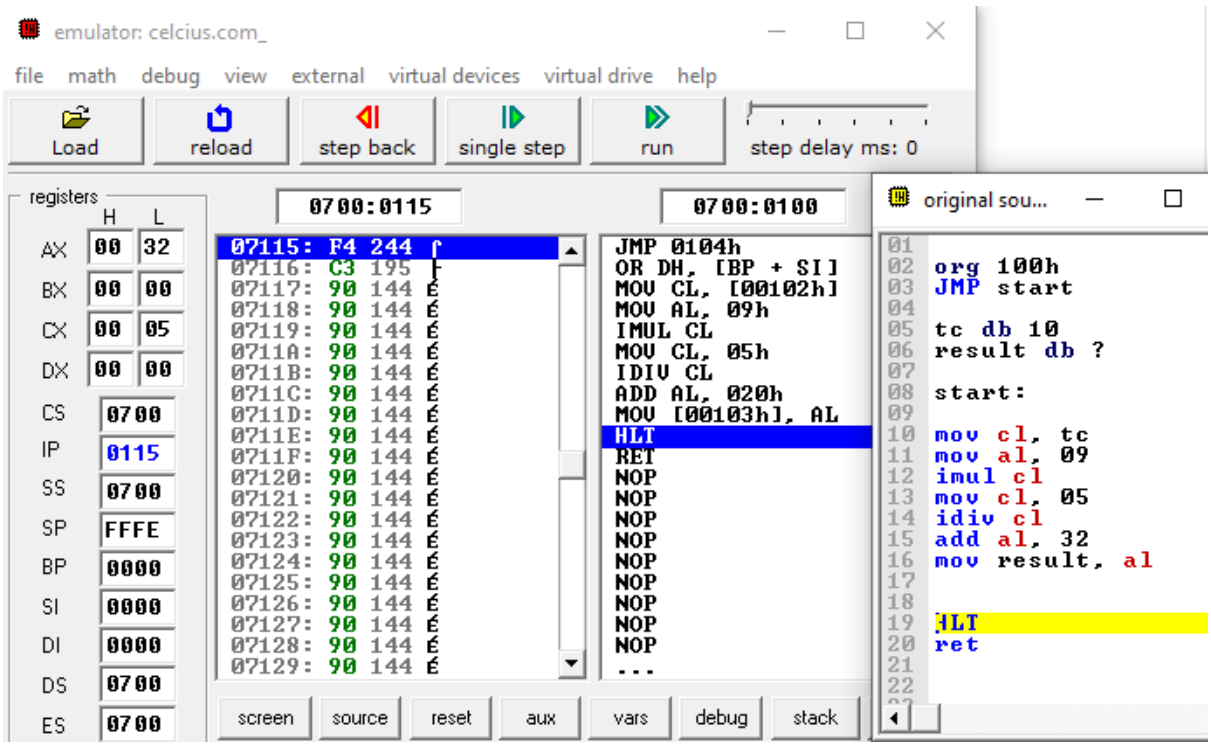

emulator: celcius.com_
file math debug view external virtual devices virtual drive help
Load reload step back single step run step delay ms: 0
registers
H L
AX 00 09
BX 00 00
CX 00 0A
DX 00 00
CS 0700
IP 010A
SS 0700
SP FFFE
BP 0000
SI 0000
DI 0000
DS 0700
ES 0700
0700:010A
0700:010A
07100: EB 235 δ
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 TAB
0710A: F6 246 ÷
0710B: E9 233 θ
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
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
...
screen source reset aux vars debug stack
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


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 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 TAB
0710A: F6 246 ÷
0710B: E9 233 θ
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
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
...
screen source reset aux vars debug stack
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


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
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 TAB
0710A: F6 246 ÷
0710B: E9 233 θ
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 ☺
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
...
screen source reset aux vars debug stack
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


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
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 TAB
0710A: F6 246 ÷
0710B: E9 233 θ
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 ☺
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
...
screen source reset aux vars debug stack
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


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 32
BX 00 00
CX 00 05
DX 00 00
CS 0700
IP 0112
SS 0700
SP FFFE
BP 0000
SI 0000
DI 0000
DS 0700
ES 0700
0700:0112
0700:0112
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 TAB
0710A: F6 246 ÷
0710B: E9 233 Θ
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
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
...
screen source reset aux vars debug stack
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


