VIT®
Vellore Institute of Technology
(Deemed to be University under section 3 of UGC Act, 1956)

SCHOOL OF COMPUTER SCIENCE ENGINEERING

WINTER SEMESTER 2022-2023

LAB ASSIGNMENT - 1

Slot: L11 – L12

Class: VL2022230504038

Programme Name & Branch: B. Tech CSE

Course code & Title: BECE204P – Microprocessors and Microcontrollers Lab

Faculty Name: Venu Allapakam

Task 1: Arithmetic Operations – Addition and Subtraction

Program 1: Addition

Aim: To perform arithmetic addition- 8051 micro controllers

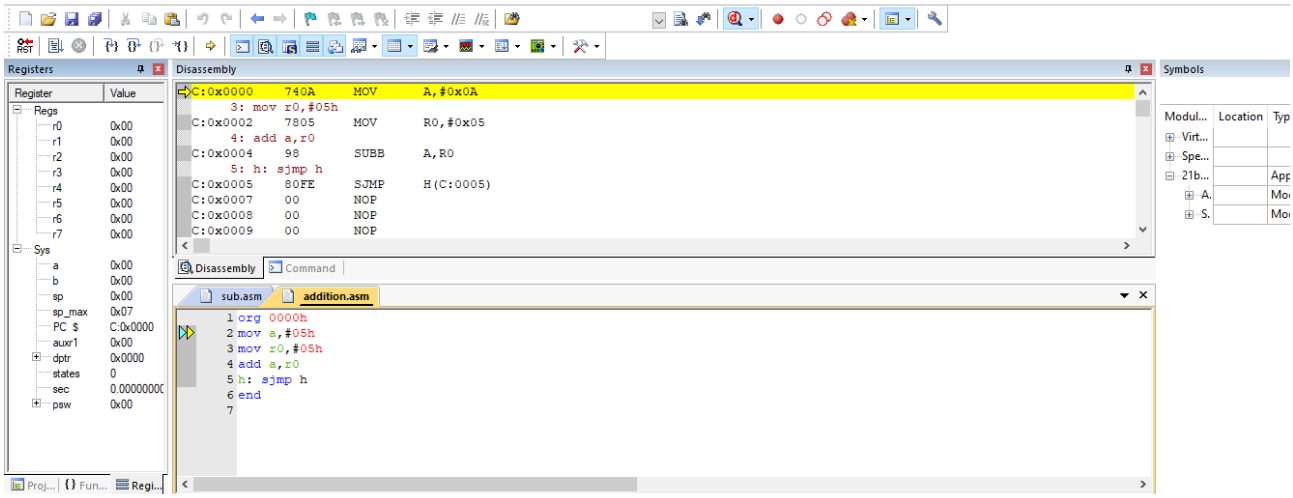
Software Requirement: Keil Software

Program:

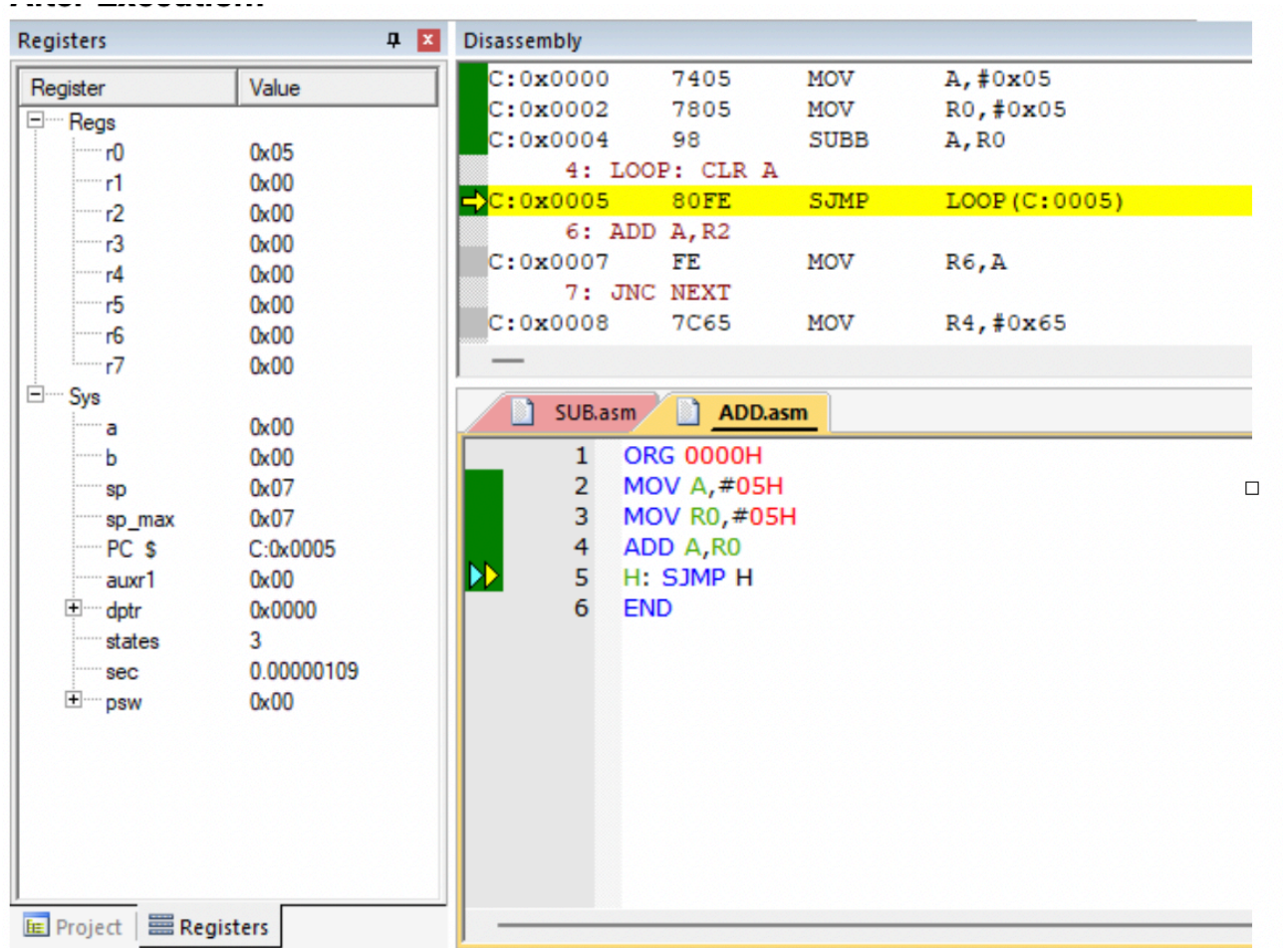
```
1  org 0000h
2  mov a,#05h
3  mov r0,#05h
4  add a,r0
5  h: sjmp h
6  end
7
```

Output:

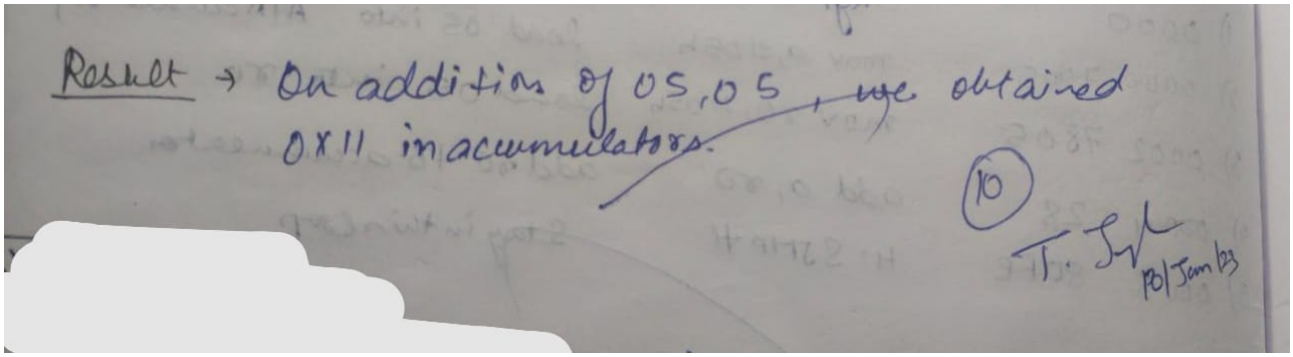
Before Execution Register status:



After Execution:



Result-



Program 2: Subtraction

Aim: To perform arithmetic subtraction- 8051 micro controllers

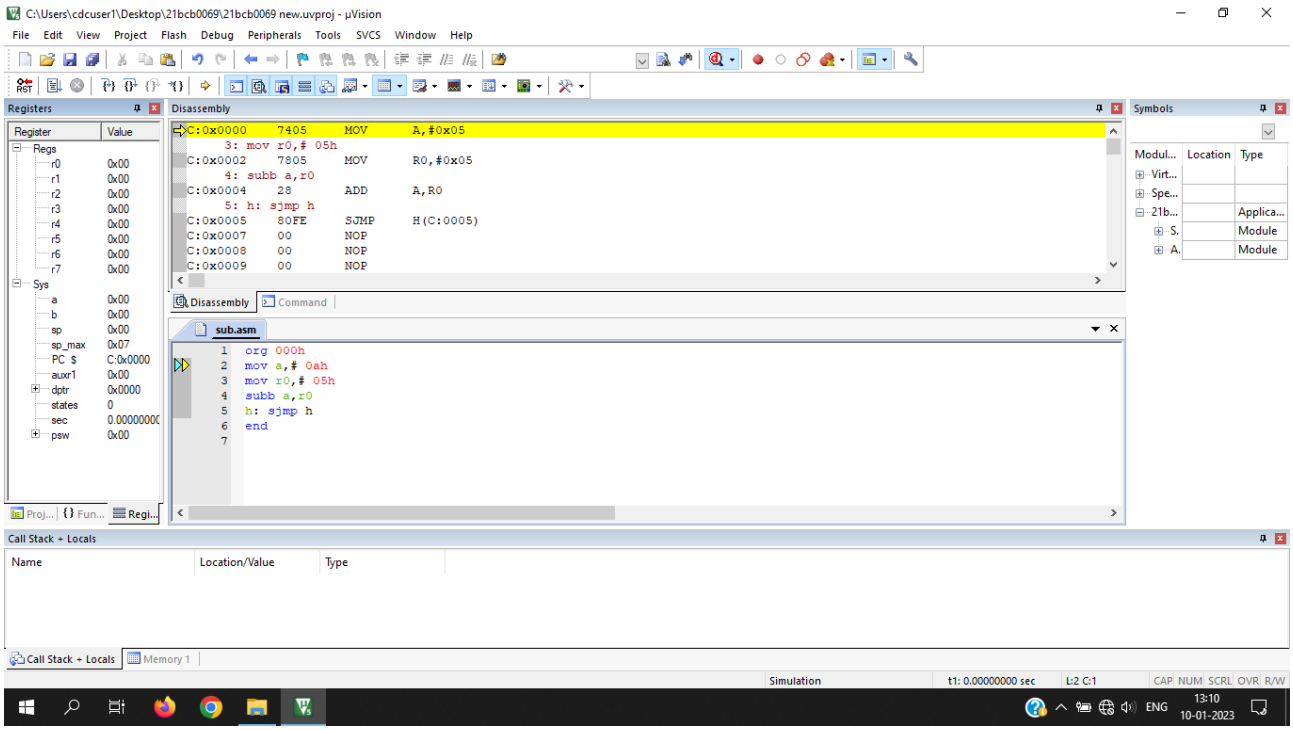
Software Requirement: Keil Software

Program:

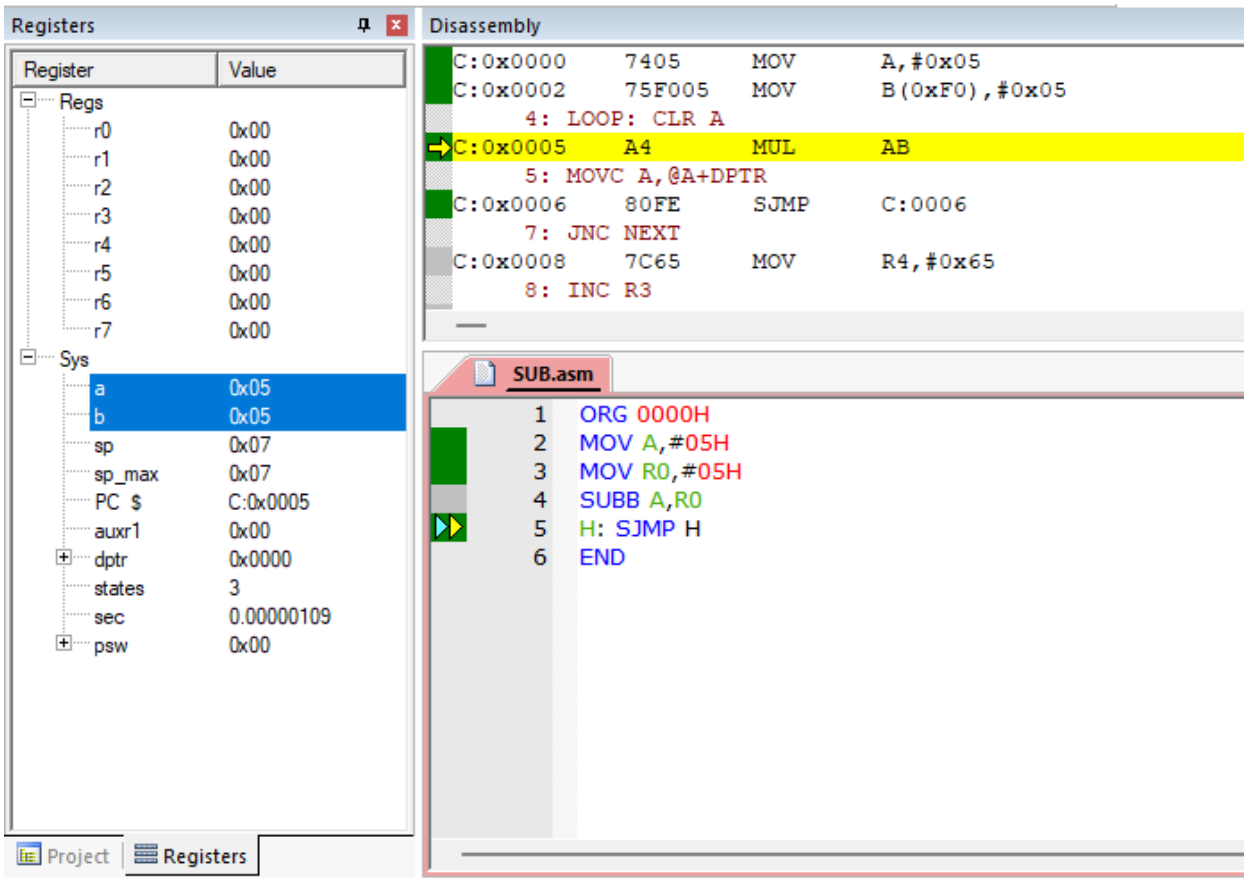
```
1  org 0000h
2  mov a,#0AH
3  mov r0,#05h
4  SUBB a,r0
5  h: sjmp h
6  end
```

Output:

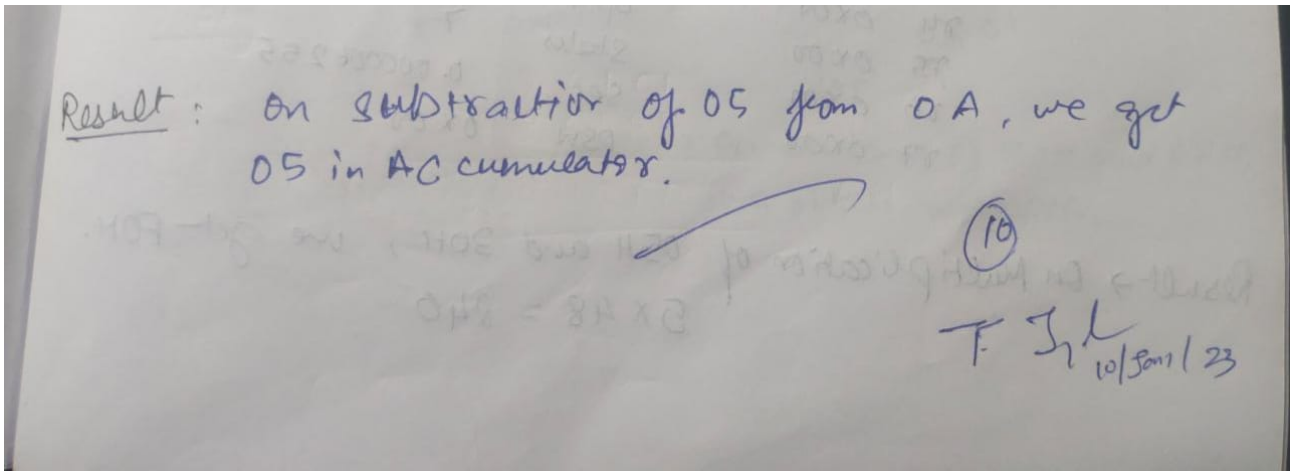
Before Execution Register status:



After Execution:



Result—



Task 2: Arithmetic Operations – Multiplication and Division

Program 1: Multiplication

Aim: To perform arithmetic multiplication- 8051 micro controllers

Software Requirement: Keil Software

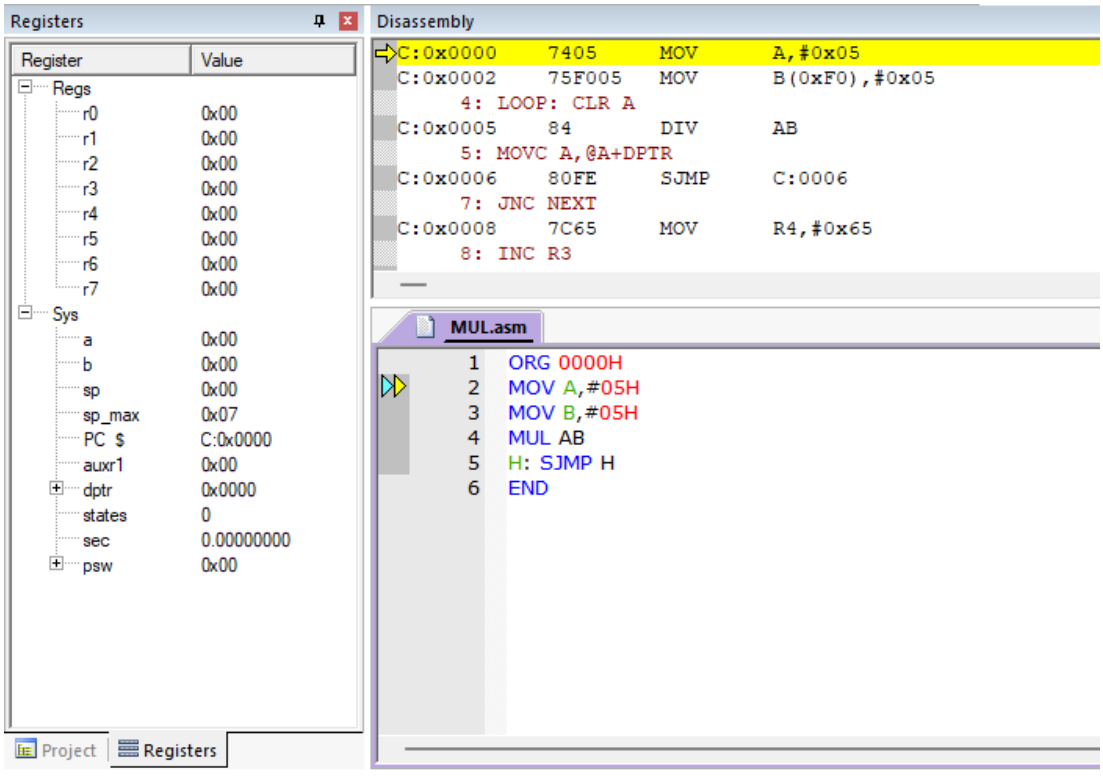
Program:

multi.asm

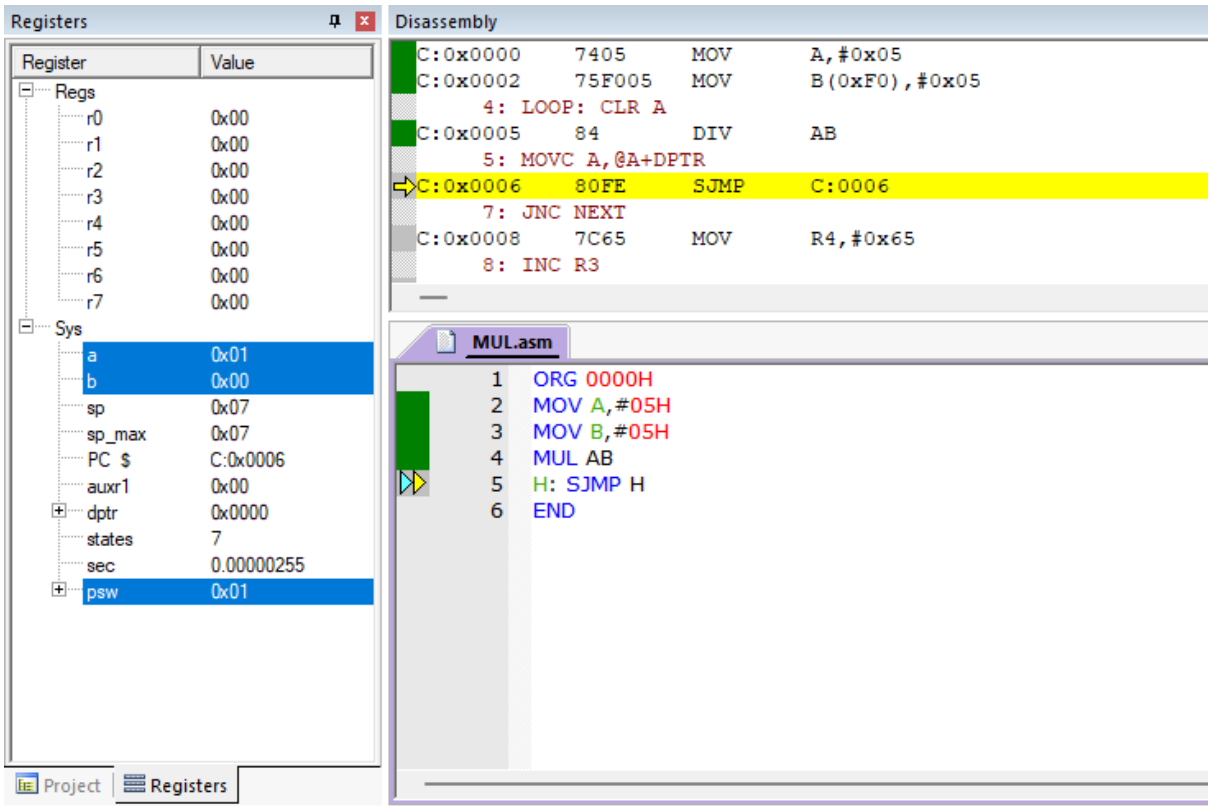
```
1 ORG 0000H
2 MOV A,#05H
3 MOV B,#30H
4 MUL AB
5 h: sjmp h
6 end
```

Output:

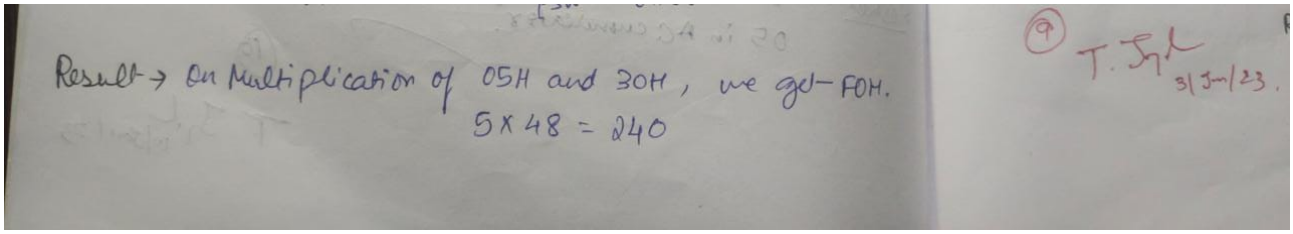
Before Execution Register status:



After Execution:



Result—



Program 2: Division

Aim: To perform arithmetic division- 8051 micro controllers

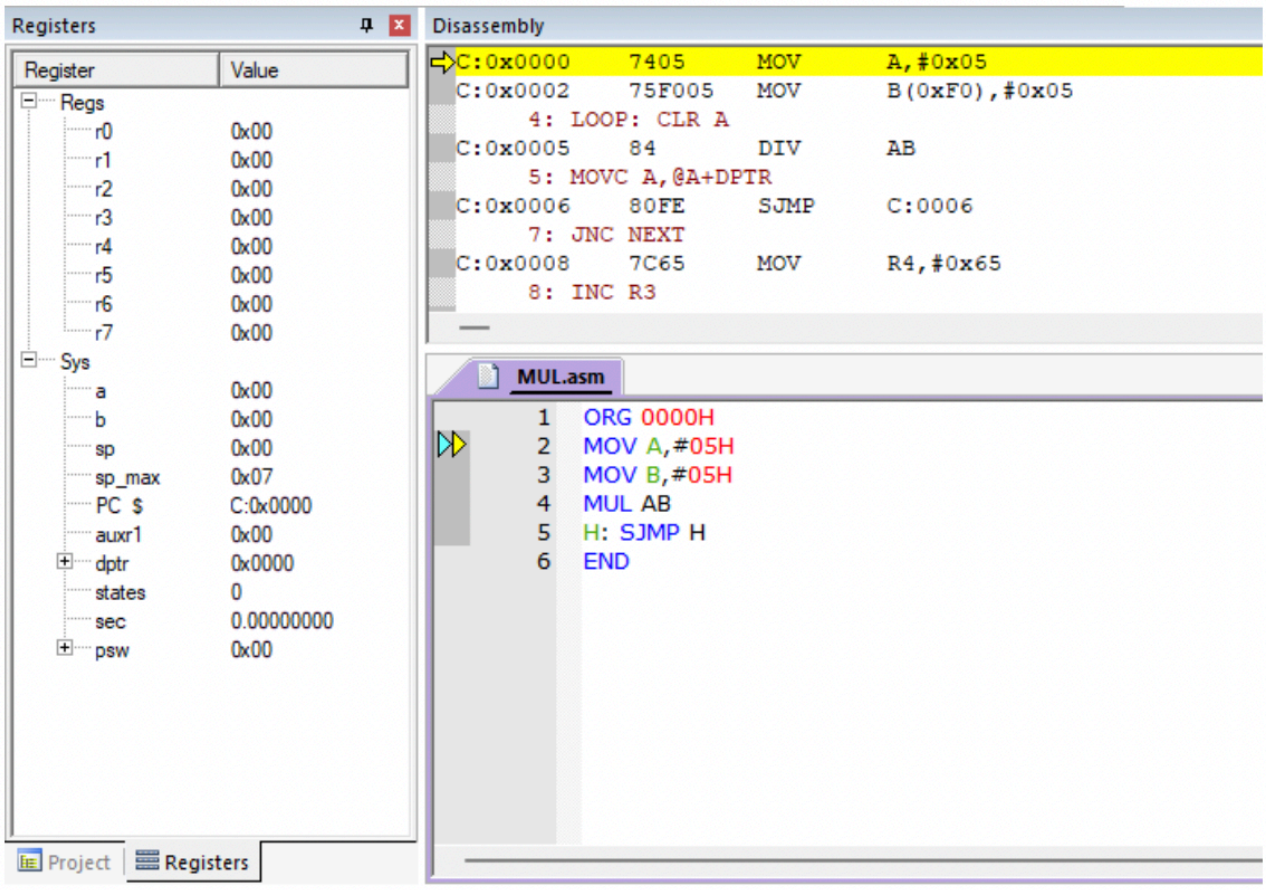
Software Requirement: Keil Software

Program:

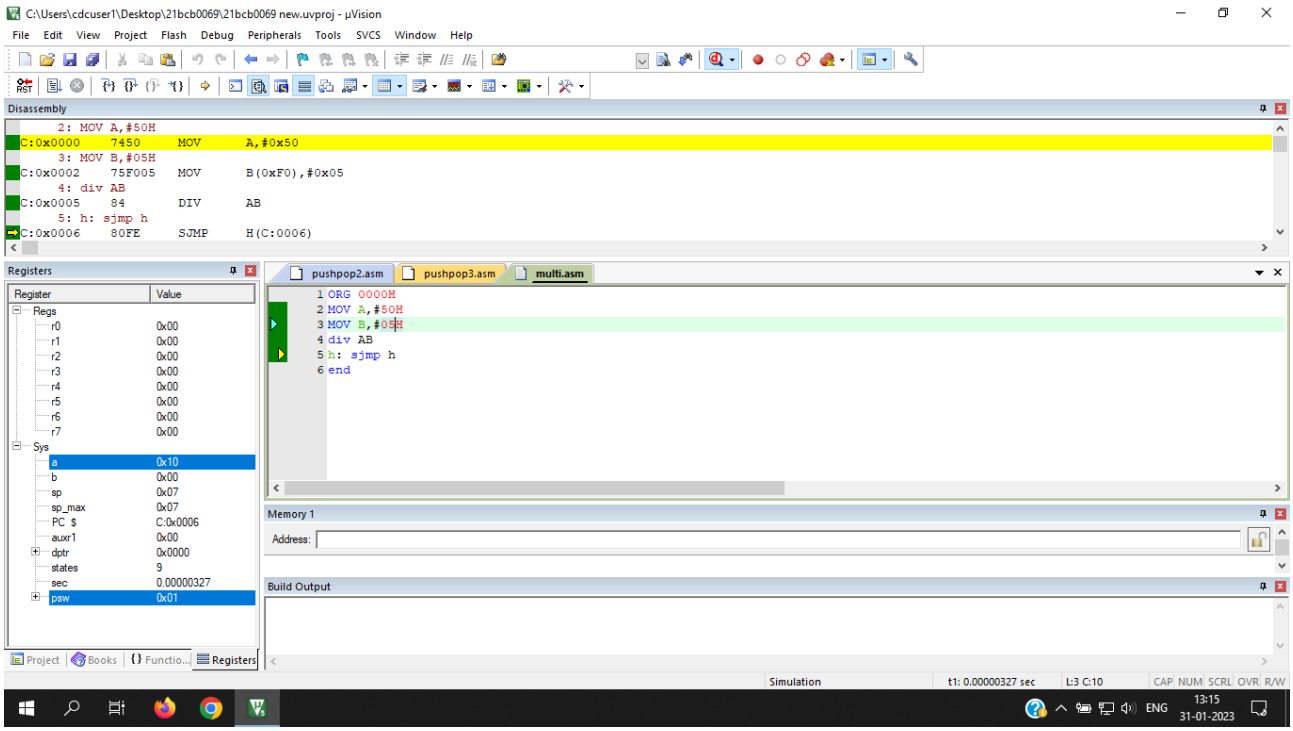
```
1 ORG 0000H
2 MOV A,#50H
3 MOV B,#05H
4 div AB
5 h: sjmp h
6 end
```

Output:

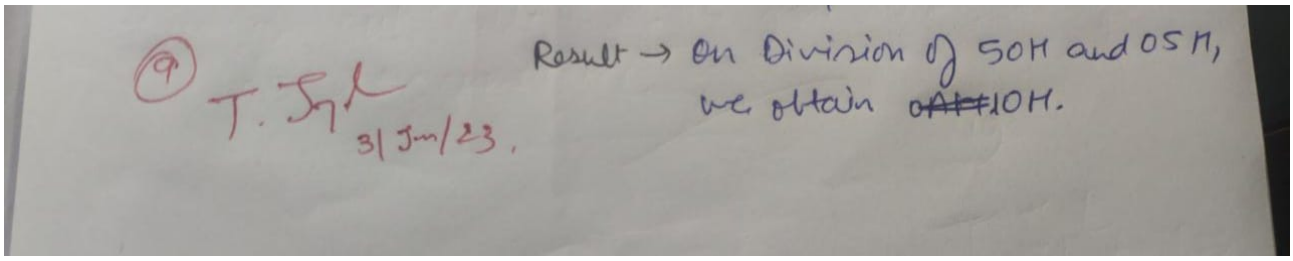
Before Execution Register status:



After Execution:



Result—



Task 3: Stack Operations - Push and Pop

Program 1: Push

Aim: To load values in each of the registers R0 to R4 and then push each of them into the stack

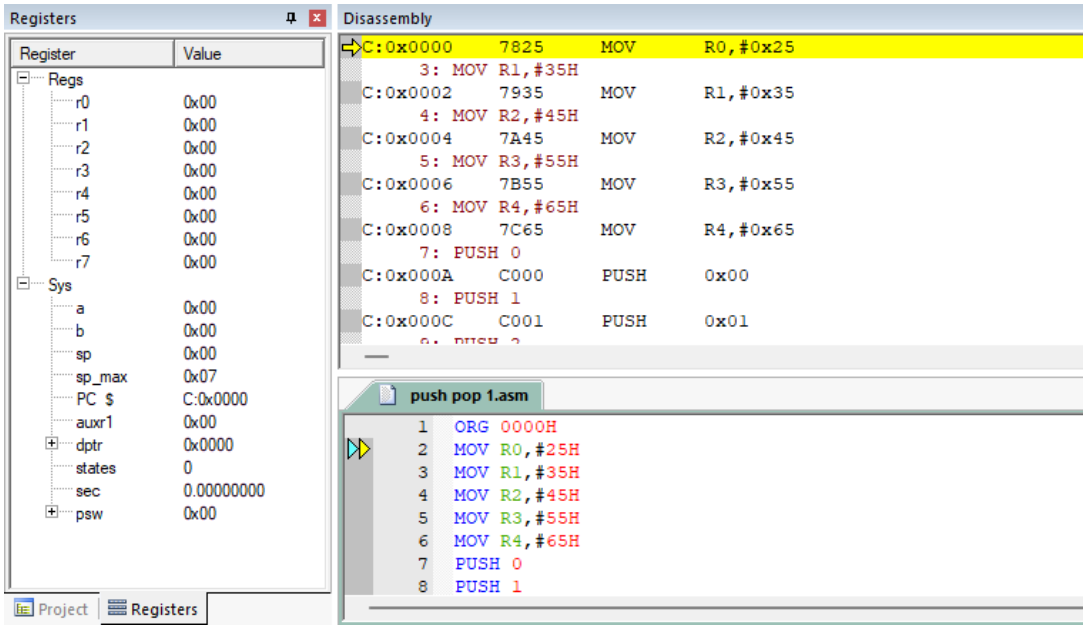
Software Requirement: Keil software

Program:

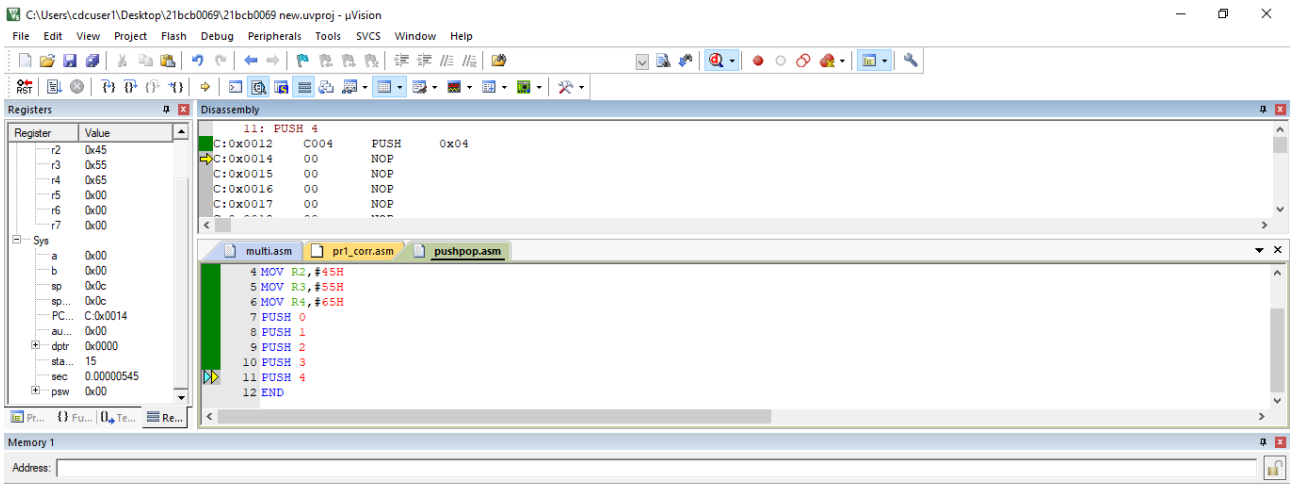
```
1   ORG 0000H
2   MOV R0,#25H
3   MOV R1,#35H
4   MOV R2,#45H
5   MOV R3,#55H
6   MOV R4,#65H
7   PUSH 0
8   PUSH 1
9   PUSH 2
10  PUSH 3
11  PUSH 4
12  END
```

