National University of Computer and Emerging Sciences 

**Laboratory Manual**

*for*

**Computer Organization and Assembly Language Programming**

| Course Instructor | Aleena Ahmad |
| --- | --- |
| Lab Instructor | Sana Ejaz |
| Semester | Fall 2024 |

Department of Computer Science

FAST-NU, Lahore, Pakistan

Page 1

**OBJECTIVES:**

∙ How to pass parameters through stack.

∙ How to use base pointer (BP) to access stack variables.

∙ How to implement a nested function call in another function.

**Instructions:**

| **1. Submit work in a single Word file with screenshots of meaningful results. 2. Do not submit asm, lst , or com files.**  **3. Press F2 if you want to step over the function Call. F1 will step into the function. (Do not submit a zip folder)** |
| --- |

**Task 1: Implement a function Subtract(int a, int b, int c, int d) that performs following operation globalVar = a – b – c – d ;**

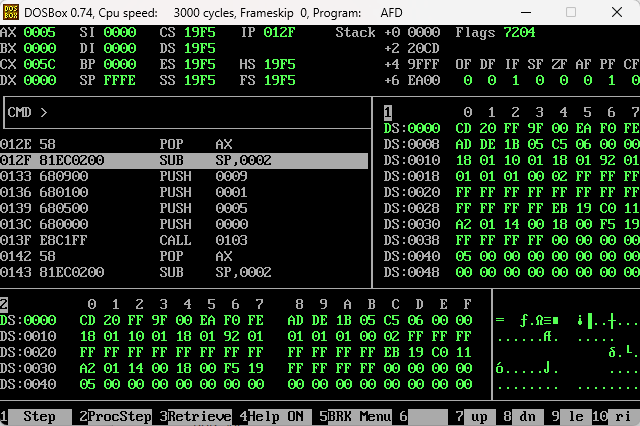
Run your code three times (Generic code to run for any test case) with three different parameters and verify the answer on AFD. Do not hard code 3 separate cases.

You have to pass parameters through stack and read them using BP. Properly clear the stack while returning from the function.

1. 0xA, 0x1, 0x2, 0x2 (Answer should be 10-1-2-2 = 5)

2. 0x9, 0x1, 0x5, 0x0 (Ans = 9-1-5-0 = 3)

3. 0xF, 0x1, 0x8, 0x4 (Ans = 15-1-8-4 = 2)



| [org 0x0100]  JMP start  subtract:  PUSH BP  MOV BP, SP  PUSH AX  MOV AX, [BP + 10]  SUB AX, [BP + 8]  SUB AX, [BP + 6]  SUB AX, [BP + 4]  MOV [BP + 12], AX  POP AX  POP BP  RET 8  start:  SUB SP, 2  ; Function Call 1  PUSH word 0xA  PUSH word 0x1  PUSH word 0x2  PUSH word 0x2  CALL subtract  POP AX  SUB SP, 2  ; Function Call 2  PUSH word 0x9  PUSH word 0x1  PUSH word 0x5  PUSH word 0x0  CALL subtract  POP AX  SUB SP, 2  ; Function Call 3  PUSH word 0xF  PUSH word 0x1  PUSH word 0x8  PUSH word 0x4  CALL subtract  POP AX  MOV AX, 0x4C00  INT 0x21 |
| --- |

**Task 2: Implement a function AnotherFunction(int a, int b, int c, int d) that performs following operation**

**int AnotherFunction(int a, int b, int c, int d)**

**{**

**localVar1 = Subtract(a,b,c,d) ; This is nested function call. Update Subtract such that it returns the result.**

**localVar2 = Subtract(c,d,0,0) ; pass zero as last two parameters.**

**return localVar1 + localVar2;**

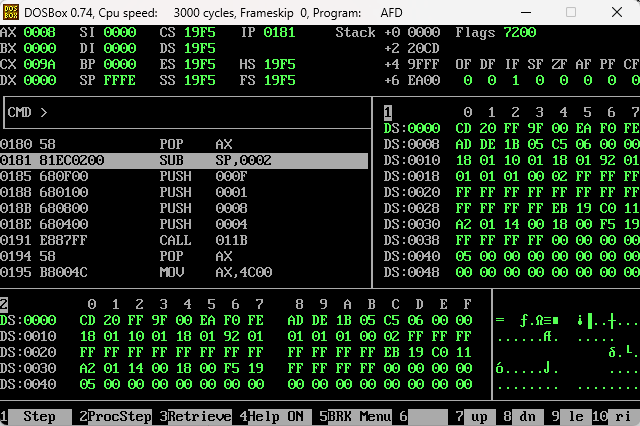
**}**

Run your code three times with three different parameters(Generic code) and verify the answer on AFD. Properly apply all the concepts and watch nested activation records. Save and Restore the registers (state) properly.

1. 0xA, 0x1, 0x2, 0x2 (Answer should be (10-1-2-2)+ (2-2-0-0) = 5+0 = 5)

2. 0x9, 0x1, 0x5, 0x0 (Ans = (9-1-5-0)+(5-0-0-0) = 3+5 = 8)

3. 0xF, 0x1, 0x8, 0x4 (Ans = (15-1-8-4)+(8-4-0-0) = 2+4 = 6)



| [org 0x0100]  JMP start  subtract:  PUSH BP  MOV BP, SP  PUSH AX  ; globalVar = a - b - c - d  MOV AX, [BP + 10]  SUB AX, [BP + 8]  SUB AX, [BP + 6]  SUB AX, [BP + 4]  MOV [BP + 12], AX  POP AX  POP BP  RET 8  anotherFunction:  PUSH BP  MOV BP, SP  SUB SP, 2  PUSH AX  PUSH BX  ; localVar1 = a - b - c - d  SUB SP, 2  PUSH word [BP + 10]  PUSH word [BP + 8]  PUSH word [BP + 6]  PUSH word [BP + 4]  CALL subtract  POP AX  ; localVar2 = c - d - 0 - 0  SUB SP, 2  PUSH word [BP + 6]  PUSH word [BP + 4]  PUSH word 0x0  PUSH word 0x0  CALL subtract  POP BX  ADD AX, BX  MOV [BP + 12], AX  POP BX  POP AX  MOV SP, BP  POP BP  RET 8  start:  SUB SP, 2  ; Function Call 1  PUSH word 0xA  PUSH word 0x1  PUSH word 0x2  PUSH word 0x2  CALL anotherFunction  POP AX  SUB SP, 2  ; Function Call 2  PUSH word 0x9  PUSH word 0x1  PUSH word 0x5  PUSH word 0x0  CALL anotherFunction  POP AX  SUB SP, 2  ; Function Call 3  PUSH word 0xF  PUSH word 0x1  PUSH word 0x8  PUSH word 0x4  CALL anotherFunction  POP AX  MOV AX, 0x4C00  INT 0x21 |
| --- |