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 32
BX 00 00
CX 00 05
DX 00 00
CS 0700
IP 0115
SS 0700
SP FFFE
BP 0000
SI 0000
DI 0000
DS 0700
ES 0700
0700:0115
0700:0100
07115: F4 244
07116: C3 195
07117: 90 144 É
07118: 90 144 É
07119: 90 144 É
0711A: 90 144 É
0711B: 90 144 É
0711C: 90 144 É
0711D: 90 144 É
0711E: 90 144 É
0711F: 90 144 É
07120: 90 144 É
07121: 90 144 É
07122: 90 144 É
07123: 90 144 É
07124: 90 144 É
07125: 90 144 É
07126: 90 144 É
07127: 90 144 É
07128: 90 144 É
07129: 90 144 É
JMP 0104h
OR DH, [BP + 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
...
screen source reset aux vars debug stack
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

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?

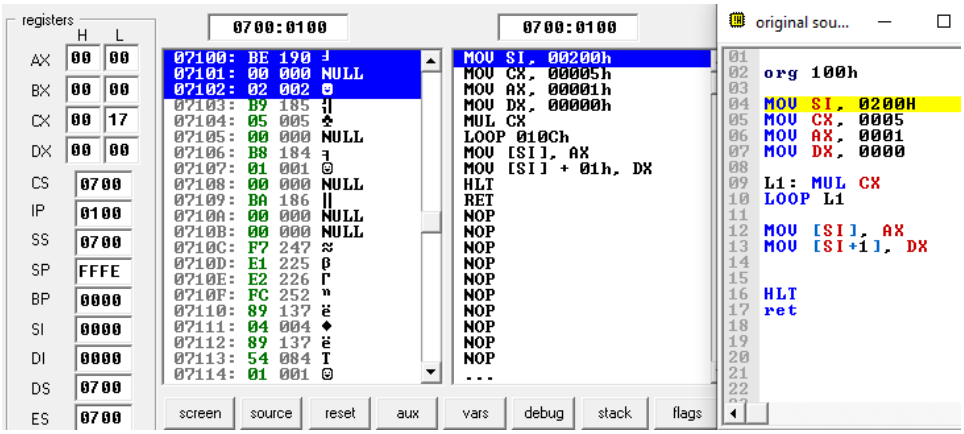
As my code uses **IMUL** thus we consider signed numbers.

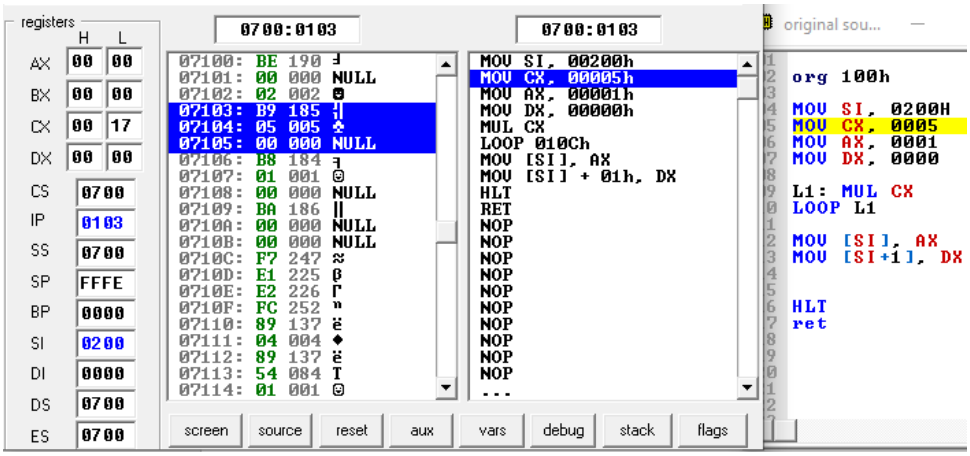
3. Write a similar kind of code for converting temperature from ºF to ºC.