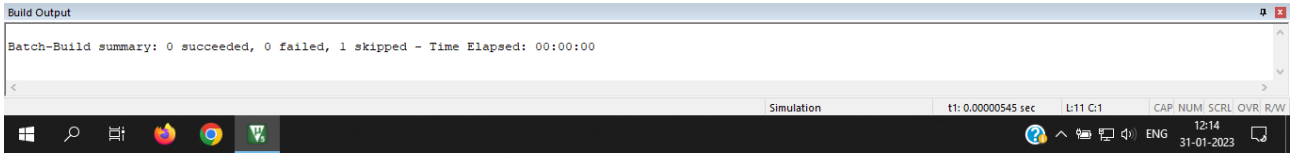
Output:

Before Execution:

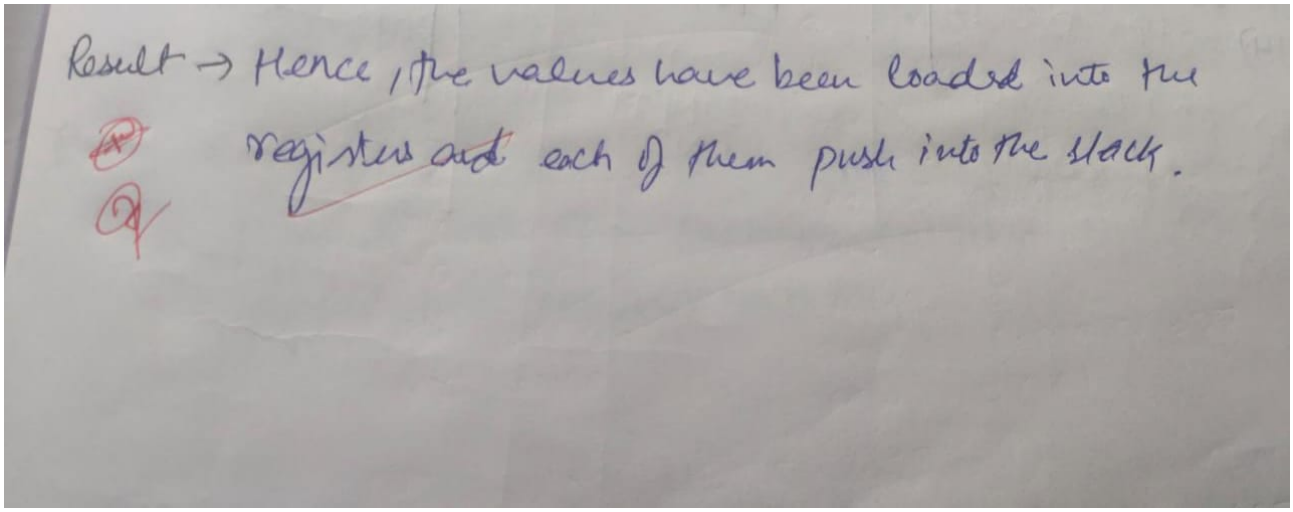


After Execution:





Result:



Program 2: Pop

Aim: To set SP to 0D and then put a different value in each of the RAM locations 08 to 0D and pop each location into registers R0 to R4

Software Requirement: Keil Software

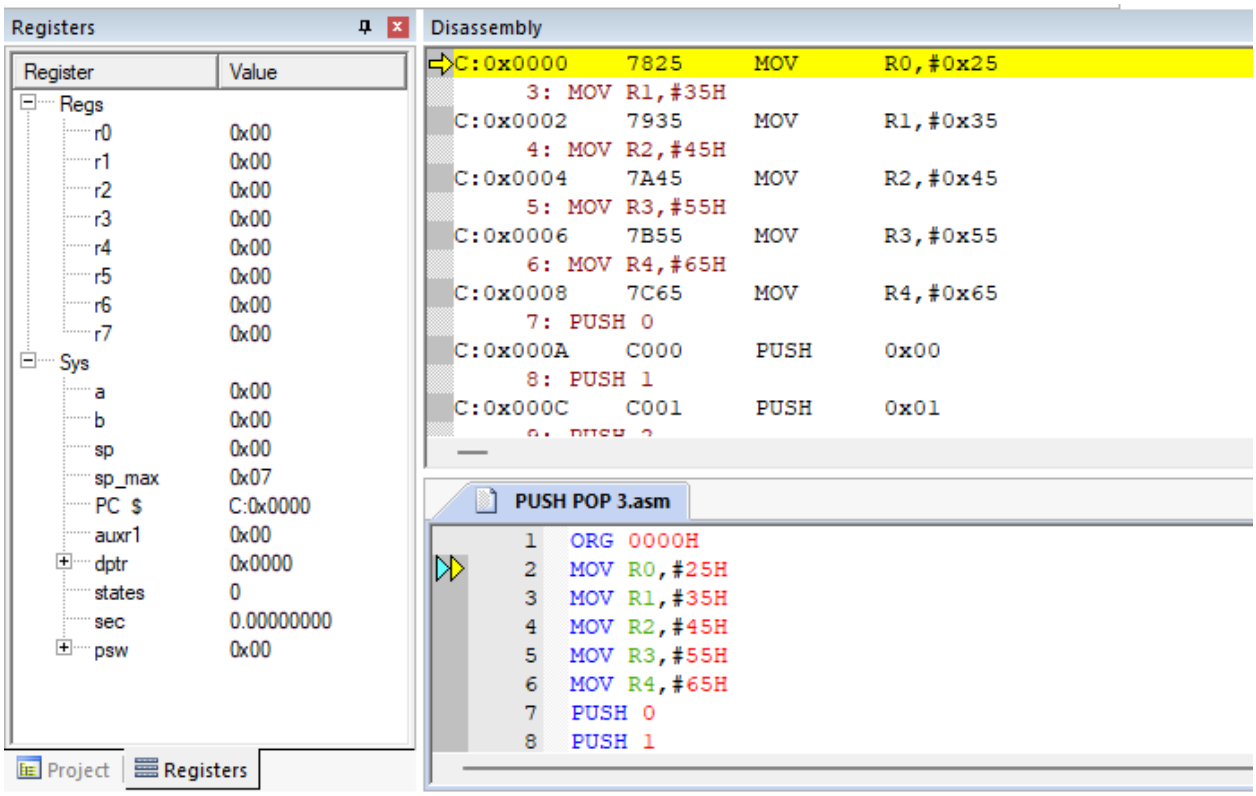
Program:

pushpop2.asm

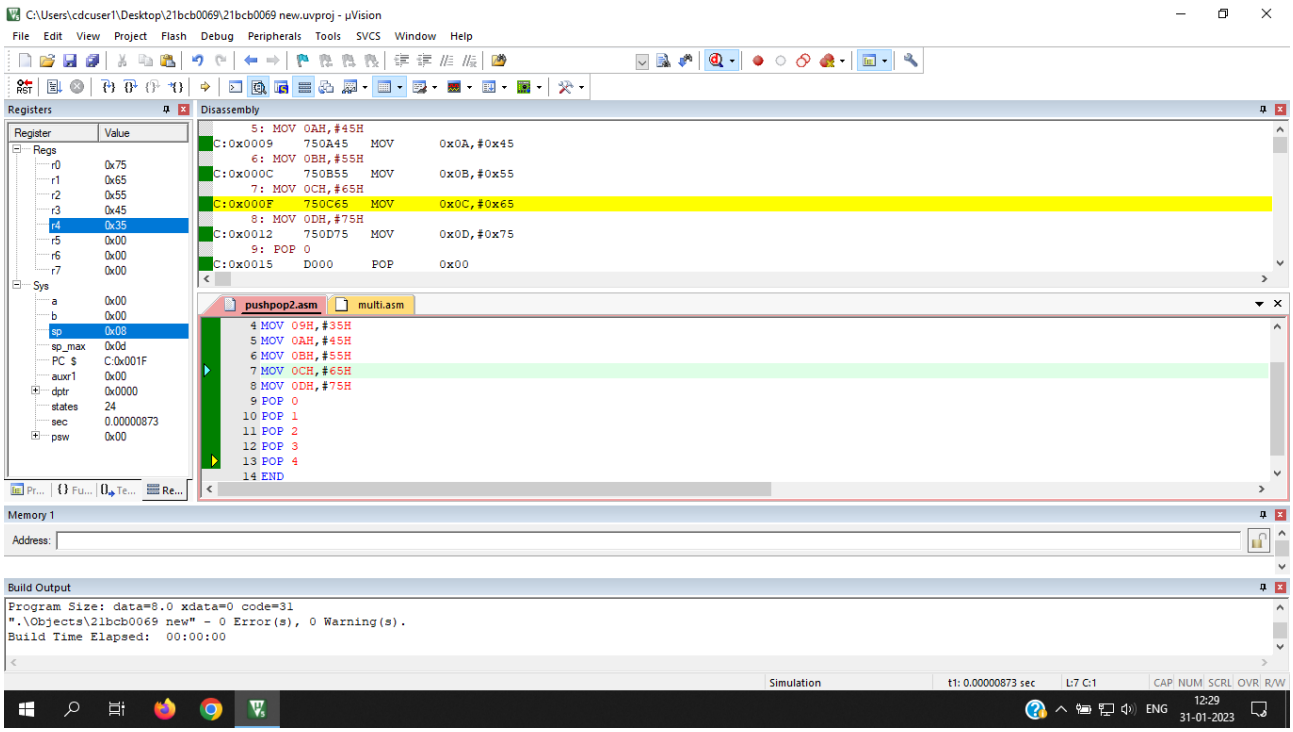
```
ORG 0000H
MOV SP,#0DH
MOV 08H,#25H
MOV 09H,#35H
MOV 0AH,#45H
MOV 0BH,#55H
MOV 0CH,#65H
MOV 0DH,#75H
POP 0
POP 1
POP 2
POP 3
POP 4
END
```

Output:

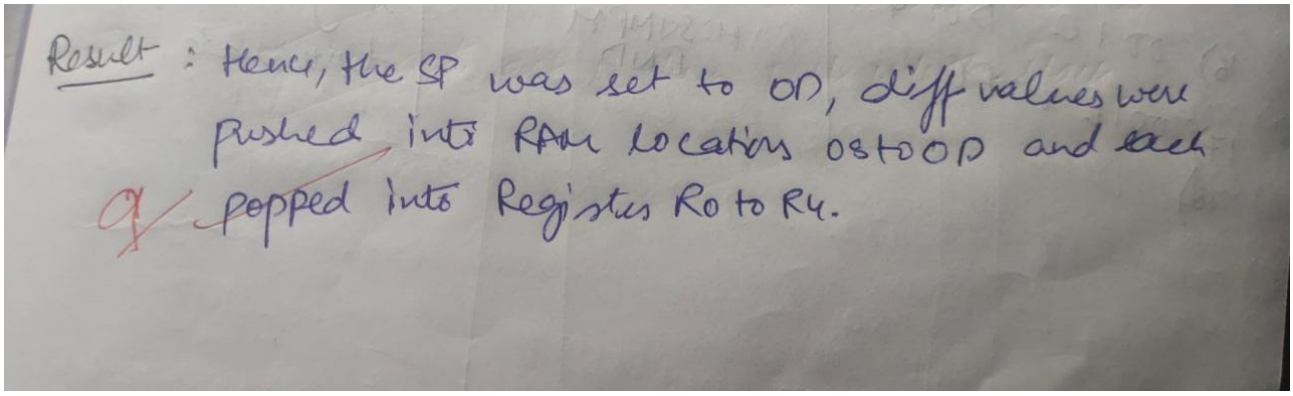
Before Execution:



After Execution:



Result:



Program 3: Push and Pop

Aim: To load values in each of the registers R0 to R4 and then push each of them into the stack and then pop them back

Software Requirement Keil Software

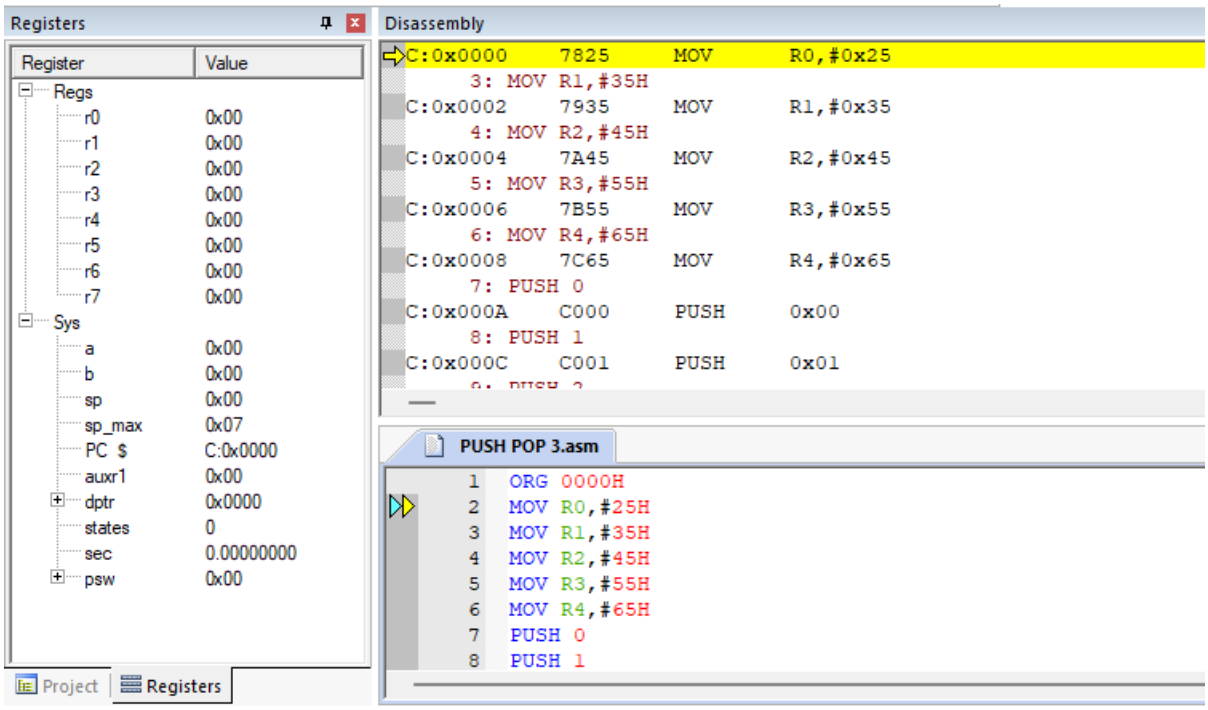
Program:

pushpop2.asm | pushpop3

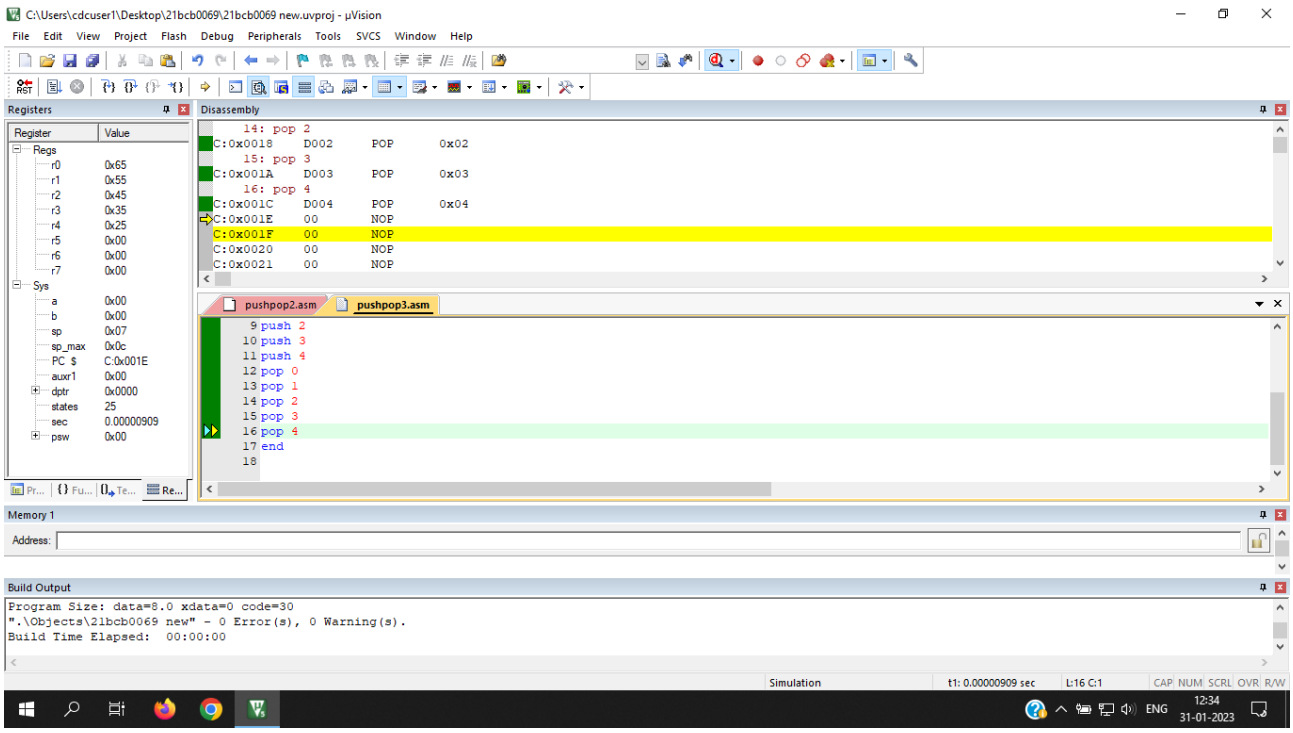
```
1  org 0000h
2  mov r0,#25h
3  mov r1,#35h
4  mov r2,#45h
5  mov r3,#55h
6  mov r4,#65h
7  push 0
8  push 1
9  push 2
10 push 3
11 push 4
12 pop 0
13 pop 1
14 pop 2
15 pop 3
16 pop 4
17 end
18
```

Output:

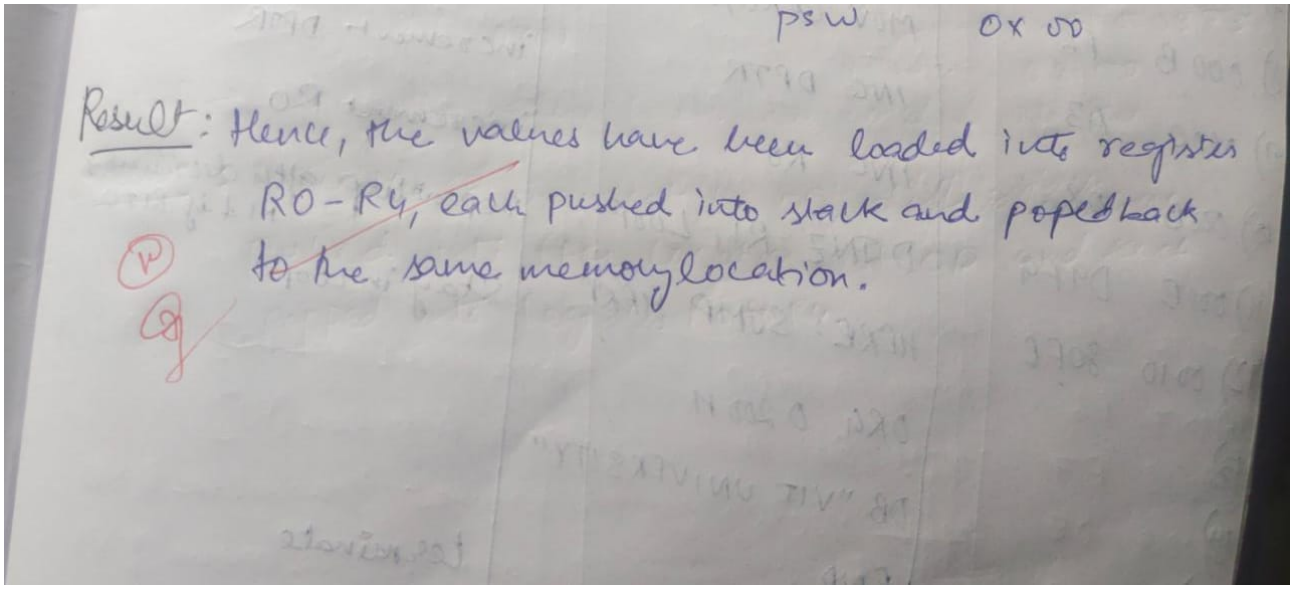
Before Execution:



After Execution:



Result:



Task 4: Data Transfer Operations

Program 1:

Aim: to transfer a string of data “VIT UNIVERSITY” from ROM locations starting at 200H to RAM locations starting at 40H

Software Requirement: Keil Software

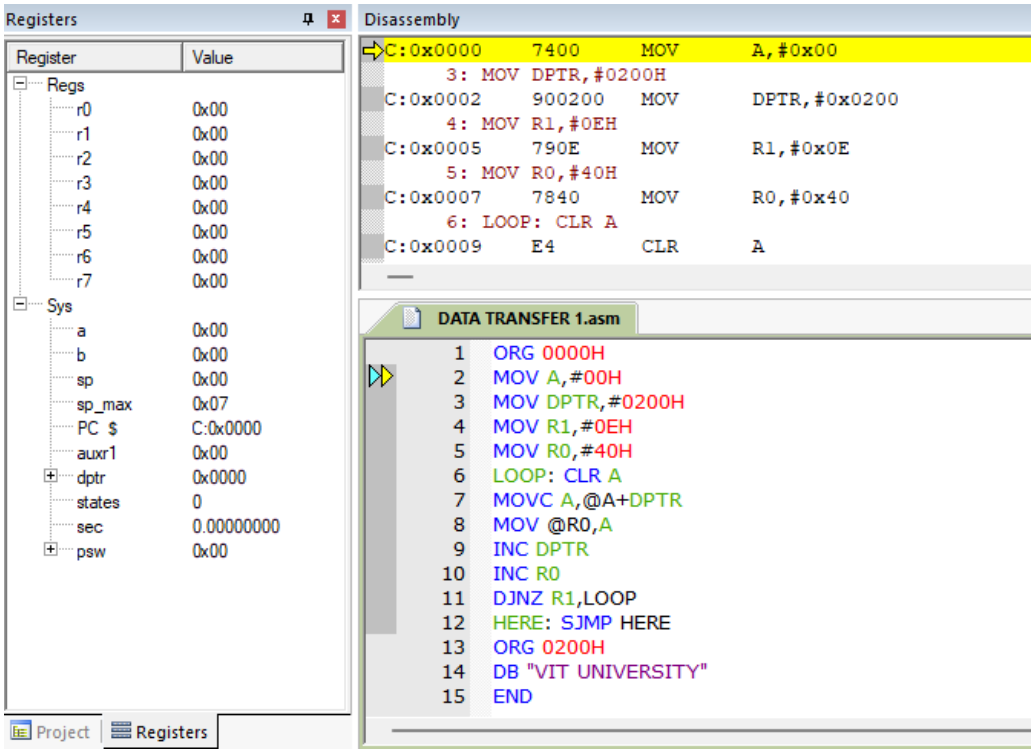
Program:

