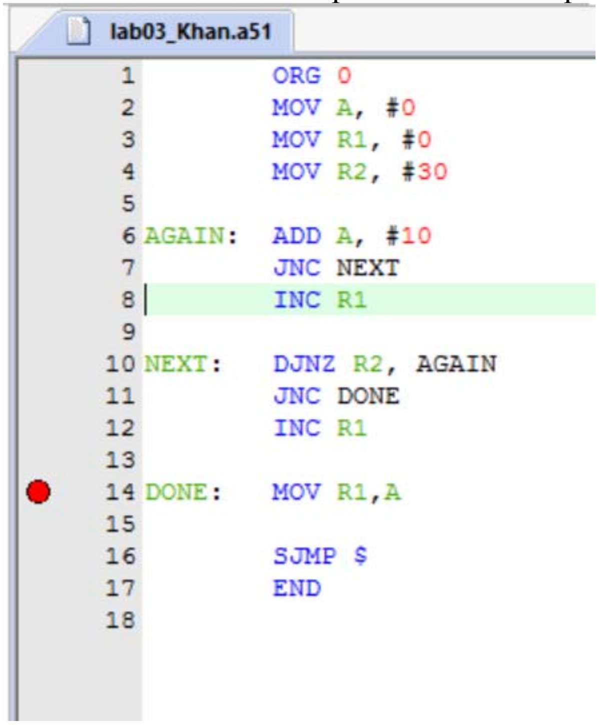
CECS 262

LAB 3
EXAMINING THE LOOP & JUMP INSTRUCTIONS

Student Name: Umar Khan
Student ID: 014477331

Tasks:

1. Write and assemble a program to add number 10 thirty times (sum=300/012CH). Put the sum in registers R0 (low byte) and R1 (high byte). Screen-shot the final results in R0 and R1. Loop instruction is required for this task.



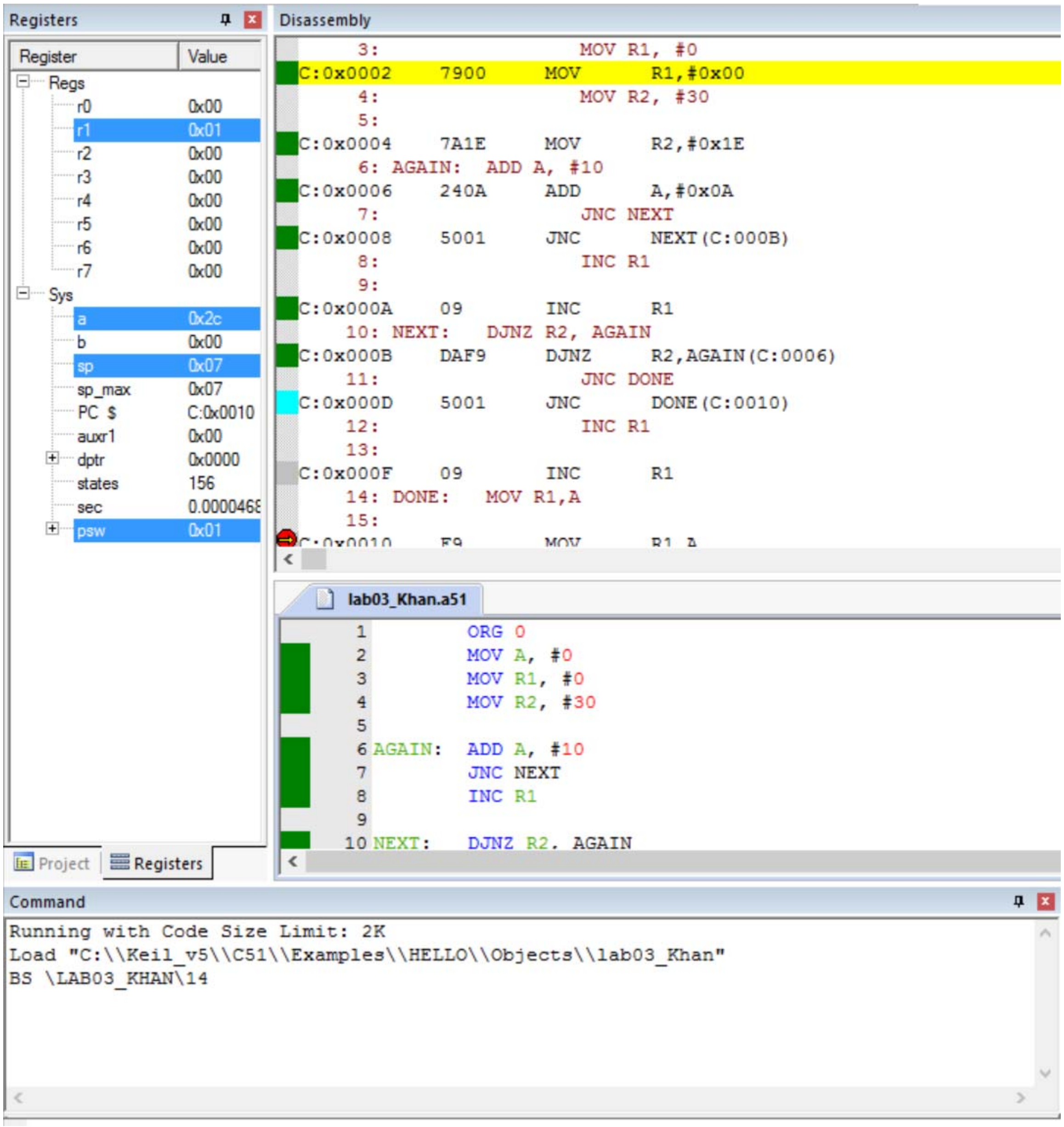
```
        ORG 0
        MOV A, #0
        MOV R1, #0
        MOV R2, #30

AGAIN:  ADD A, #10
        JNC NEXT
        INC R1

NEXT:   DJNZ R2, AGAIN
        JNC DONE
        INC R1

DONE:   MOV R1,A

        SJMP $
        END
```



Register	Value
Regs	
r0	0x00
r1	0x01
r2	0x00
r3	0x00
r4	0x00
r5	0x00
r6	0x00
r7	0x00
Sys	
a	0x2c
b	0x00
sp	0x07
sp_max	0x07
PC $	C:0x0010
auxr1	0x00
dptr	0x0000
states	156
sec	0.0000468
psw	0x01

```
     3:                MOV R1, #0
C:0x0002    7900     MOV      R1,#0x00
     4:                MOV R2, #30
     5:
C:0x0004    7A1E     MOV      R2,#0x1E
     6: AGAIN:  ADD A, #10
C:0x0006    240A     ADD      A,#0x0A
     7:                JNC NEXT
C:0x0008    5001     JNC      NEXT(C:000B)
     8:                INC R1
     9:
C:0x000A    09       INC      R1
    10: NEXT:    DJNZ R2, AGAIN
C:0x000B    DAF9     DJNZ     R2,AGAIN(C:0006)
    11:                JNC DONE
C:0x000D    5001     JNC      DONE(C:0010)
    12:                INC R1
    13:
C:0x000F    09       INC      R1
    14: DONE:    MOV R1,A
    15:
C:0x0010    F9       MOV      R1,A
```

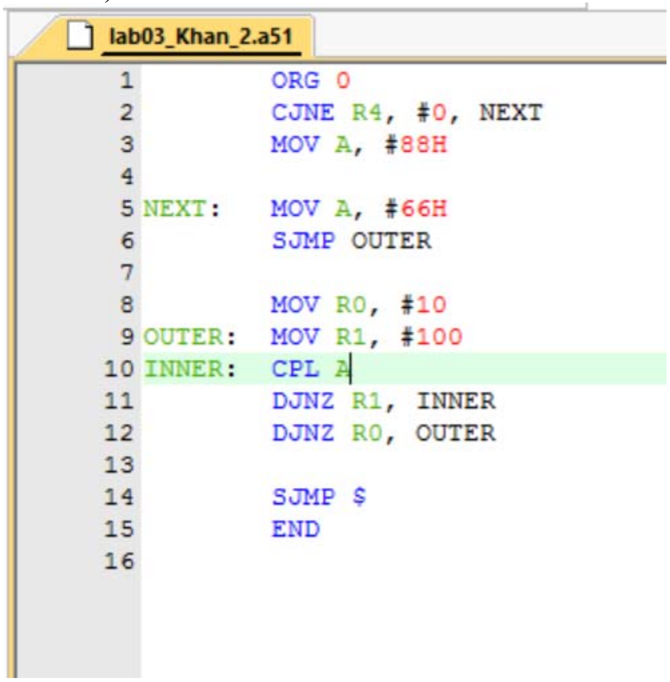
```
          ORG 0
          MOV A, #0
          MOV R1, #0
          MOV R2, #30

AGAIN:    ADD A, #10
          JNC NEXT
          INC R1

NEXT:     DJNZ R2, AGAIN
```

```
Running with Code Size Limit: 2K
Load "C:\\Keil_v5\\C51\\Examples\\HELLO\\Objects\\lab03_Khan"
BS \LAB03_KHAN\14
```

2. Write and assemble a program to determine if R4 contains the value 0. If so, load the value 66H to the accumulator; otherwise, load value 88H to the accumulator. Then use nested loop to complement A (use instruction CPL A) 1000 times. Screen-shot the final result in A.(make sure the value of R4 shows in your screen-shot)



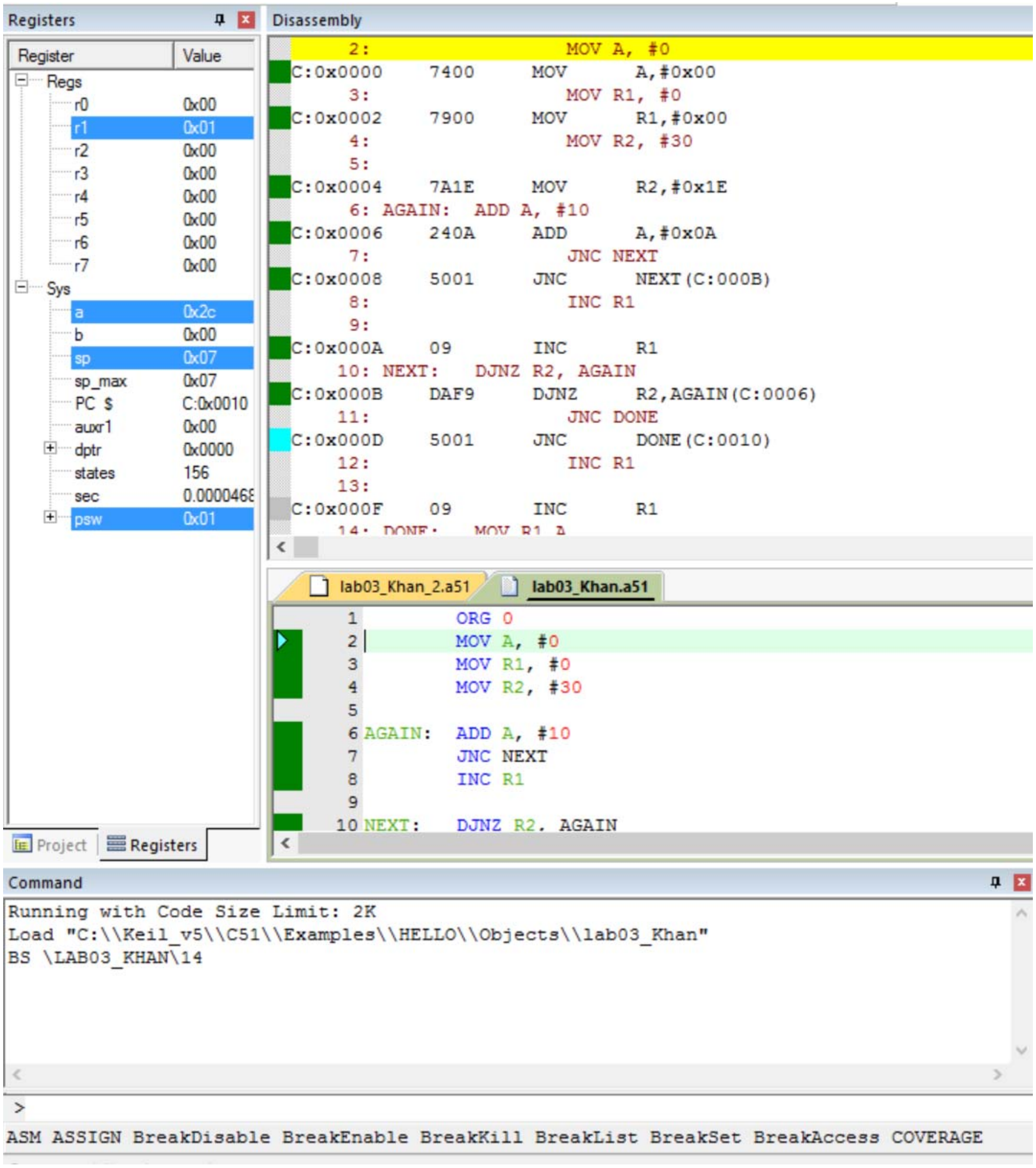
```
        ORG 0
        CJNE R4, #0, NEXT
        MOV A, #88H

NEXT:   MOV A, #66H
        SJMP OUTER

        MOV R0, #10
OUTER:  MOV R1, #100
INNER:  CPL A
        DJNZ R1, INNER
        DJNZ R0, OUTER

        SJMP $
        END

```



Register	Value
Regs	
r0	0x00
r1	0x01
r2	0x00
r3	0x00
r4	0x00
r5	0x00
r6	0x00
r7	0x00
Sys	
a	0x2c
b	0x00
sp	0x07
sp_max	0x07
PC $	C:0x0010
auxr1	0x00
dptr	0x0000
states	156
sec	0.0000468
psw	0x01

```
     2:                    MOV A, #0
C:0x0000    7400      MOV      A,#0x00
     3:                    MOV R1, #0
C:0x0002    7900      MOV      R1,#0x00
     4:                    MOV R2, #30
     5:
C:0x0004    7A1E      MOV      R2,#0x1E
     6: AGAIN:  ADD A, #10
C:0x0006    240A      ADD      A,#0x0A
     7:                    JNC NEXT
C:0x0008    5001      JNC      NEXT(C:000B)
     8:                    INC R1
     9:
C:0x000A    09        INC      R1
    10: NEXT:    DJNZ R2, AGAIN
C:0x000B    DAF9      DJNZ     R2,AGAIN(C:0006)
    11:                    JNC DONE
C:0x000D    5001      JNC      DONE(C:0010)
    12:                    INC R1
    13:
C:0x000F    09        INC      R1
    14: DONE:    MOV R1, A
```

```
        ORG 0
        MOV A, #0
        MOV R1, #0
        MOV R2, #30

AGAIN:  ADD A, #10
        JNC NEXT
        INC R1

NEXT:   DJNZ R2, AGAIN
```

```
Running with Code Size Limit: 2K
Load "C:\\Keil_v5\\C51\\Examples\\HELLO\\Objects\\lab03_Khan"
BS \LAB03_KHAN\14
```

```
>
ASM ASSIGN BreakDisable BreakEnable BreakKill BreakList BreakSet BreakAccess COVERAGE
```