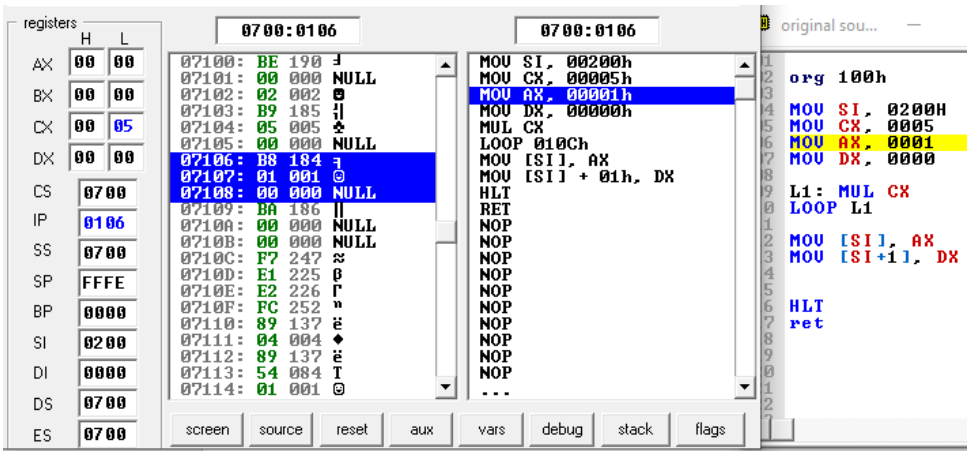
```
org 100h

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
```

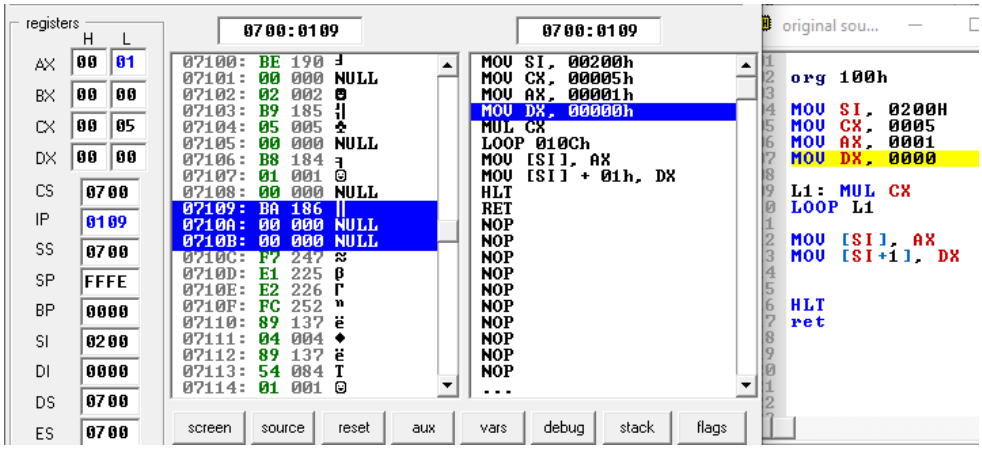
2. Factorial of number :

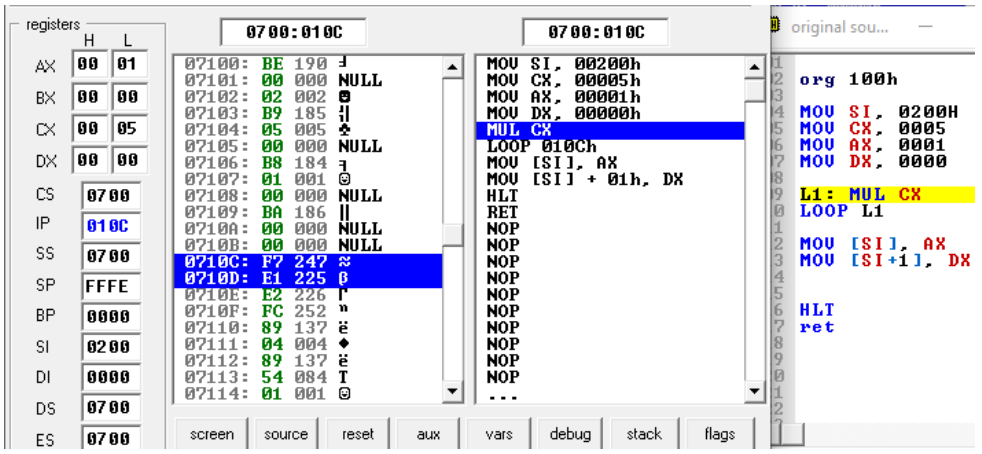
```
org 100h

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
```