DATA TRANSFER 1.asm

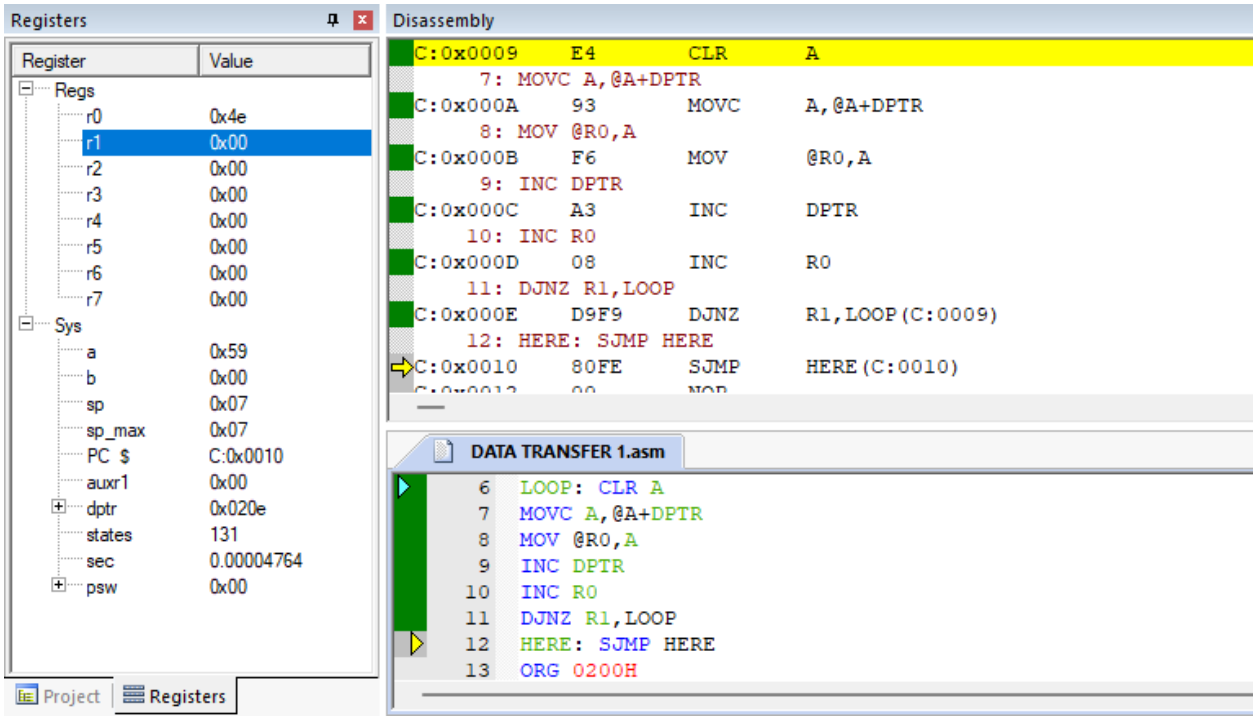
```
1   ORG 0000H
2   MOV A,#00H
3   MOV DPTR,#0200H
4   MOV R1,#0EH
5   MOV R0,#40H
6   LOOP: CLR A
7   MOVC A,@A+DPTR
8   MOV @R0,A
9   INC DPTR
10  INC R0
11  DJNZ R1,LOOP
12  HERE: SJMP HERE
13  ORG 0200H
14  DB "VIT UNIVERSITY"
15  END
```

Output:

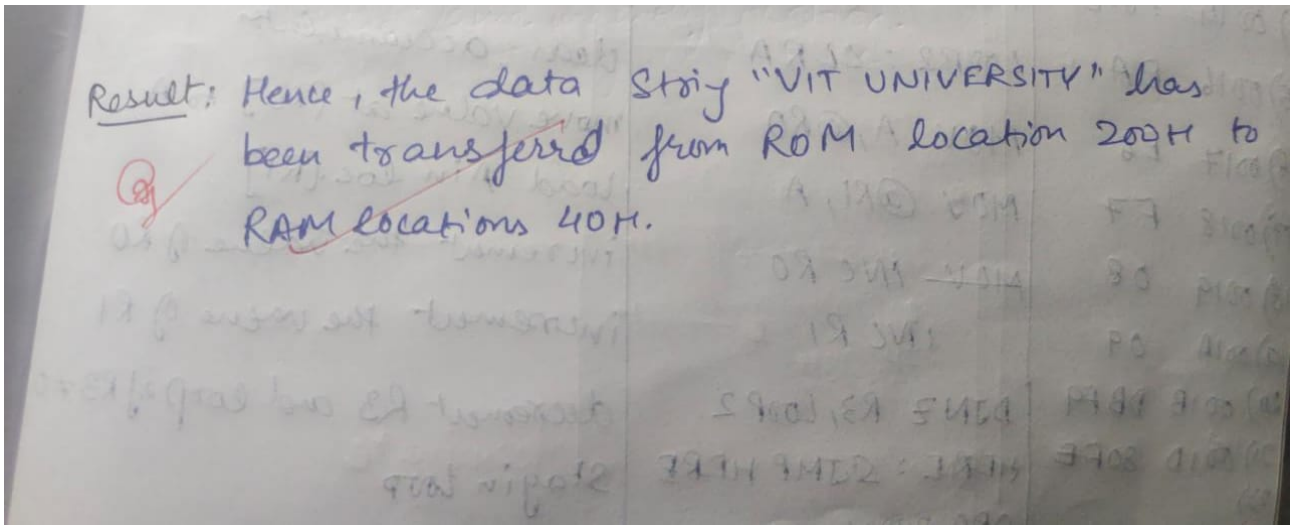
Before Execution:



After Execution:



Result:



Program 2:

Aim: To Transfer a string of data “VIT UNIVERSITY” from ROM locations starting at 200H to RAM locations starting at 40H and then move it to RAM locations starting at 60H

Software Requirement: Keil Software

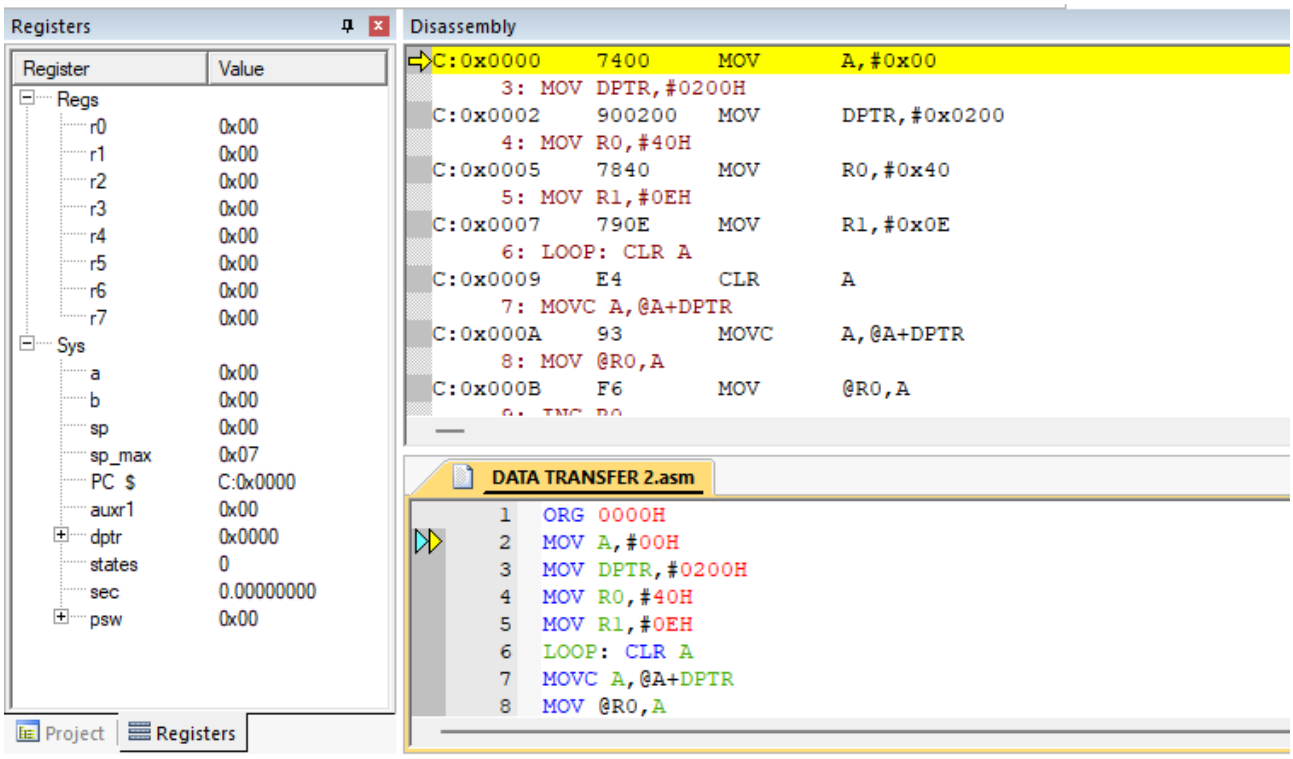
Program:

DATA TRANSFER 2.asm

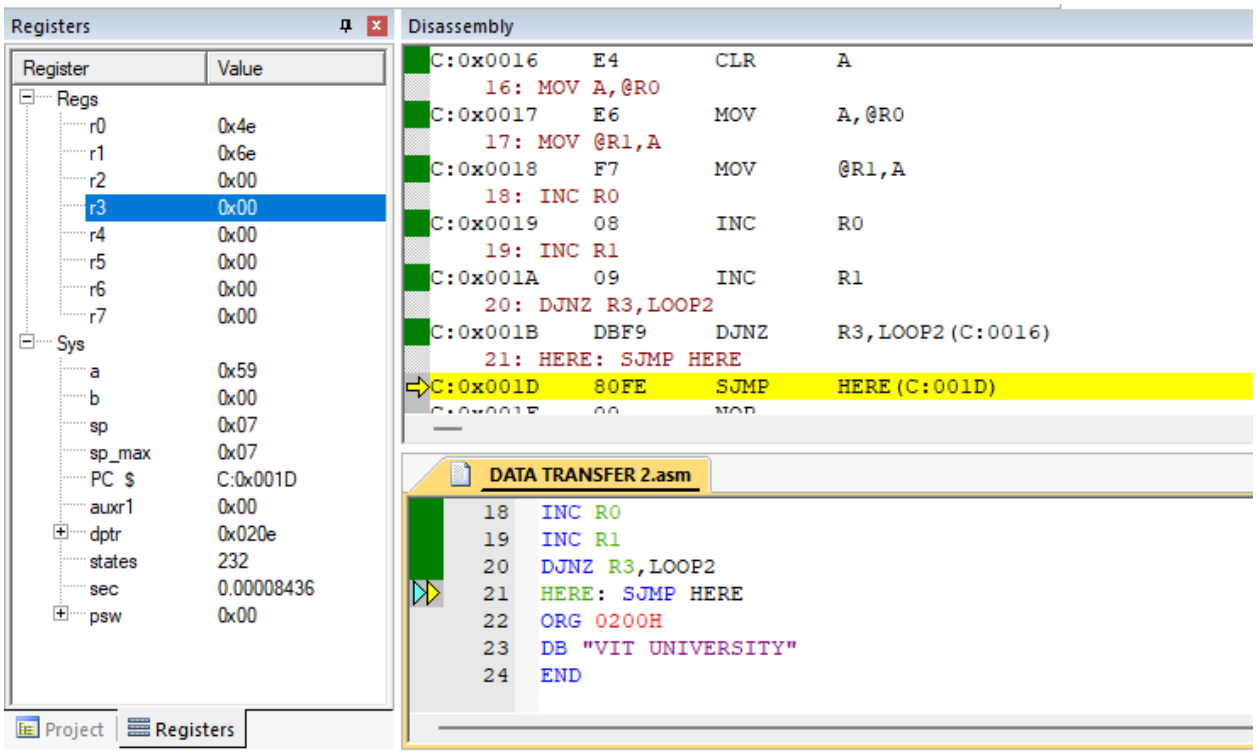
```
ORG 0000H
MOV A,#00H
MOV DPTR,#0200H
MOV R0,#40H
MOV R1,#0EH
LOOP: CLR A
MOVC A,@A+DPTR
MOV @R0,A
INC R0
INC DPTR
DJNZ R1,LOOP
MOV R0,#40H
MOV R1,#60H
MOV R3,#0EH
LOOP2: CLR A
MOV A,@R0
MOV @R1,A
INC R0
INC R1
DJNZ R3,LOOP2
HERE: SJMP HERE
ORG 0200H
DB "VIT UNIVERSITY"
END
```

