



Vidyavardhini's College of Engineering & Technology
Department of Artificial Intelligence and Data Science (AI&DS)

Name:	Pranav Shetty
Roll No:	53
Class/Sem:	SE/IV
Experiment No.:	2B
Title:	Program for calculating factorial using assembly language
Date of Performance:	30/01/2024
Date of Submission:	06/02/2024
Marks:	
Sign of Faculty:	



Vidyavardhini's College of Engineering & Technology

Department of Artificial Intelligence and Data Science (AI&DS)

Aim: Program to calculate the Factorial of a number.

Theory:

To calculate the factorial of any number, we use MUL instruction. Here, initially, we initialize the first register by value 1. The second register is initialized by the value of the second register. After multiplication, decrement the value of the second register and repeat the multiplying step till the second register value becomes zero. The result is stored in the first register.

Algorithm:

1. Start.
2. Set AX=01H, and BX with the value whose factorial we want to find.
3. Multiply AX and BX.
4. Decrement BX=BX-1.
5. Repeat steps 3 and 4 till BX=0.
6. Stop.

Code:

```
ORG 100H

MOV AX, 00H

MOV BX, 04H

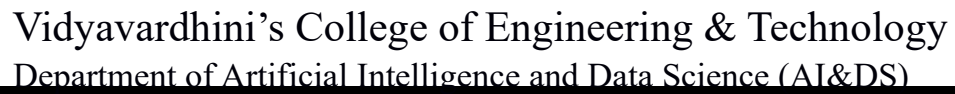
MOV CX, 02H

L1: ADD AX, BX

    DEC CX

    JNZ L1

RET
```



The screenshot shows the NoName.com emulator interface. At the top, the title bar reads "emulator: noname.com". Below it is a menu bar with "file", "math", "debug", "view", "external", "virtual devices", "virtual drive", and "help". A toolbar contains icons for "Load", "reload", "step back", "single step", "run", and a "step delay ms: 0" input field.

The main window is divided into several sections. On the left is the "registers" section, showing a table of 16-bit registers with their high (H) and low (L) bytes. The "IP" (Instruction Pointer) register is highlighted with the value 0154.

	H	L
AX	00	18
BX	00	00
CX	00	0C
DX	00	00
CS	F400	
IP	0154	
SS	0700	
SP	FFFA	
BP	0000	
SI	0000	
DI	0000	
DS	0700	
ES	0700	

In the center is a memory dump window titled "F400:0154". It displays a list of memory addresses and their contents, along with a disassembly of the instructions. The instruction at address F4154 is highlighted in blue.

Address	Hex	Dec	Comment
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

On the right is another memory dump window titled "F400:0154", showing the BIOS interrupt vector table. The instruction at address 020h is highlighted in blue.

Address	Hex	Dec	Comment
BIOS DI			
INT 020h			
IRET			
ADD [BX + SI], AL			
ADD [BX + SI], AL			
ADD [BX + SI], AL			
ADD [BX + SI], AL			
ADD [BX + SI], AL			
ADD BH, BH			
DEC BP			
SBB CL, BH			
ADD [BX + SI], AL			
ADD [BX + SI], AL			
ADD [BX + SI], AL			
ADD [BX + SI], AL			
ADD [BX + SI], AL			
ADD BH, BH			
DEC BP			
ADD BH, CL			
ADD [BX + SI], AL			
ADD [BX + SI], AL			
...			

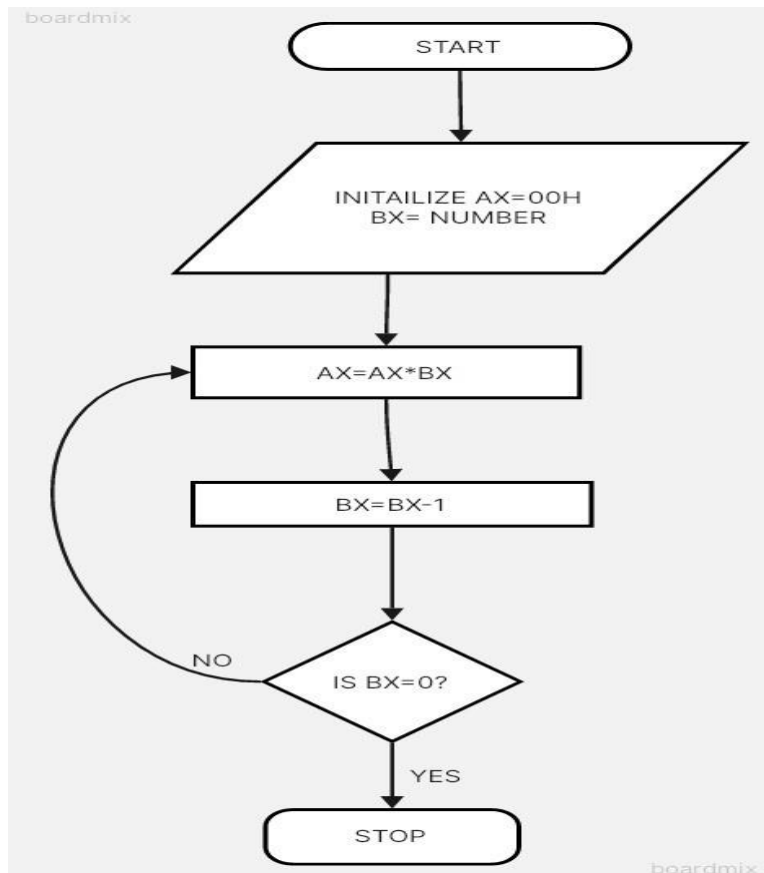
At the bottom of the window is a toolbar with buttons for "screen", "source", "reset", "aux", "vars", "debug", "stack", and "flags".



Vidyavardhini's College of Engineering & Technology

Department of Artificial Intelligence and Data Science (AI&DS)

Flowchart:





Conclusion:

1. Explain shift instructions.
2. Explain rotate instructions.

1)The 8086 can perform two types of Shift operations; the logical shift and the arithmetic shift. There are four shift operations (SHL, SAL, SHR, and SAR).

2)The 8086 can perform two types of rotate operations; the rotate without carry and the rotate through carry. There are four rotate operations (ROL, ROR, RCL, and RCR). ROL shifts each bit of a register to the left. The highest bit is copied into both the Carry flag and into the lowest bit of the register