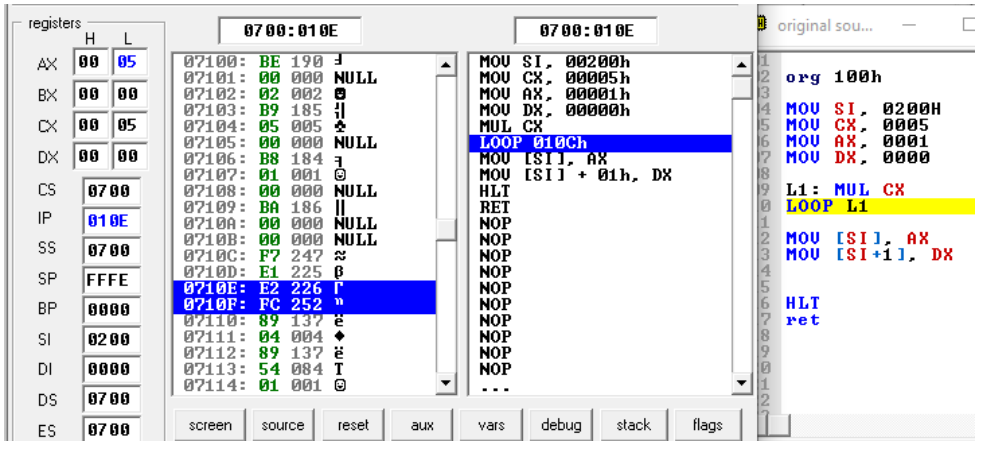


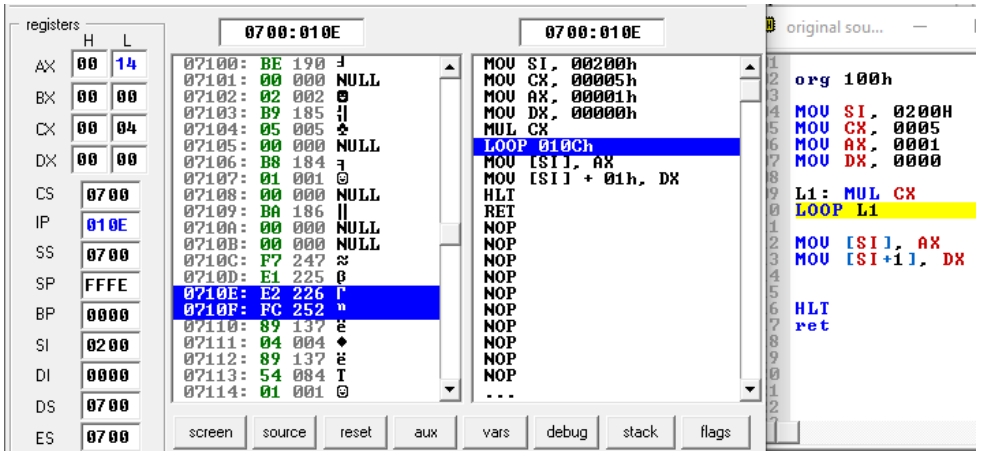


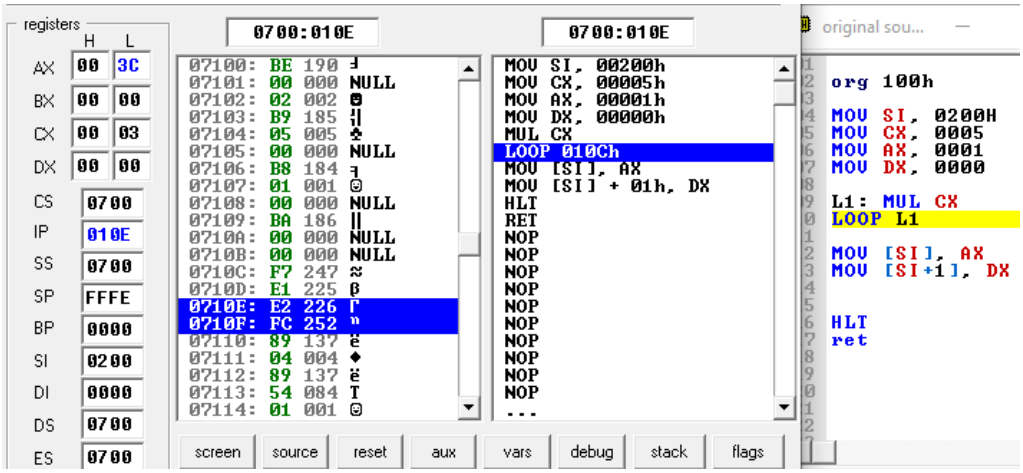


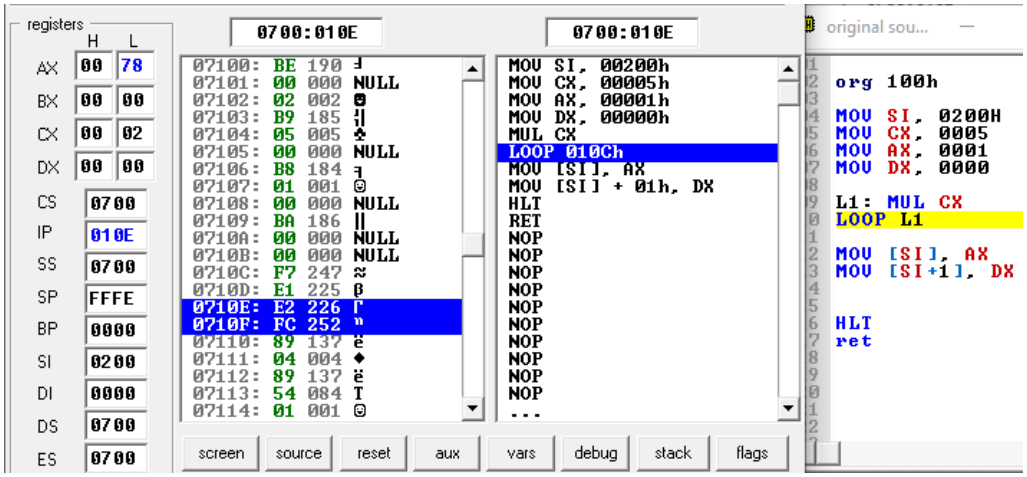

registers
H L
AX 00 01
BX 00 00
CX 00 05
DX 00 00
CS 0700
IP 0109
SS 0700
SP FFFE
BP 0000
SI 0200
DI 0000
DS 0700
ES 0700
0700:0109
0700:0109
07100: BE 190
07101: 00 000 NULL
07102: 02 002
07103: B9 185
07104: 05 005
07105: 00 000 NULL
07106: B8 184
07107: 01 001
07108: 00 000 NULL
07109: BA 186
0710A: 00 000 NULL
0710B: 00 000 NULL
0710C: F7 247
0710D: E1 225
0710E: E2 226
0710F: FC 252
07110: 89 137
07111: 04 004
07112: 89 137
07113: 54 084
07114: 01 001
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
...
screen
source
reset
aux
vars
debug
stack
flags
original sou...
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

