

Problem One — Code Analysis

Code:

```
ORG 100h

MOV AX, 1

MOV CX, data

START_LOOP:

    MUL CX

    LOOP START_LOOP data

DW 0005h

END
```

Explanation of Functionality:

The program initializes AX = 1 and loads CX with the memory value data = 0005h. The MUL CX instruction multiplies AX by CX each iteration. The LOOP instruction decrements CX and repeats the loop until CX = 0.

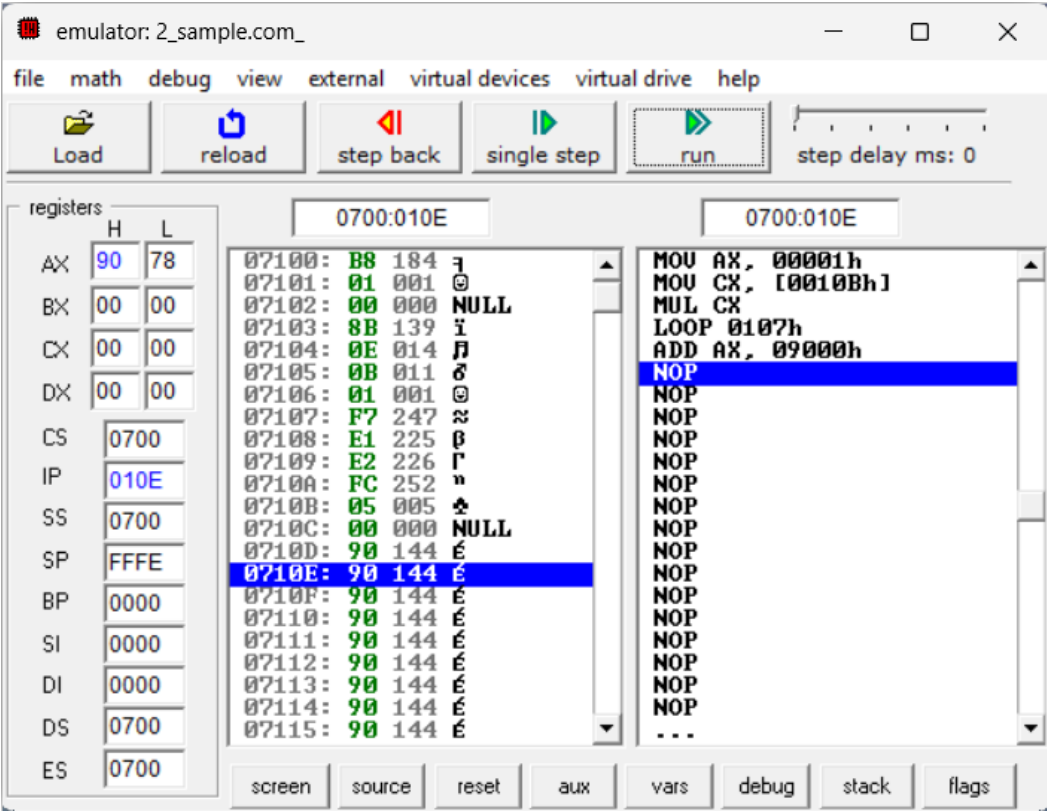
Behavior summary:

$AX = 1 \times 5 \times 4 \times 3 \times 2 \times 1 = 120$ (decimal)

Register Behavior:

- AX: Updated after each multiplication; final value is the factorial of 5.
- CX: Initially loaded with 5, then decremented each iteration.

You can see the result of factorial in register **AL** :



Problem Two — Summation Program

Problem Description:

Write a program that reads two bytes: a start value and a range, then sums all numbers from start to start + range -

Code:

```
ORG 100h

MOV BL, 5      ; Start Value
MOV BH, 0
MOV CL, 10     ; Range Value
MOV CH, 0

MOV AX, 0      ; accumulator
sum_loop:
    ADD AX, BX
    INC BX
    LOOP sum_loop

; Display result
MOV BL, 10
DIV BL

MOV BH, AH

ADD AL, 48
MOV DL, AL
MOV AH, 2
INT 21h

MOV DL, BH
ADD DL, 48
MOV AH, 2
INT 21h

RET
```

Result of the loop is printed on the Screen: 95 (result from the lab example)

The screenshot displays an x86 emulator window titled "emulator: noname.com_". The interface includes a menu bar (file, math, debug, view, external, virtual devices, virtual drive, help) and a toolbar with buttons for Load, reload, step back, single step, run, and a step delay slider set to 0 ms.

Registers:

	H	L
AX	02	35
BX	05	0A
CX	00	00
DX	00	35
CS	F400	
IP	0154	
SS	0700	
SP	FFFA	
BP	0000	
SI	0000	
DI	0000	
DS	0700	
ES	0700	

Memory (F400:0154):

F4150:	FF	255	RES
F4151:	FF	255	RES
F4152:	CD	205	=
F4153:	20	032	SPA
F4154:	CF	207	±
F4155:	00	000	NULL
F4156:	00	000	NULL
F4157:	00	000	NULL
F4158:	00	000	NULL
F4159:	00	000	NULL
F415A:	00	000	NULL
F415B:	00	000	NULL
F415C:	00	000	NULL
F415D:	00	000	NULL
F415E:	00	000	NULL
F415F:	00	000	NULL
F4160:	FF	255	RES
F4161:	FF	255	RES
F4162:	CD	205	=
F4163:	1A	026	→
F4164:	CF	207	±
F4165:	00	000	NULL

Screen Window:

The screen window, titled "emul...", shows the number "95". Below the screen are buttons for "clear screen" and "change font". The assembly code visible at the bottom of the screen is:

```
ADD [BX + SI], AL
ADD [BX + SI], AL
...
```

Problem Three — Adding Two 10-Byte Numbers

Description:

Write a program that stores two 10-byte numbers, adds them byte by byte with carry, and stores the result at memory location 500h.

Program Code:

```

ORG 100h

; 10-Byte Addition Routine
MOV     SI, OFFSET num1 + 9
MOV     DI, OFFSET num2 + 9
MOV     BX, 500h
MOV     CX, 10
CLC

add_loop:
    MOV     AL, [SI]
    ADC     AL, [DI]
    MOV     [BX], AL
    DEC     SI
    DEC     DI
    INC     BX

    LOOP    add_loop

; Write final carry byte

MOV     AL, 0
ADC     AL, 0
MOV     [BX], AL

RET

;input Section

num1 DB 0, 1, 2, 3, 4, 5, 6, 7, 8, 9
num2 DB 10, 9, 8, 7, 6, 5, 4, 3, 2, 1

```

Result can be seen in address 500 in memory for this example:

Random Access Memory

0700:0500 update table list

Address	Value
0700:0500	0A 0A 0A 0A 0A 0A 0A 0A 0A-0A 0A 00 00 00 00 00 00
0700:0510	00 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00
0700:0520	00 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00
0700:0530	00 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00
0700:0540	00 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00
0700:0550	00 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00
0700:0560	00 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00