National University of Computer and Emerging Sciences

Lab Manual

"Subroutines"

COMPUTER ORGANIZATION AND ASSEMBLY LANGUAGE

SOLVED 20L-0921

Course Instructor	Miss Aleena Ahmed
Lab Instructor(s)	Maham Saleem
Section	3E
Semester	Fall 2021

Department of Computer Science
FAST-NU, Lahore, Pakistan

Task 1

Dry run the code given in q1.asm, and answer the questions asked in comments. You are not allowed to run the code in debugger.

Answers in .asm file

Task 2

See sum_sub_routine.asm file.

1. In this file, there is a sub routine that takes as parameter the address of array and the number of elements in it.
2. The sub routine finds the sum of elements and stores the sum in memory.
3. This routine is using some registers such as bx, bp, ax, cx for the task. When the routine exits, the values of these registers are not the original values.

To Do: Change the sub routine so that the original values of these register (those registers that are used by the routine) is unchanged after running the routine. You must not use memory for storing the registers; also, you are not allowed to change the logic. The program must run fine after you have added your own code (Hint: Temporarily push the registers on the stack and pop them before returning from the routine. Do not use pusha / popa instructions).

Answers in .asm file

Task 3:

Write the sub-routine to calculate factorial. The sub-routine should take as parameter the number to calculate the factorial and returns factorial in AX register.

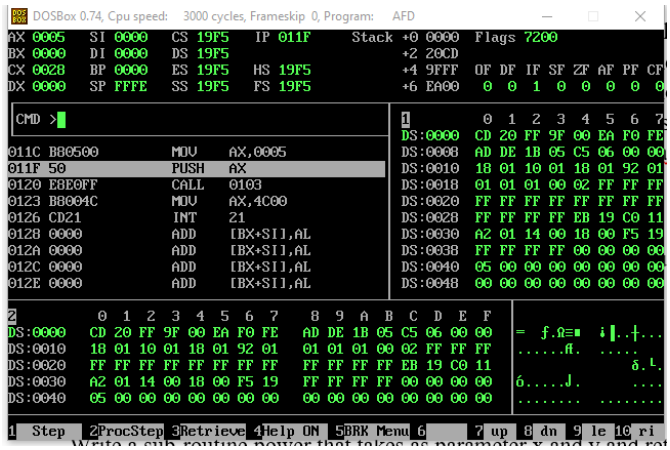
Code:

f - Notepad

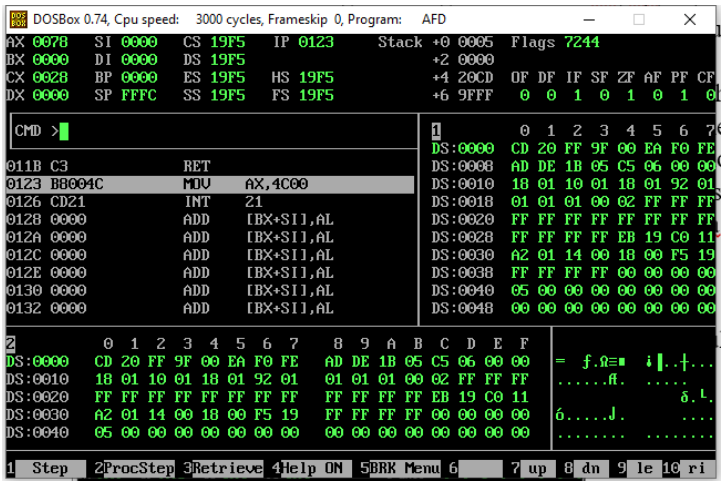
File Edit Format View Help

```
[org 0x0100]
jmp start
factorial:
push bp
mov bp,sp
push cx
mov cx,ax
sub cx,1

l1: cmp cx,1
jz done
mul cx
dec cx
call l1

done:
pop cx
pop bp
ret

start :
mov ax,5
push ax
call factorial

mov ax,4c00h
int 0x21
```

Before



After



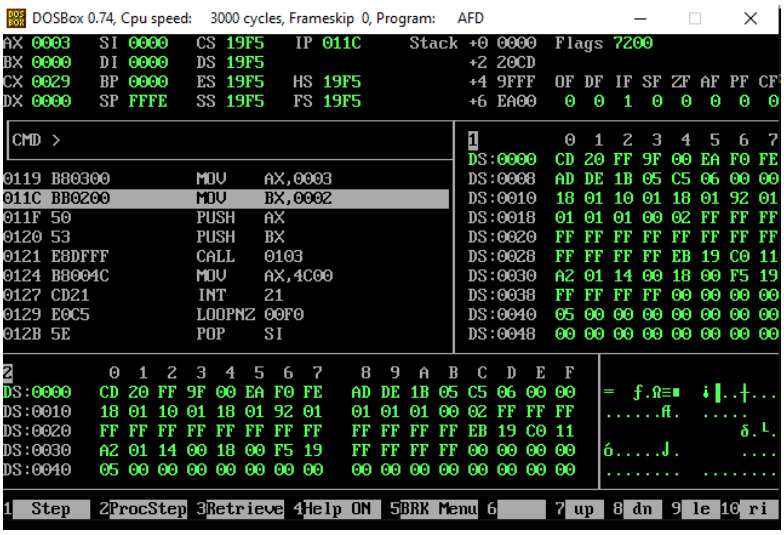
Task 4:

Write a sub-routine power that takes as parameter x and y and returns the answer of x raise to the power of y in AX register.

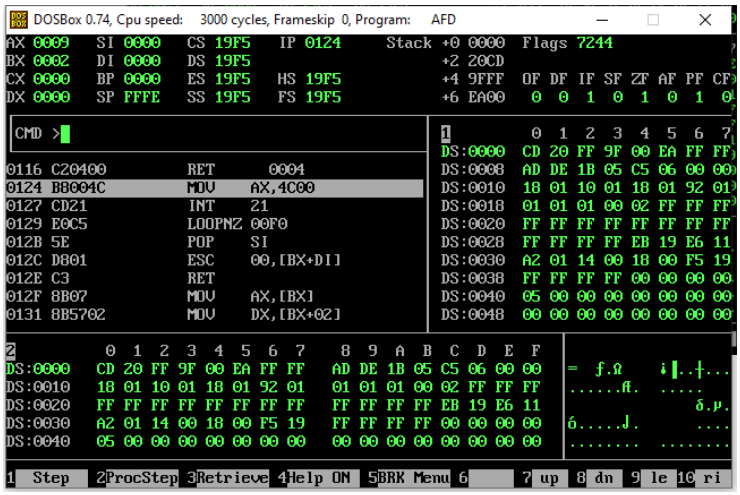
Code:

```
[org 0x0100]

jmp start
power:
push bp
mov bp,sp
push bx
push cx
mov cx,ax

l1:
mul cx
dec bx
cmp bx,1
jnz l1

pop cx
pop bx
pop bp

ret 4

start:
mov ax,3
mov bx,2
push ax
push bx
call power

mov ax,0x4c00
int 21h
```

Before:


DOSBox 0.74, Cpu speed: 3000 cycles, Frameskip 0, Program: AFD
AX 0003 SI 0000 CS 19F5 IP 011C Stack +0 0000 Flags 7200
BX 0000 DI 0000 DS 19F5 +2 20CD
CX 0029 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 B80300 MOV AX,0003
011C BB0200 MOV BX,0002
011F 50 PUSH AX
0120 53 PUSH BX
0121 EBDFFF CALL 0103
0124 B8004C MOV AX,4C00
0127 CD21 INT 21
0129 E0C5 LOOPNZ 00F0
012B 5E POP SI
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 1B 01 10 01 18 01 92 01
DS:0010 1B 01 10 01 18 01 92 01 01 01 01 00 02 FF FF FF
DS:0018 01 01 01 00 02 FF FF FF FF FF FF FF FF FF FF FF
DS:0020 FF FF FF FF FF FF FF FF FF FF FF FF EB 19 C0 11
DS:0028 FF FF FF FF EB 19 C0 11 A2 01 14 00 18 00 F5 19
DS:0030 A2 01 14 00 18 00 F5 19 FF FF FF FF 00 00 00 00
DS:0038 FF FF FF FF 00 00 00 00 05 00 00 00 00 00 00 00
DS:0040 05 00 00 00 00 00 00 00 00 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

After:


DOSBox 0.74, Cpu speed: 3000 cycles, Frameskip 0, Program: AFD
AX 0009 SI 0000 CS 19F5 IP 0124 Stack +0 0000 Flags 7244
BX 0002 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 >
0116 C20400 RET 0004
0124 B8004C MOV AX,4C00
0127 CD21 INT 21
0129 E0C5 LOOPNZ 00F0
012B 5E POP SI
012C D801 ESC 00,[BX+DI]
012E C3 RET
012F 8B07 MOV AX,[BX]
0131 8B5702 MOV DX,[BX+02]
DS:0000 CD 20 FF 9F 00 EA FF FF AD DE 1B 05 C5 06 00 00
DS:0008 AD DE 1B 05 C5 06 00 00 1B 01 10 01 18 01 92 01
DS:0010 1B 01 10 01 18 01 92 01 01 01 01 00 02 FF FF FF
DS:0018 01 01 01 00 02 FF FF FF FF FF FF FF FF FF FF FF
DS:0020 FF FF FF FF FF FF FF FF FF FF FF FF EB 19 E6 11
DS:0028 FF FF FF FF EB 19 E6 11 A2 01 14 00 18 00 F5 19
DS:0030 A2 01 14 00 18 00 F5 19 FF FF FF FF 00 00 00 00
DS:0038 FF FF FF FF 00 00 00 00 05 00 00 00 00 00 00 00
DS:0040 05 00 00 00 00 00 00 00 00 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