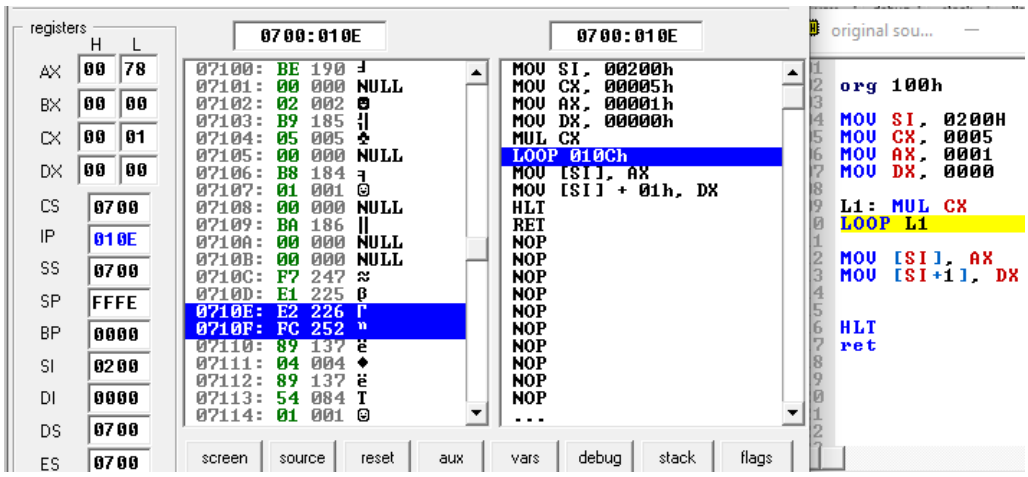

registers
H L
AX 00 01
BX 00 00
CX 00 05
DX 00 00
CS 0700
IP 010C
SS 0700
SP FFFE
BP 0000
SI 0200
DI 0000
DS 0700
ES 0700
0700:010C
0700:010C
07100: BE 190
07101: 00 000 NULL
07102: 02 002
07103: B9 185
07104: 05 005
07105: 00 000 NULL
07106: B8 184
07107: 01 001
07108: 00 000 NULL
07109: BA 186
0710A: 00 000 NULL
0710B: 00 000 NULL
0710C: F7 247
0710D: E1 225
0710E: E2 226
0710F: FC 252
07110: 89 137
07111: 04 004
07112: 89 137
07113: 54 084
07114: 01 001
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
...
screen
source
reset
aux
vars
debug
stack
flags
original sou...
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

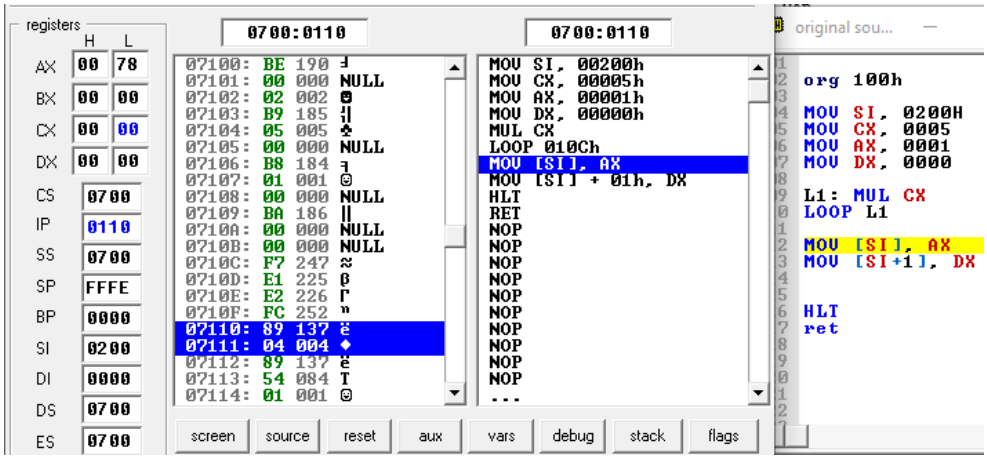

registers
H L
AX 00 05
BX 00 00
CX 00 05
DX 00 00
CS 0700
IP 010E
SS 0700
SP FFFE
BP 0000
SI 0200
DI 0000
DS 0700
ES 0700
0700:010E
0700:010E
07100: BE 190
07101: 00 000 NULL
07102: 02 002
07103: B9 185
07104: 05 005
07105: 00 000 NULL
07106: B8 184
07107: 01 001
07108: 00 000 NULL
07109: BA 186
0710A: 00 000 NULL
0710B: 00 000 NULL
0710C: F7 247
0710D: E1 225
0710E: E2 226
0710F: FC 252
07110: 89 137
07111: 04 004
07112: 89 137
07113: 54 084
07114: 01 001
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
...
screen
source
reset
aux
vars
debug
stack
flags
original sou...
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