Output:

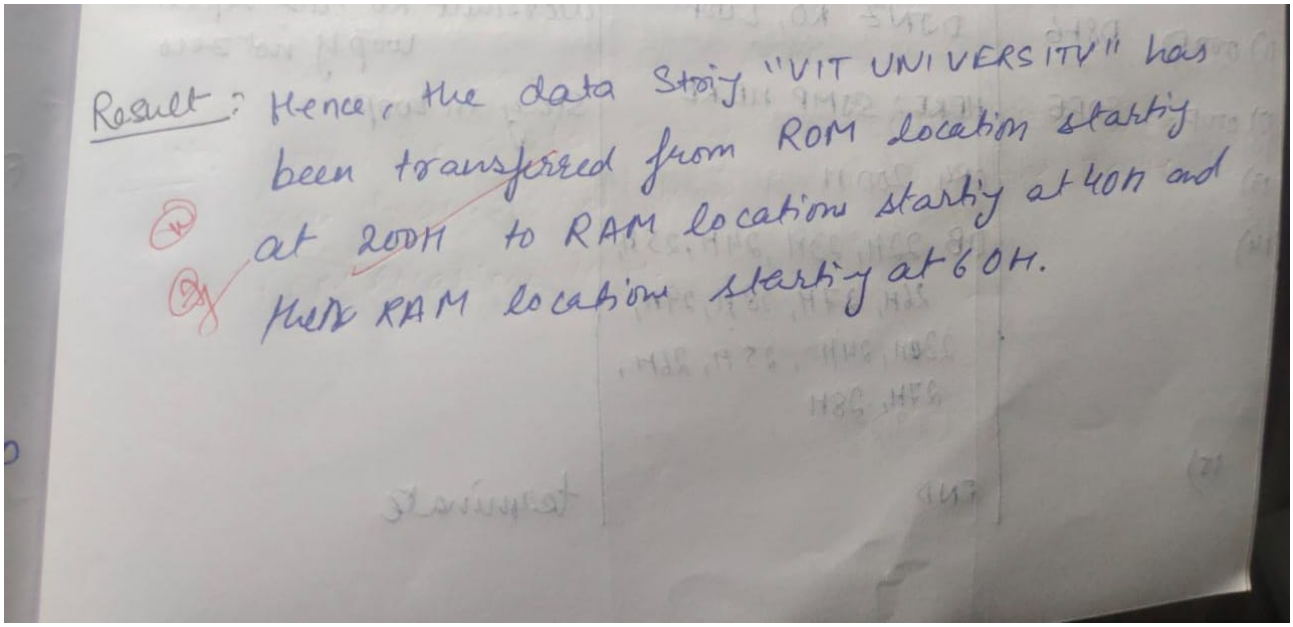
Before Execution:



After Execution:



Result:



Task 5: Data Transfer and Arithmetic Operations

Program 1:

Aim: To add 10 bytes of data stored in ROM locations starting at 200H and store the results in registers R2 and R3

Software Requirement: Keil Software

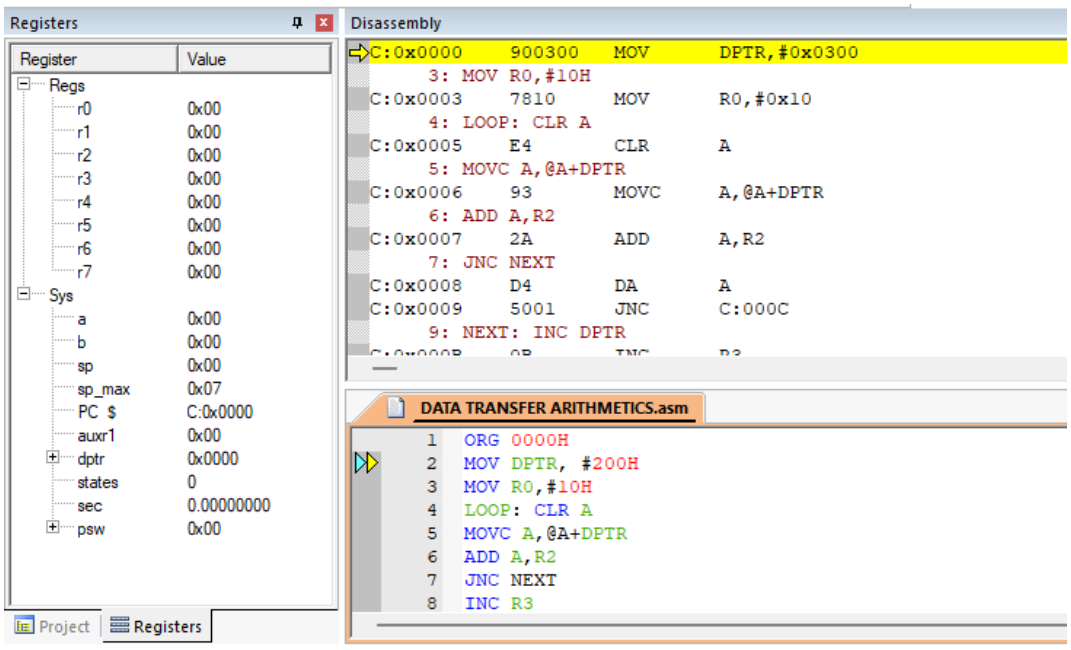
Program:

DATA TRANSFER ARITHMETICS.asm*

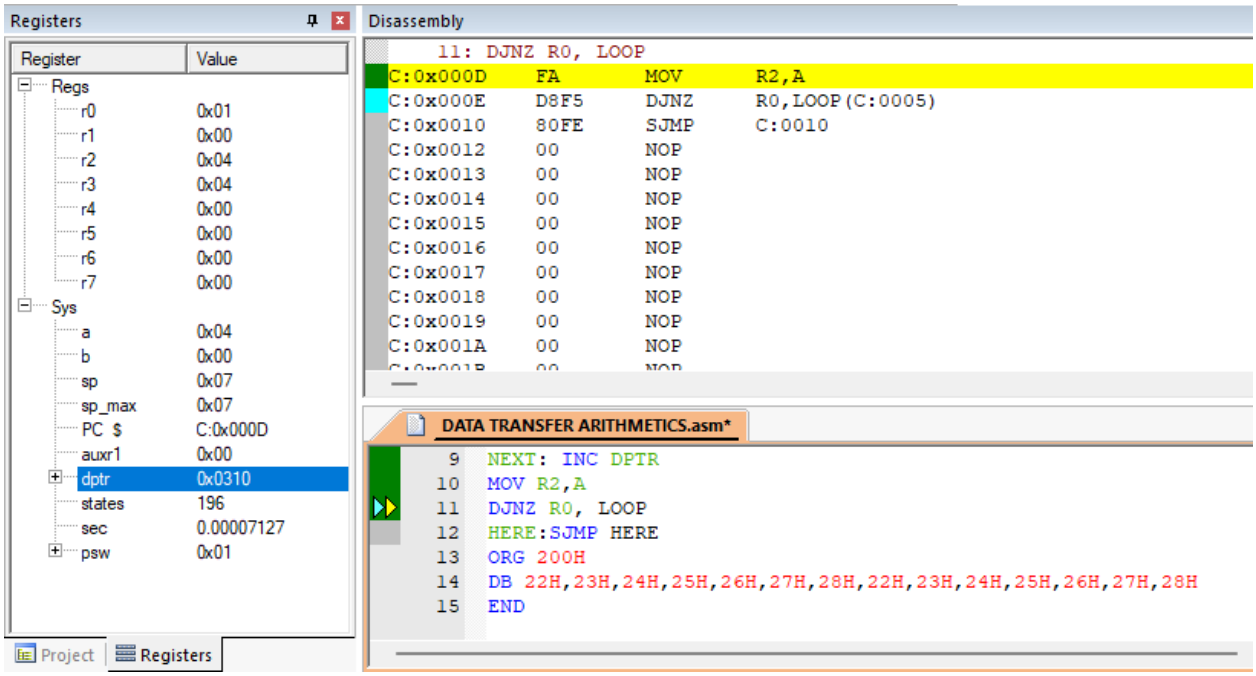
```
1  ORG 0000H
2  MOV DPTR, #200H
3  MOV R0,#10H
4  LOOP: CLR A
5  MOVC A,@A+DPTR
6  ADD A,R2
7  JNC NEXT
8  INC R3
9  NEXT: INC DPTR
10 MOV R2,A
11 DJNZ R0, LOOP
12 HERE:SJMP HERE
13 ORG 200H
14 DB 22H,23H,24H,25H,26H,27H,28H,22H,23H,24H,25H,26H,27H,28H
15 END
```

Output:

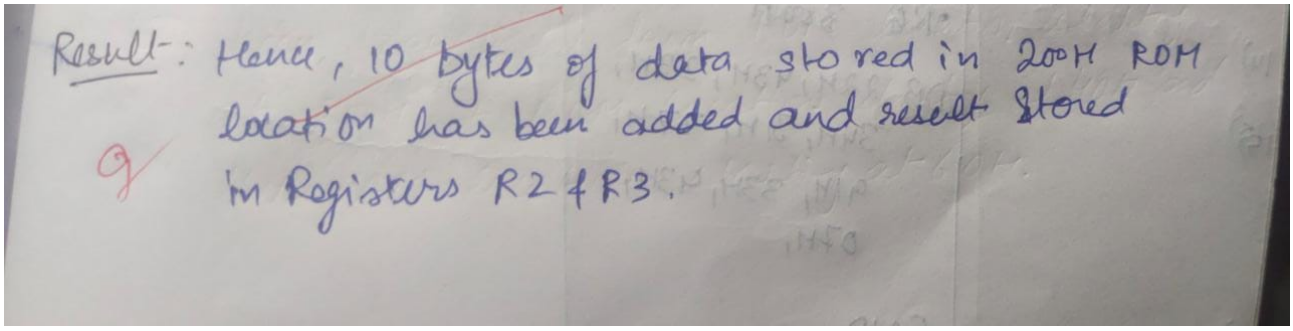
Before Execution:



After Execution:



Result:



Program 2:

Aim: To add 10 bytes od BCD data stored in ROM locations starting at 300H and store the results in registers R2 and R3

Software Requirement: Keil Software

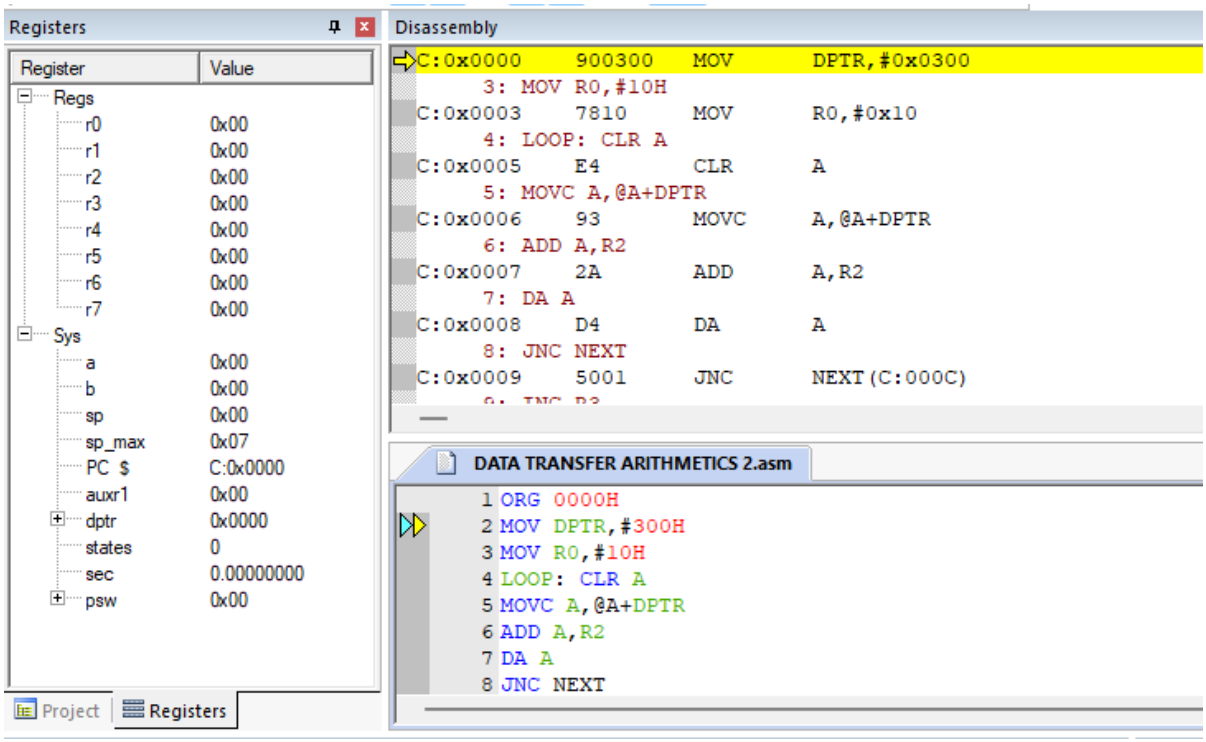
Program:

