CSUS, Computer Science Department, CSC35, Spring 2021, Dr. Ghansah

Homework 4 (PROCEDURES, CONDITIONAL BRANCH, LOGICAL INSTRUCTIONS)

Note: Submission of this Homework Assignment is optional. It will not be graded but if you do it, you might be better prepared for exams. I will provide solution to this homework. All exams will be multiple choice on Canvas so you should know your material very well.

The following problems are from your Irvine Textbook Ed 7

Do the following problems from your text Irvine Ed 7

Q1 Sec 5.8.1 Do problems 1,5,9,13,17,19

Q2 Sec 6.10.1 Do problems 1, 7, 13

Q3. Sec 6.10.2 Do problems 4, 7

SUBMISSION (OPTIONAL): Submit Electronically via *Canvas*. FileName must be according to the format specified in the course syllabus.

SOLUTIONS

5.8.1

1. PUSHAD

5. The stack starts from a high address and expands to a lower address so pushing decrements the value of the stack pointer for 32 bit register we have 4 bytes.

```
SUB esp, 4
MOV [esp], eax
```

9. False

13. False

17. C, Before the Ex3Sub eax has 30. In the procedure call all the 16 bit registers are pushed from the stack. This includes ax which is the lowest 16 bits of eac (which contains the value 30 and the highest 16 bits are all zero). When 80 is moved to eax the highest 16 bits will still be zero and the lowest 16 bits will contain 80. After the Proc ends all the 16 bit registers are popped back and az will contain the 30 that was popped and maintain the zeros in the highest 16 bits.

19. A, Initially 0 is moved into edx and 40 into eax. Then only eax is pushed onto the stack. When Ex5Sub is called its return address is pushed on the stack. When we pop eax it holds the return address back to main. The next pop into edx puts the value 40 into edx. Eax is pushed back on the stack so the stack now has the return address again and when the procedure returns it has the correct address to return so we will have edx with a value of 40 and no runtime errors. The table below shows the stack after key instructions execute

Executed Instruction	3: push eax	4: call Ex5Sub	10: pop eax	11: pop edx	12: push eax	13: ret (pops from stack into esp)
Stack	40	40	40		Return address	
		Return address				
eax	40	40	Return address	Return address	Return address	Return address
edx	0	0	0	40	40	40

6.10.1

1. after mov bx will have 1111 1111 1111 1111 in binary.

1111 1111 1111 1111

0000 0000 0110 1011 (6bh in hex)

Result from And: 0000 0000 0110 1011 (6bh)

7. a) 0010 1101b (2Dh)

b) 6dh = 0110 1101

4ah = 0100 1010

And Result: 0100 1000b (48h)

c) 0000 1111

61h = 0110 0001

Or Result: 0110 1111b (6Fh)

d) 94h = 1001 0100

37h = 0011 0111

Xor Result: 1010 0011 (0A3h)

13. 1

6.10.2

4. CMP dx, cx

JBE L1

7. CMP val1, ecx

JLE: ELSE

CMP ecx, edx

JLE: Else

MOV X, 1

```
    JMP END
ELSE:
    MOV X, 2
END:
```