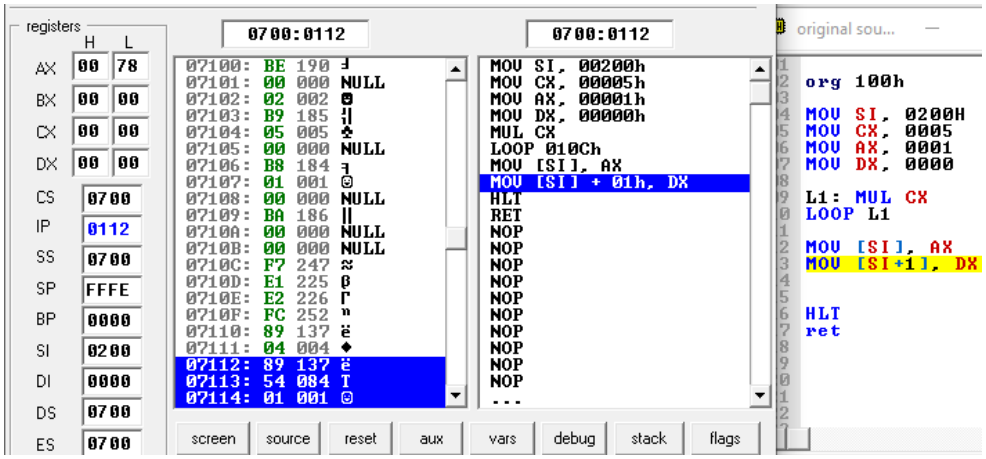

registers
H L
AX 00 14
BX 00 00
CX 00 04
DX 00 00
CS 0700
IP 010E
SS 0700
SP FFFE
BP 0000
SI 0200
DI 0000
DS 0700
ES 0700
0700:010E
0700:010E
07100: BE 190
07101: 00 000 NULL
07102: 02 002
07103: B9 185
07104: 05 005
07105: 00 000 NULL
07106: B8 184
07107: 01 001
07108: 00 000 NULL
07109: BA 186
0710A: 00 000 NULL
0710B: 00 000 NULL
0710C: F7 247
0710D: E1 225
0710E: E2 226
0710F: FC 252
07110: 89 137
07111: 04 004
07112: 89 137
07113: 54 084
07114: 01 001
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
...
screen
source
reset
aux
vars
debug
stack
flags
original sou...
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

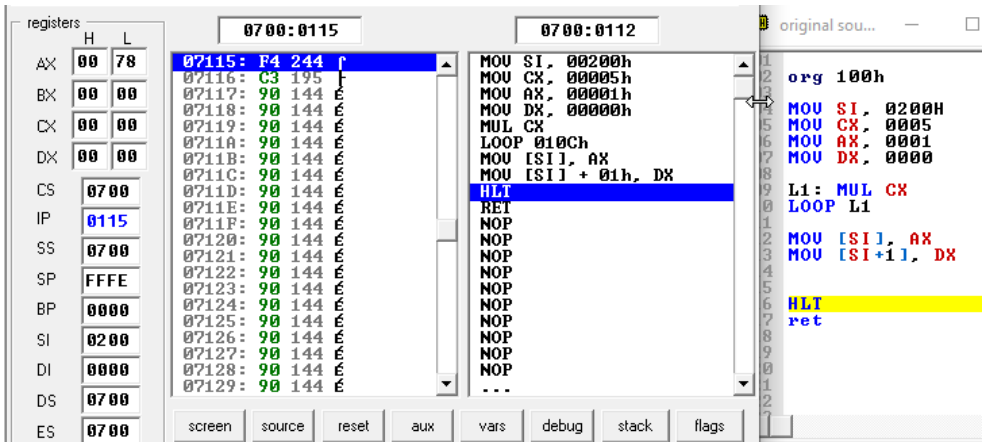

registers
H L
AX 00 3C
BX 00 00
CX 00 03
DX 00 00
CS 0700
IP 010E
SS 0700
SP FFFE
BP 0000
SI 0200
DI 0000
DS 0700
ES 0700
0700:010E
0700:010E
07100: BE 190
07101: 00 000 NULL
07102: 02 002
07103: B9 185
07104: 05 005
07105: 00 000 NULL
07106: B8 184
07107: 01 001
07108: 00 000 NULL
07109: BA 186
0710A: 00 000 NULL
0710B: 00 000 NULL
0710C: F7 247
0710D: E1 225
0710E: E2 226
0710F: FC 252
07110: 89 137
07111: 04 004
07112: 89 137
07113: 54 084 T
07114: 01 001
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
...
screen
source
reset
aux
vars
debug
stack
flags
original sou...
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

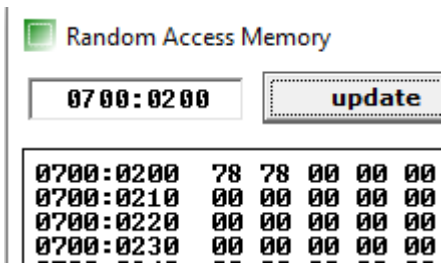

registers
H L
AX 00 78
BX 00 00
CX 00 02
DX 00 00
CS 0700
IP 010E
SS 0700
SP FFFE
BP 0000
SI 0200
DI 0000
DS 0700
ES 0700
0700:010E
0700:010E
07100: BE 190
07101: 00 000 NULL
07102: 02 002
07103: B9 185
07104: 05 005
07105: 00 000 NULL
07106: B8 184
07107: 01 001
07108: 00 000 NULL
07109: BA 186
0710A: 00 000 NULL
0710B: 00 000 NULL
0710C: F7 247
0710D: E1 225
0710E: E2 226
0710F: FC 252
07110: 89 137
07111: 04 004
07112: 89 137
07113: 54 084 T
07114: 01 001
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
...
screen
source
reset
aux
vars
debug
stack
flags
original sou...
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 78
BX 00 00
CX 00 01
DX 00 00
CS 0700
IP 010E
SS 0700
SP FFFE
BP 0000
SI 0200
DI 0000
DS 0700
ES 0700
0700:010E
0700:010E
07100: BE 190
07101: 00 000 NULL
07102: 02 002
07103: B9 185
07104: 05 005
07105: 00 000 NULL
07106: B8 184
07107: 01 001
07108: 00 000 NULL
07109: BA 186
0710A: 00 000 NULL
0710B: 00 000 NULL
0710C: F7 247
0710D: E1 225
0710E: E2 226
0710F: FC 252
07110: 89 137
07111: 04 004
07112: 89 137
07113: 54 084 T
07114: 01 001
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
...
screen
source
reset
aux
vars
debug
stack
flags
original sou...
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 78
BX 00 00
CX 00 00
DX 00 00
CS 0700
IP 0110
SS 0700
SP FFFE
BP 0000
SI 0200
DI 0000
DS 0700
ES 0700
0700:0110
0700:0110
07100: BE 190 ╛
07101: 00 000 NULL
07102: 02 002 ☻
07103: B9 185 ╣
07104: 05 005 ♣
07105: 00 000 NULL
07106: B8 184 ╕
07107: 01 001 ☺
07108: 00 000 NULL
07109: BA 186 ║
0710A: 00 000 NULL
0710B: 00 000 NULL
0710C: F7 247 ≈
0710D: E1 225 ß
0710E: E2 226 Γ
0710F: FC 252 ⁿ
07110: 89 137 ë
07111: 04 004 ♦
07112: 89 137 ë
07113: 54 084 T
07114: 01 001 ☺
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
...
screen
source
reset
aux
vars
debug
stack
flags
original sou...
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 78
BX 00 00
CX 00 00
DX 00 00
CS 0700
IP 0112
SS 0700
SP FFFE
BP 0000
SI 0200
DI 0000
DS 0700
ES 0700
0700:0112
0700:0112
07100: BE 190 ╛
07101: 00 000 NULL
07102: 02 002 ☻
07103: B9 185 ╣
07104: 05 005 ♣
07105: 00 000 NULL
07106: B8 184 ╕
07107: 01 001 ☺
07108: 00 000 NULL
07109: BA 186 ║
0710A: 00 000 NULL
0710B: 00 000 NULL
0710C: F7 247 ≈
0710D: E1 225 ß
0710E: E2 226 Γ
0710F: FC 252 ⁿ
07110: 89 137 ë
07111: 04 004 ♦
07112: 89 137 ë
07113: 54 084 T
07114: 01 001 ☺
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
...
screen
source
reset
aux
vars
debug
stack
flags
original sou...
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 78
BX 00 00
CX 00 00
DX 00 00
CS 0700
IP 0115
SS 0700
SP FFFE
BP 0000
SI 0200
DI 0000
DS 0700
ES 0700
0700:0115
0700:0112
07115: F4 244 ⌠
07116: C3 195 ├
07117: 90 144 É
07118: 90 144 É
07119: 90 144 É
0711A: 90 144 É
0711B: 90 144 É
0711C: 90 144 É
0711D: 90 144 É
0711E: 90 144 É
0711F: 90 144 É
07120: 90 144 É
07121: 90 144 É
07122: 90 144 É
07123: 90 144 É
07124: 90 144 É
07125: 90 144 É
07126: 90 144 É
07127: 90 144 É
07128: 90 144 É
07129: 90 144 É
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
...
screen
source
reset
aux
vars
debug
stack
flags
original sou...
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


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
```