DATA TRANSFER ARITHMETICS 2.asm

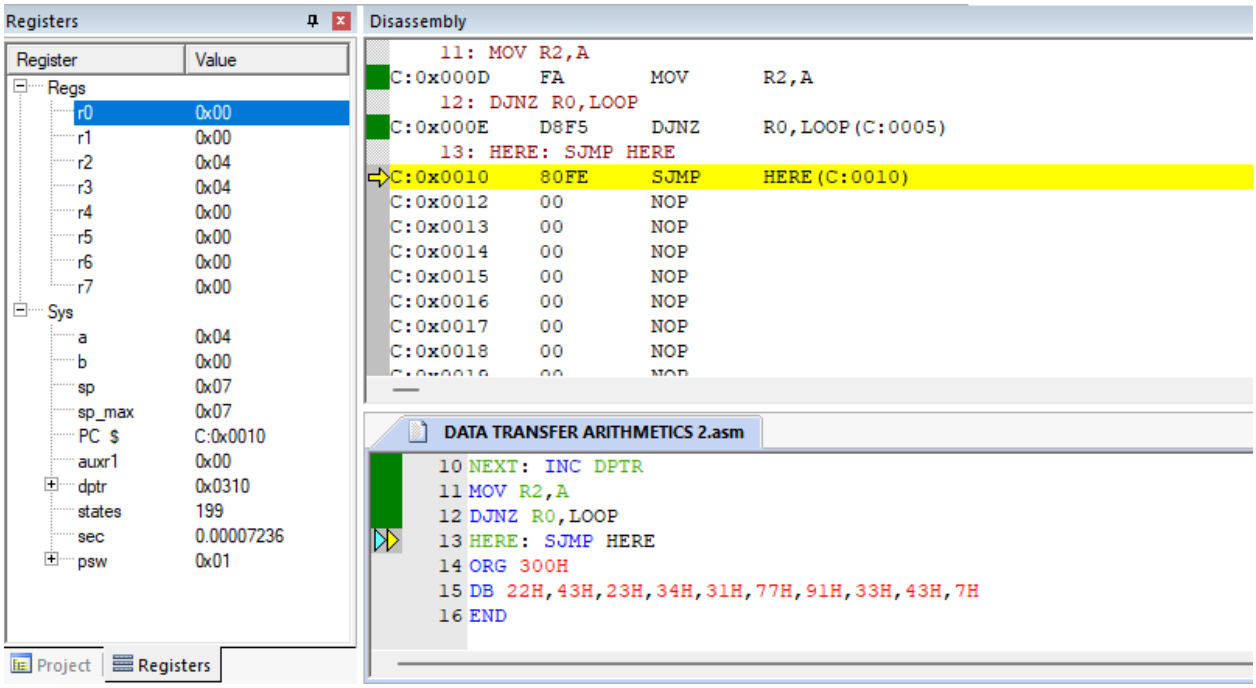
```
1  ORG 0000H
2  MOV DPTR,#300H
3  MOV R0,#10H
4  LOOP: CLR A
5  MOVC A,@A+DPTR
6  ADD A,R2
7  DA A
8  JNC NEXT
9  INC R3
10 NEXT: INC DPTR
11 MOV R2,A
12 DJNZ R0,LOOP
13 HERE: SJMP HERE
14 ORG 300H
15 DB 22H,43H,23H,34H,31H,77H,91H,33H,43H,7H
16 END
```

Output:

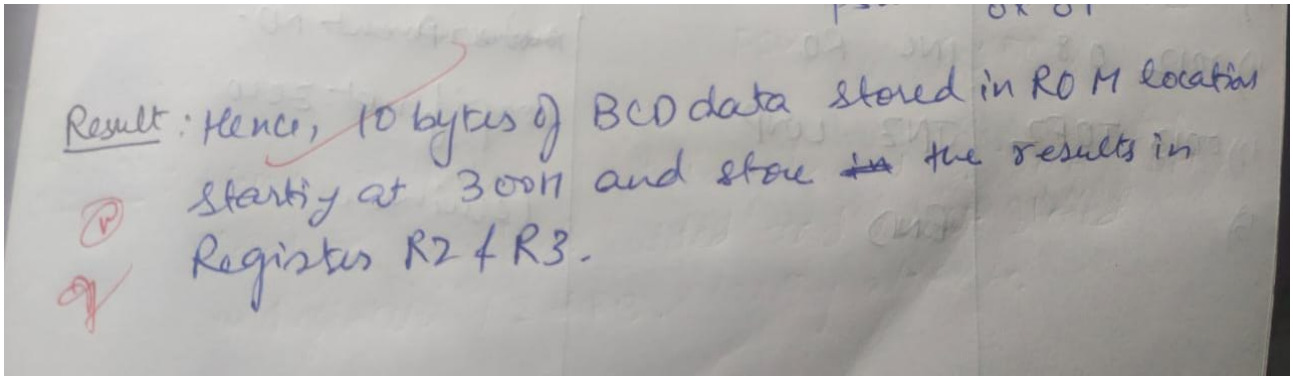
Before Execution:



After Execution:



Result:



Task 6: Base Conversion Operations

Program 1:

Aim: to get a byte of hex data from P1, convert it to decimal and then to ASCII. Then place the ASCII results into RAM locations starting at 40H

Software requirement: Keil Software

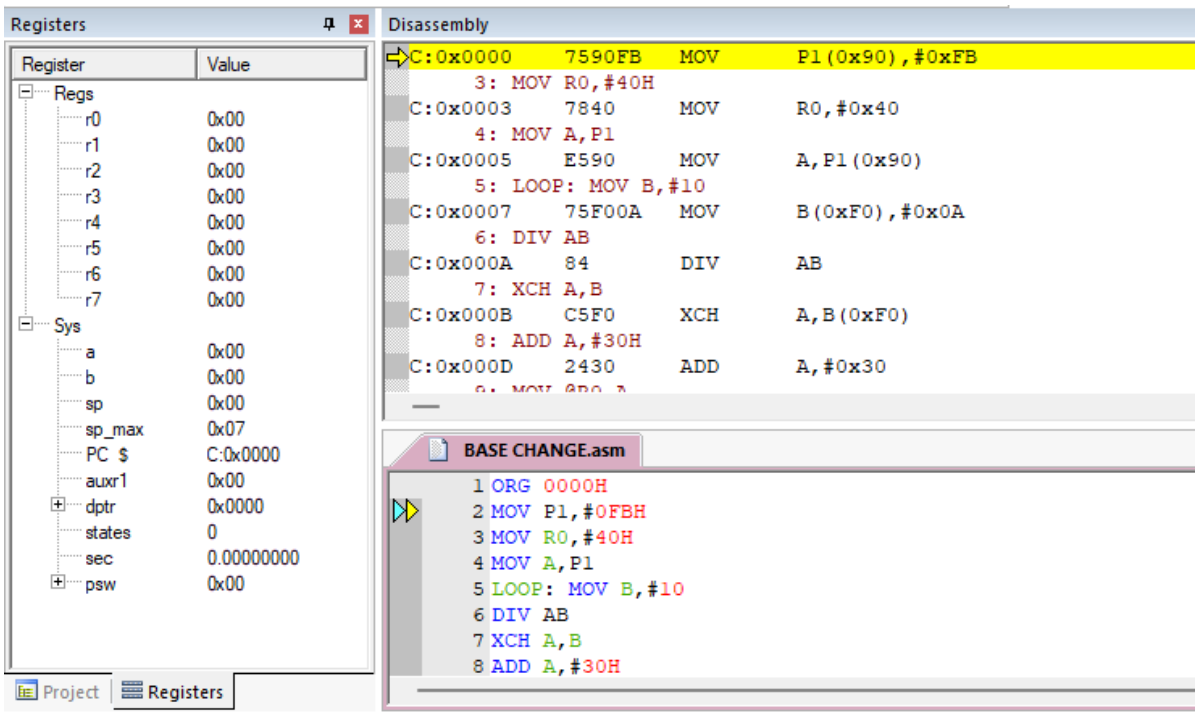
Program:

BASE CHANGE.asm

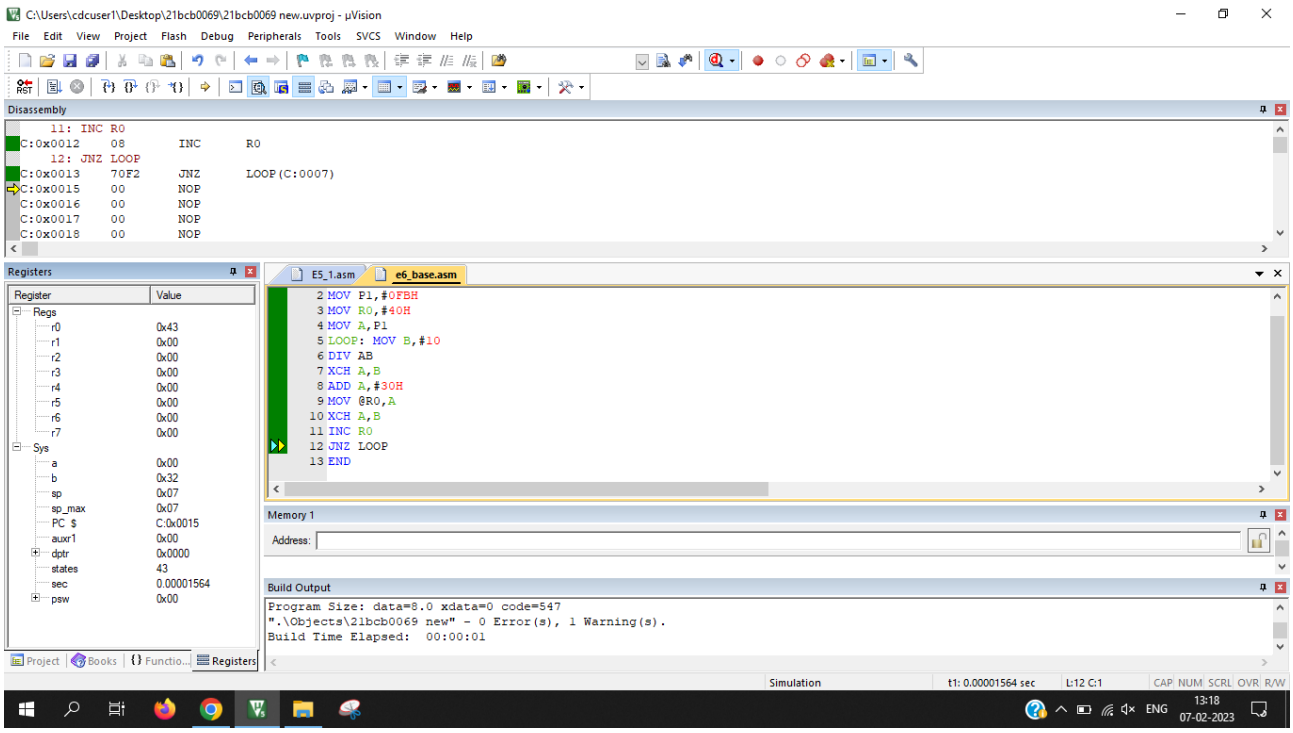
```
ORG 0000H
MOV P1,#0FBH
MOV R0,#40H
MOV A,P1
LOOP: MOV B,#10
DIV AB
XCH A,B
ADD A,#30H
MOV @R0,A
XCH A,B
INC R0
JNZ LOOP
END
```

Output:

Before Execution:



After Execution:



Result: