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

DIGITAL ASSIGNMENT-2 REPORT

REGISTER NO.	21BEC1851
NAME	RAHUL KARTHIK S
SCHOOL/PROGRAM	SENSE/B. Tech (ECE/ECM)
SEMESTER/SLOT	Winter 2022-23 / A1+TA1 / A2+TA2
COURSE CODE / NAME	BECE204L – Microprocessors and Microcontrollers
DATE OF ANNOUNCEMENT	05/03/2023
LAST DATE FOR SUBMISSION	12/03/2023 (Sunday 11:59 PM)

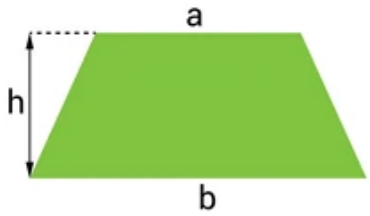
Q. No.	TITLE	MARKS
1	Arithmetic operation	
2	Timer	
3	Counter	
4	Serial communication	
5	Interrupt & LCD	
	Total Marks	

COURSE HANDLER'S NAME	Dr. V. PRAKASH
COURSE HANDLER'S SIGNATURE	

1. Arithmetic Operation

Question:

1. Write an 8051 assembly language program to find the area of the Trapezoid as given below,



$$\text{Area} = \frac{(a+b)\times h}{2}$$

Assume the value of "a" and "b" as first and last two digits of your register number. (For example, if Reg. no is 21BEC1073 assume "a" as 10H and "b" as 73H). Assume the value of "h" as 01H.

Algorithm:

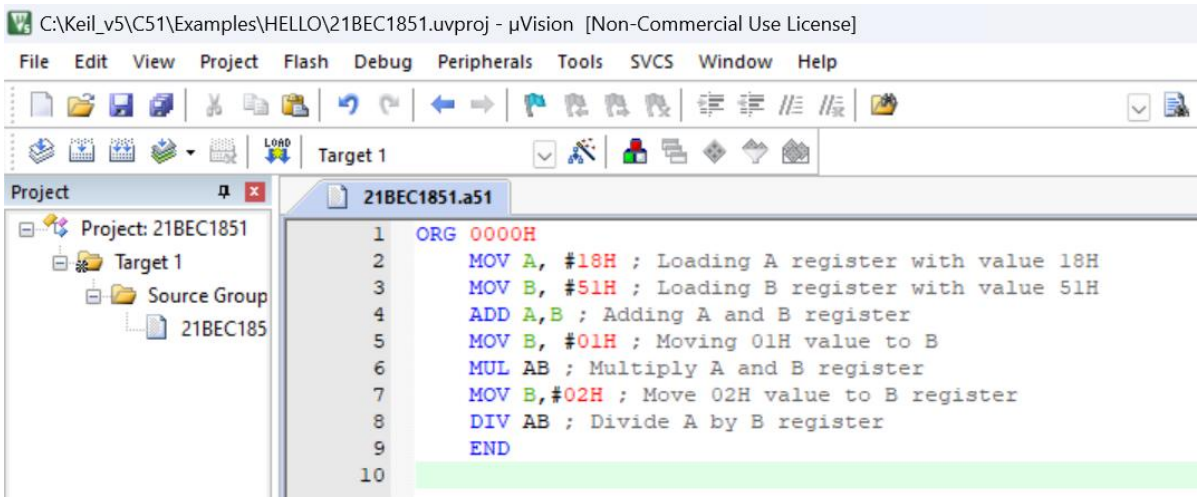
- Loading A register with value 18H
- Loading B register with value 51H
- Adding A and B register
- Moving 01H value to B.
- Multiply A and B register
- Move 02H value to B register
- Divide A by B

Program:

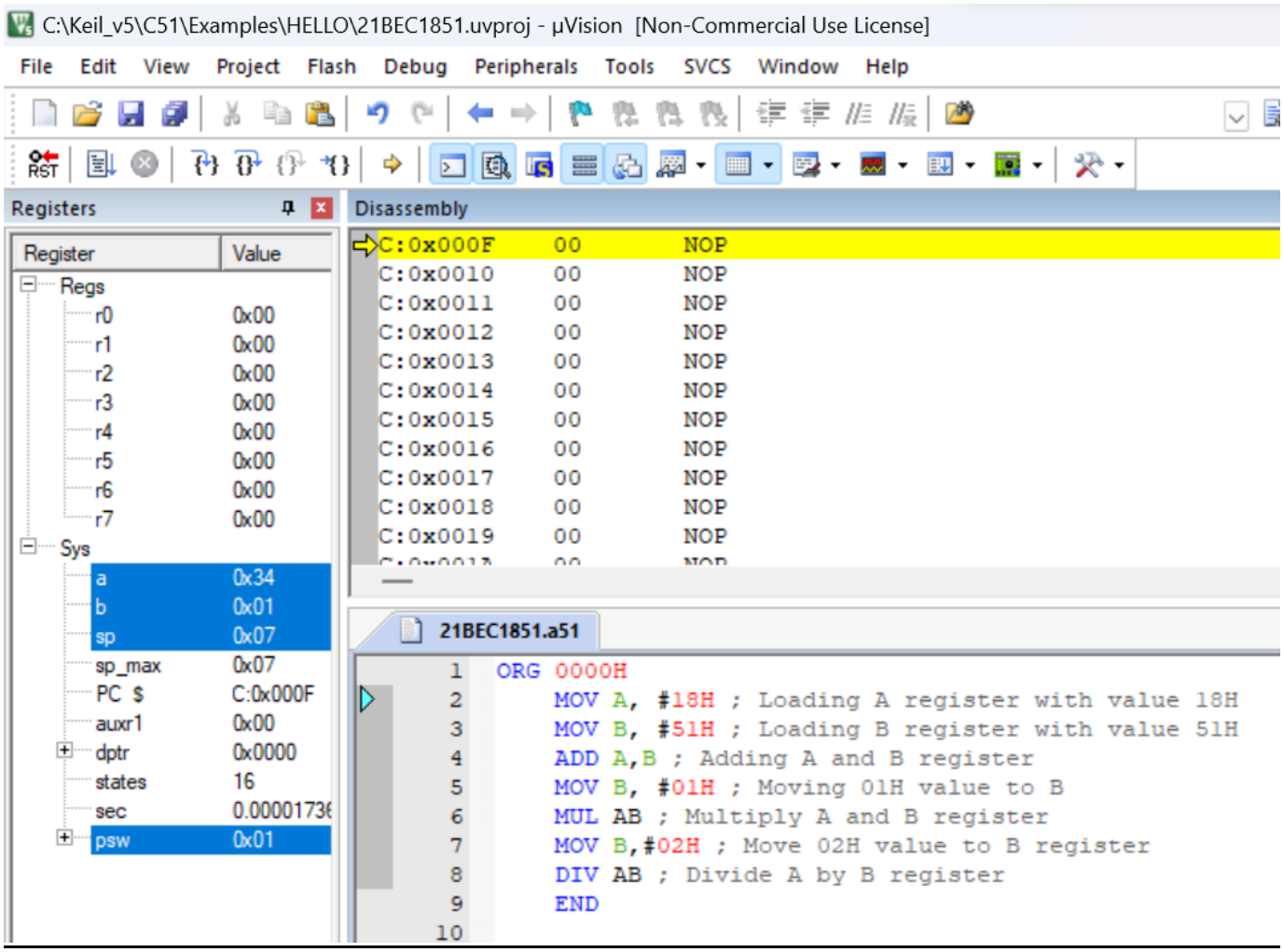
```
ORG 0000H
    MOV A, #18H ; Loading A register with value 18H
    MOV B, #51H ; Loading B register with value 51H
    ADD A,B ; Adding A and B register
    MOV B, #01H ; Moving 01H value to B
    MUL AB ; Multiply A and B register
    MOV B,#02H ; Move 02H value to B register
    DIV AB ; Divide A by B register
    END
```

Screenshots:

(a) Program



(b)Output



Inference:

- In this program the result is stored in the A register and it is 34H.

2. Waveform Generation

Question:

Generate a waveform on Port 0.1 pin by writing a program using 8051 assembly program as stated below. The waveform must have fixed OFF time, which is last two digits of "(Your Reg. no)" ms (example for 21BEC1015 time delay is 15 ms). The ON time must depend on the 8-bit number that is read from the Port 1 as given in the table.

Number read from Port 1	ON Time	OFF Time
ODD Number	5 ms	last two digit of "(Your Reg. no)" ms
EVEN Number	10 ms	last two digit of "(Your Reg. no)" ms

Assume Port 0 as output port, Port 1 as input port and XTAL = 11.0592 MHz.

Algorithm:

- Storing an even value to P1 Port
- Moving P1 to Accumulator
- Rotating Right bit-by-bit with carry to check whether last digit is zero or one
- Jump if it is odd
- Using Waveform Generation Code for different ON and OFF times.
- Now, in the delay Generation Code, storing TMOD Value as 10H.
- Storing the TL1 and TH1 value from calculations.
- Set TR1 pin
- Checking the overflow conditions.
- Clear TR1 and TF1 pins.
- Return back to call statement.
- Follow the same procedure to create delay for different time periods.

Program:

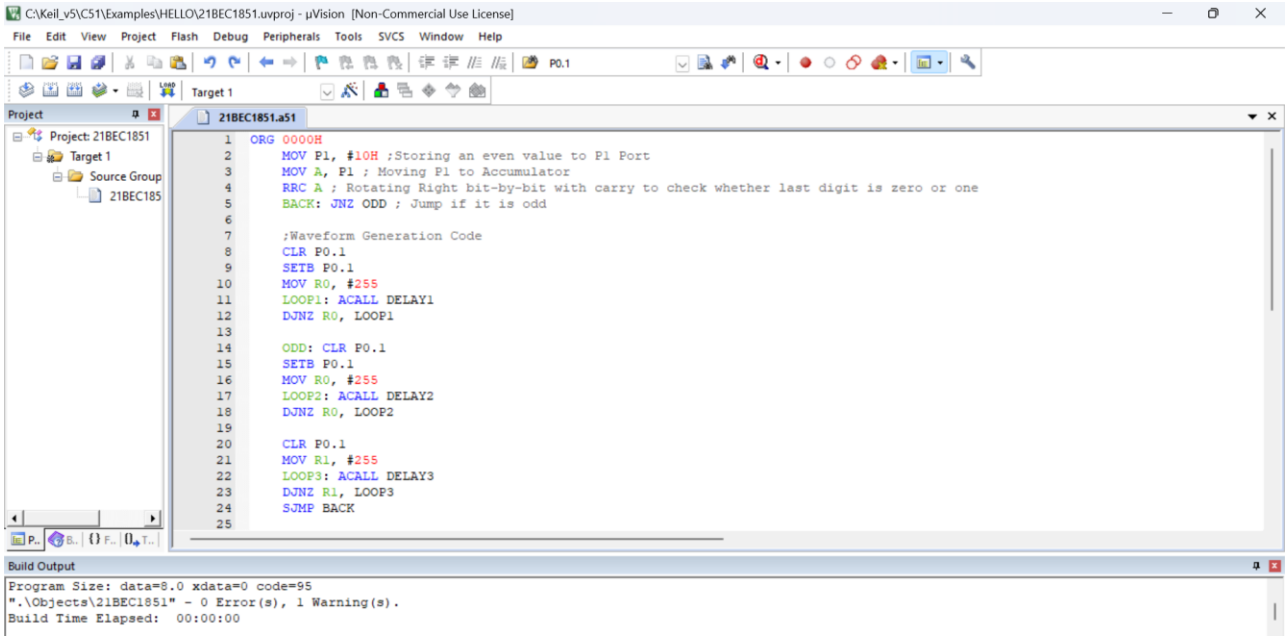
```
ORG 0000H
        MOV P1, #10H; Storing an even value to P1 Port
        MOV A, P1; Moving P1 to Accumulator
        RRC A ; Rotating Right bit-by-bit with carry to check whether last digit is zero or
one
        BACK: JNZ ODD ; Jump if it is odd

        ;Waveform Generation Code
        CLR P0.1
        SETB P0.1
```

```
MOV R0, #255
LOOP1: ACALL DELAY1
DJNZ R0, LOOP1

ODD: CLR P0.1
SETB P0.1
MOV R0, #255
LOOP2: ACALL DELAY2
DJNZ R0, LOOP2

CLR P0.1
MOV R1, #255
LOOP3: ACALL DELAY3
DJNZ R1, LOOP3
SJMP BACK

; 10 ms Delay
DELAY1: MOV TMOD, #10H ; Storing TMOD Value as 10H
MOV TL1, #0DCH ; Storing the TL1 value from calculations
MOV TH1, #0FDH ; Storing the TH1 value from calculations
SETB TR1 ; Set TR1 pin
BACK1: JNB TF1, BACK1 ; Checking the overflow conditions
CLR TR1 ; Clear TR1 pin
CLR TF1 ; Clear TF1 pin
RET ; Return to call statement

; 5 ms Delay
DELAY2: MOV TMOD, #10H
MOV TL1, #0EEH
MOV TH1, #0FFH
SETB TR1
BACK2: JNB TF1, BACK2
CLR TR1
```

```
CLR TF1
RET

;51 ms Delay
DELAY3: MOV TMOD, #10H
MOV TL1, #48H
MOV TH1, #0FFH
SETB TR1
BACK3: JNB TF1, BACK3
CLR TR1
CLR TF1
RET
```

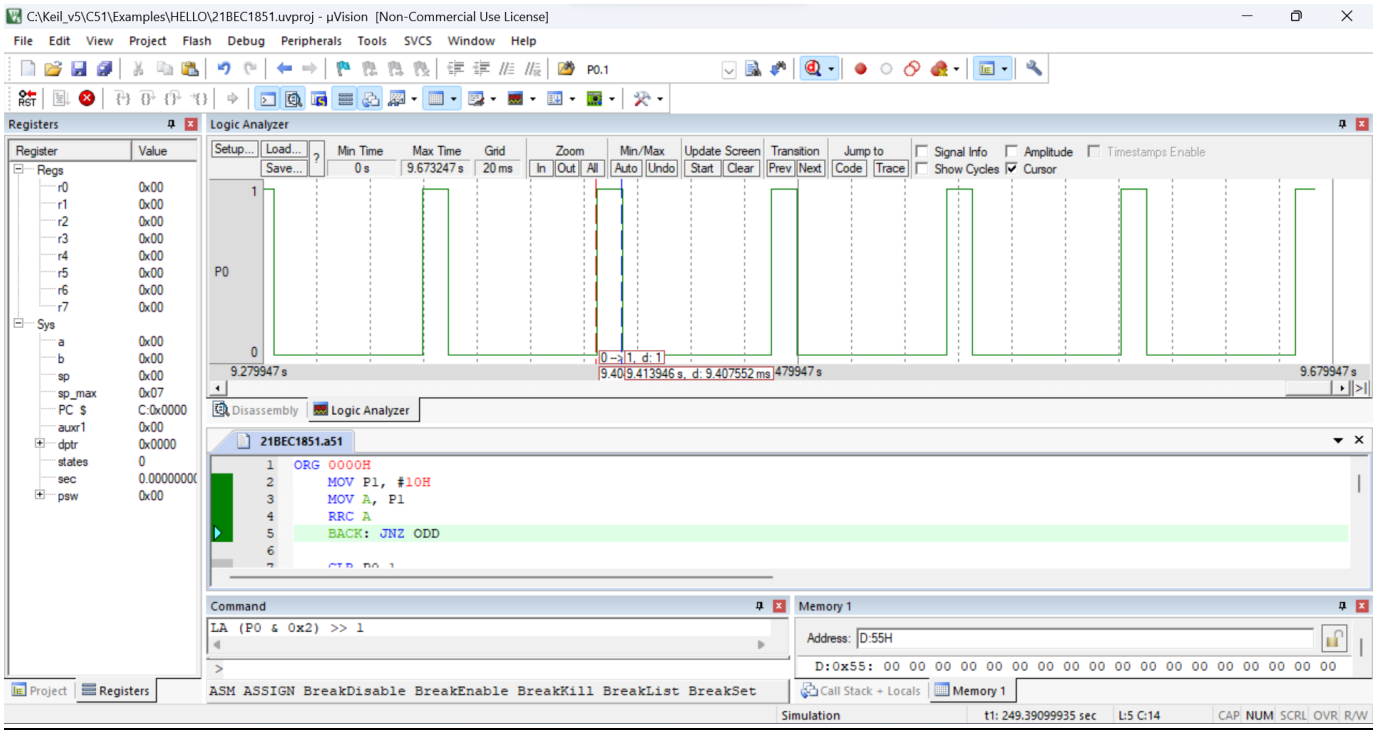
Screenshots:

(a) Program

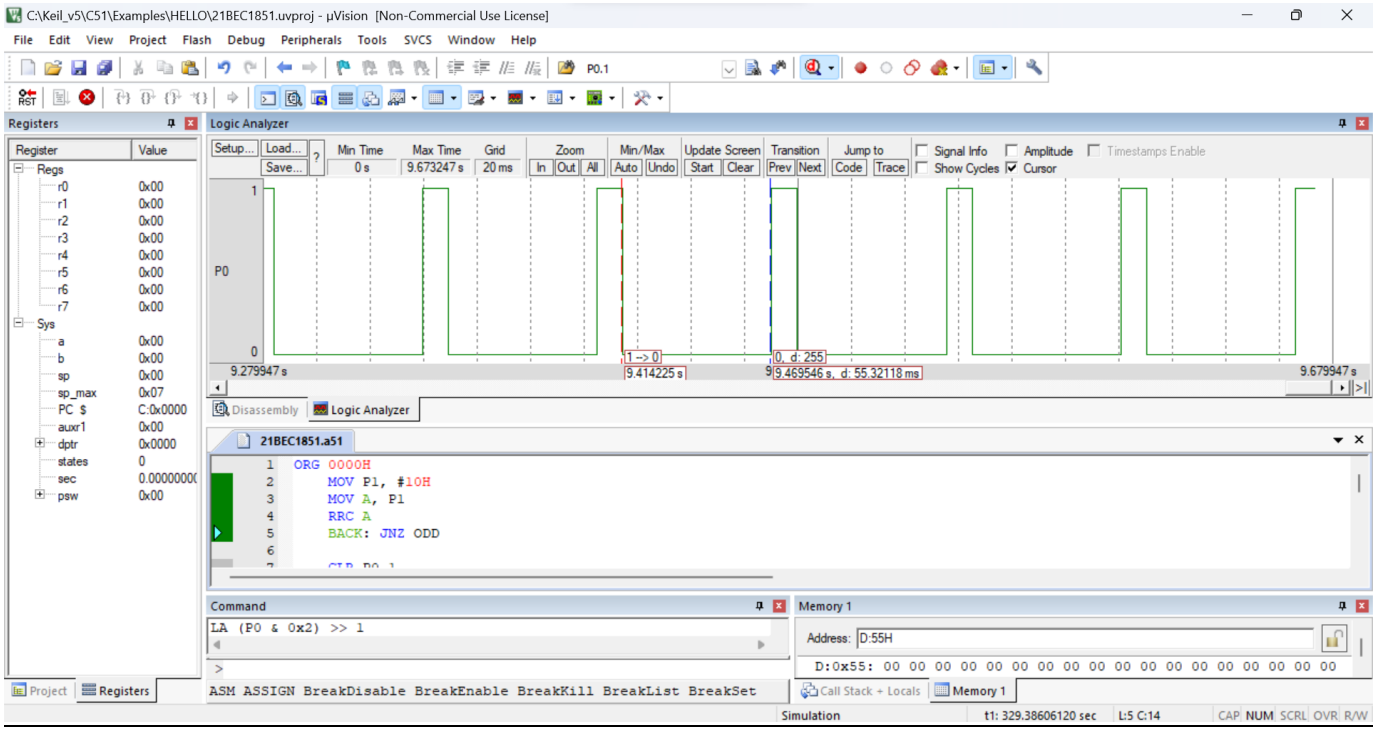


(b) Output

ON Time:



Off Time:



Inference:

If the port P1 pin is an even number, ON time waveform for 10 ms delay and if it is odd number, ON time waveform for 5 ms delay is generated. It has a default OFF time of 51 ms delay is generated.

3. Counters

Question:

In the packaging department of a Noodle manufacturing company, each noodle packet is rolls down on a conveyor and packed into the empty box for shipment. Assume each empty box can accommodate "X" number of Noodle's packet, where "X" is last two digits of your register number. Each Noodle packet is allowed to pass through IR scanner, which generates a one clock pulse for every packet that crosses the scanner. Whenever "X" number of packet gets filled in the empty box, activate the sealing machine by sending HIGH signal. Write an 8051 assembly language program by assuming clock pulses are received at port P3.4 and the sealing machine is connected at port P2.1 pin.

Algorithm:

- Maximum number of packets that can fit in a box is stored.
- Set P2.1 as output and P3.4 as input.
- Number of packets in the current box in R0.
- Check if pulse is received.
- Check if box is full.
- Wait for interrupts
- In IR pulse interrupt service routine, Clear carry flag, Increment packet count and Continue waiting for interrupts.
- Check if box is full subroutine, Check if box is not full, Seal the box, Reset packet count and Return from subroutine.

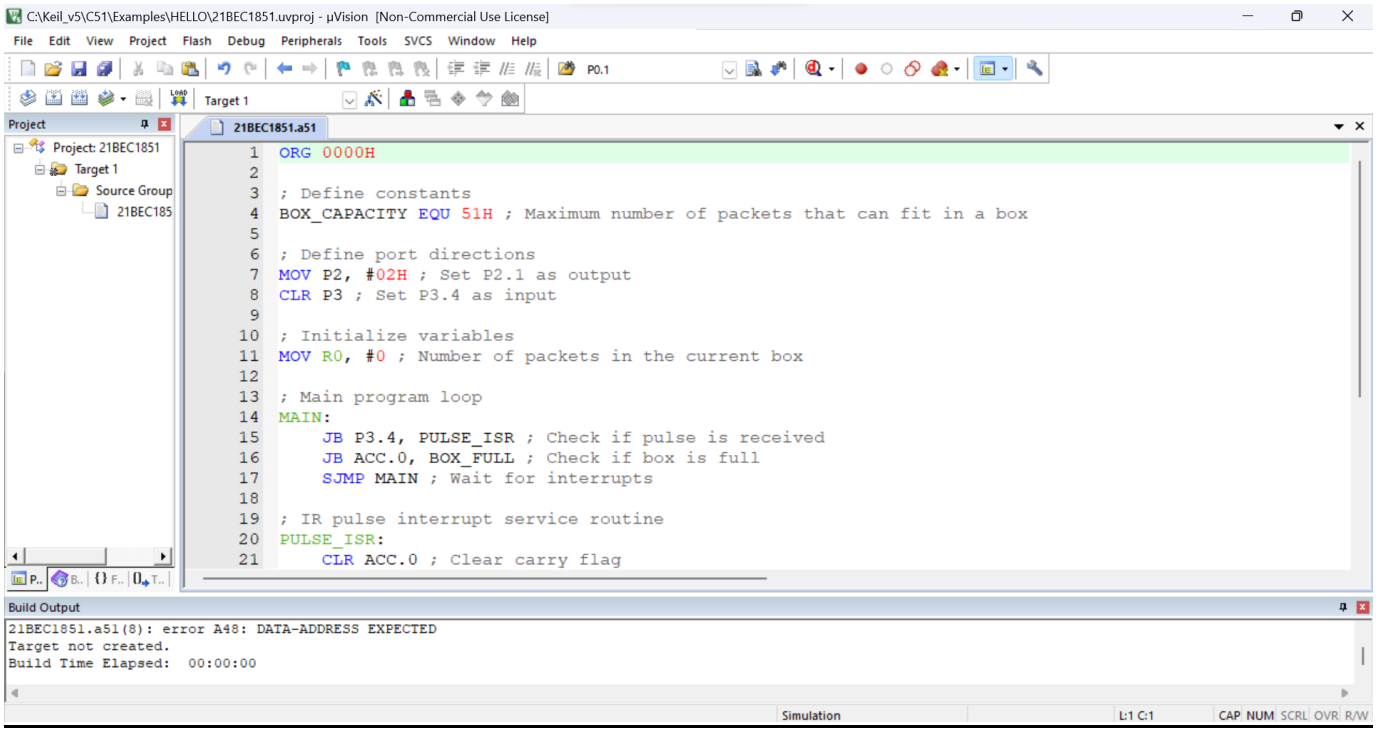
Program:

```
ORG 0000H

; Define constants
BOX_CAPACITY EQU 51H ; Maximum number of packets that can fit in a box

; Define port directions
MOV P2, #02H ; Set P2.1 as output
CLR P3 ; Set P3.4 as input

; Initialize variables
MOV R0, #0 ; Number of packets in the current box
```

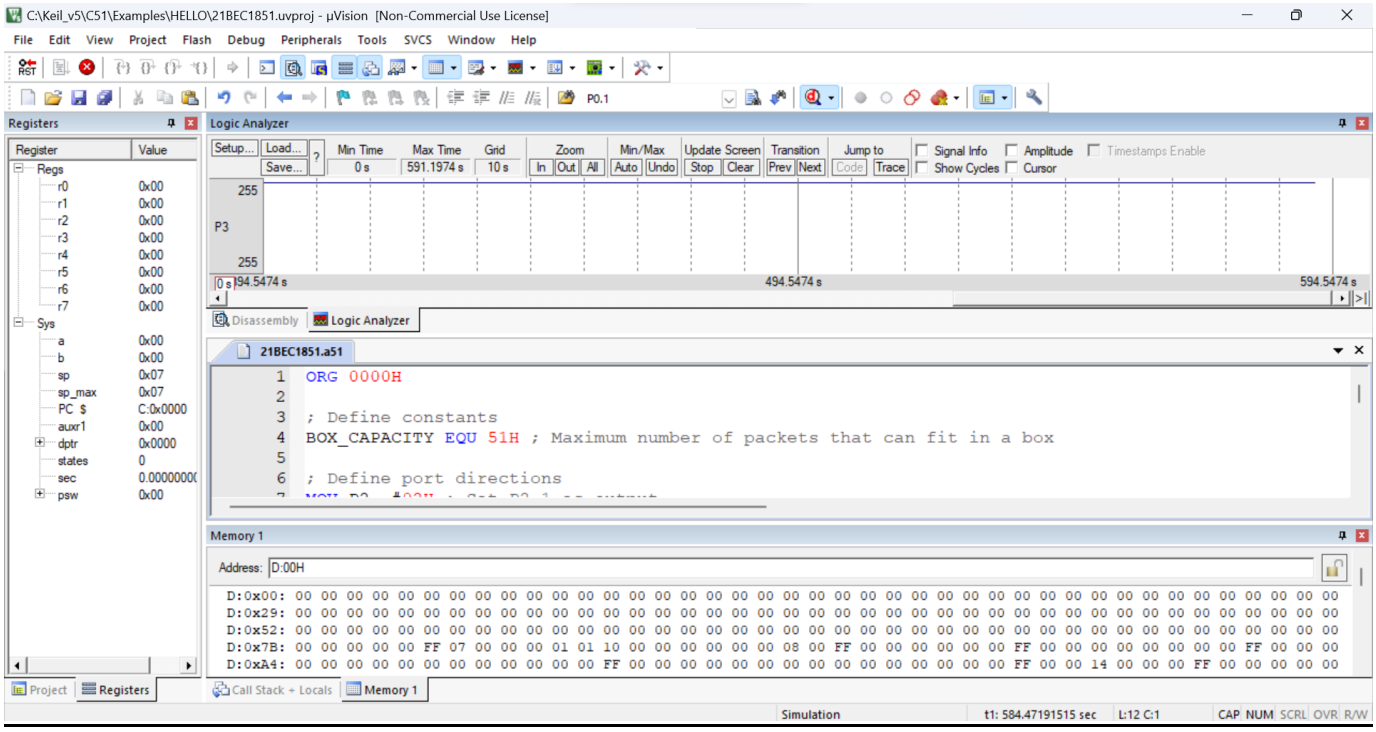
```
; Main program loop
MAIN:
    JB P3.4, PULSE_ISR ; Check if pulse is received
    JB ACC.0, BOX_FULL ; Check if box is full
    SJMP MAIN ; Wait for interrupts

; IR pulse interrupt service routine
PULSE_ISR:
    CLR ACC.0 ; Clear carry flag
    INC R0 ; Increment packet count
    SJMP MAIN ; Continue waiting for interrupts

; Check if box is full subroutine
BOX_FULL:
    CJNE R0, #BOX_CAPACITY, MAIN ; Check if box is not full
    SETB P2.1 ; Seal the box
    MOV R0, #0 ; Reset packet count
    RET ; Return from subroutine

END
```

Screenshots:

(a)Program



(b) Output



Inference:

Thus, the high signals are produced at sealing machine when box is full.

4. Serial Communication

Question:

Assume that the 8051 serial port is connected to the COM port of PC, and on the PC, we are using the terminal.exe program to send and receive data serially. Whenever a character "R" is send by PC to 8051, transmit a message "your register no" to PC. Whenever a character "N" is send by PC to 8051, transmit a message "Your Name" to PC.

Algorithm:

- Initialize serial port with Timer 1, Mode 2, Baud rate 9600 and Start timer 1.
- In Main loop, wait for receive interrupt flag, Read received character
- If character is not 'R', jump to NOT_R, If character is 'R', transmit "21BEC1851" to PC
- Continue receiving characters using SJMP.
- If character is 'N', transmit "RAHUL" to PC.
- Continue receiving characters.
- Wait for transmit interrupt flag and Clear transmit interrupt flag

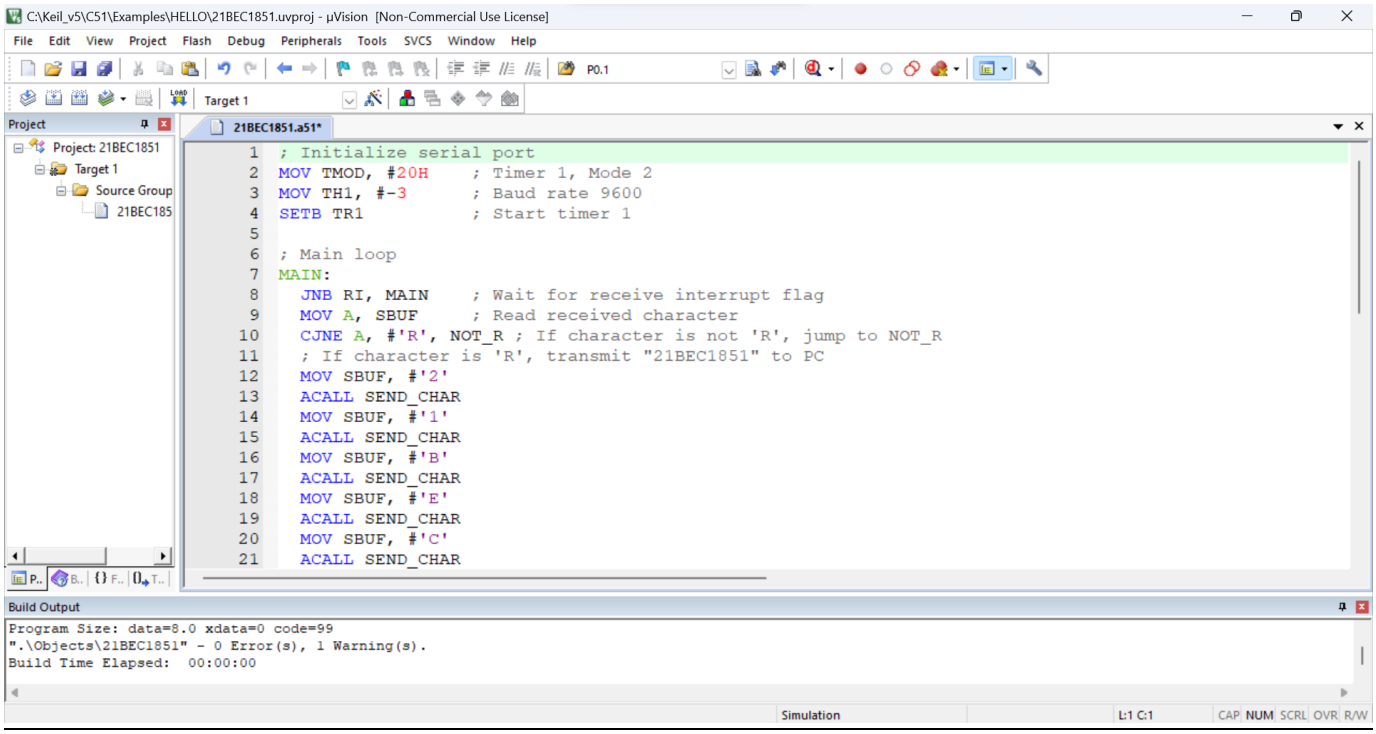
Program:

```
; Initialize serial port
MOV TMOD, #20H    ; Timer 1, Mode 2
MOV TH1, #-3      ; Baud rate 9600
SETB TR1          ; Start timer 1

; Main loop
MAIN:
  JNB RI, MAIN    ; Wait for receive interrupt flag
  MOV A, SBUF     ; Read received character
  CJNE A, #'R', NOT_R ; If character is not 'R', jump to NOT_R
  ; If character is 'R', transmit "21BEC1851" to PC
  MOV SBUF, #'2'
  ACALL SEND_CHAR
  MOV SBUF, #'1'
  ACALL SEND_CHAR
  MOV SBUF, #'B'
  ACALL SEND_CHAR
  MOV SBUF, #'E'
  ACALL SEND_CHAR
  MOV SBUF, #'C'
```

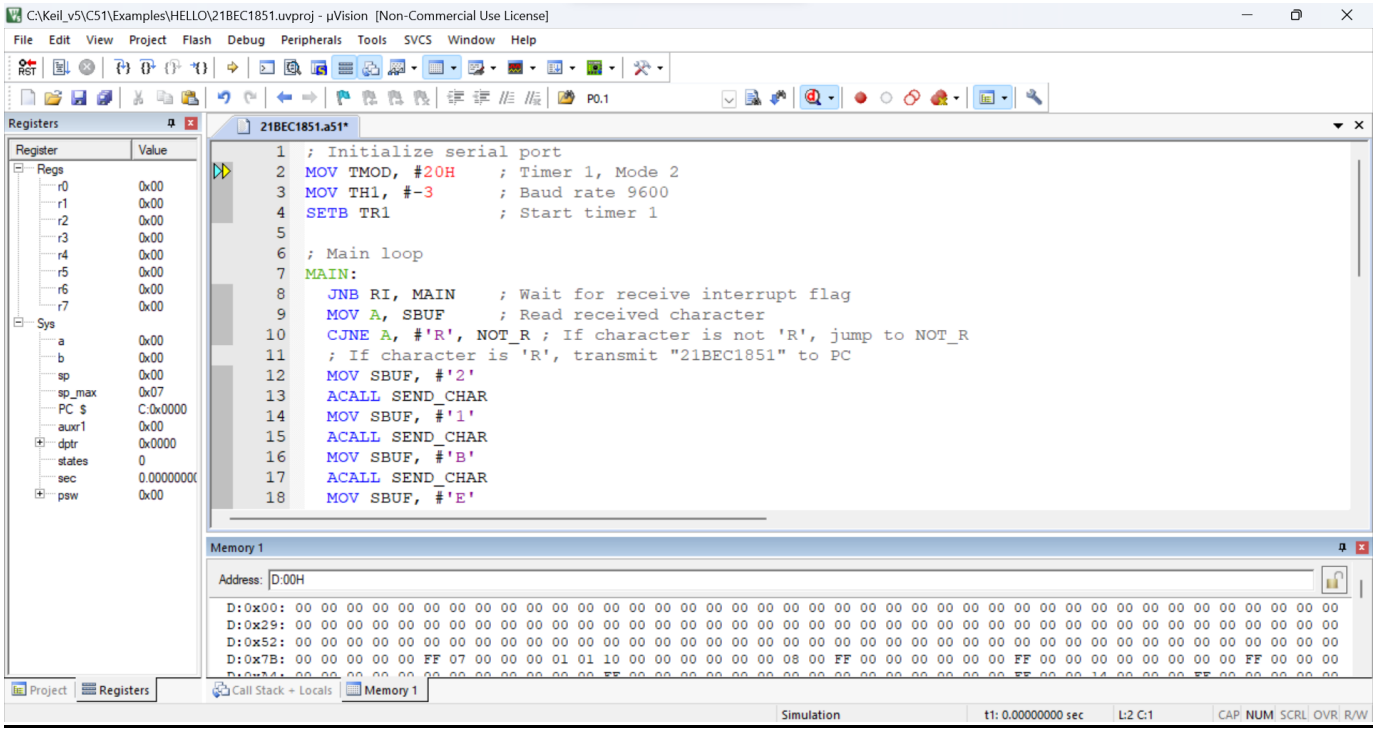
```
  ACALL SEND_CHAR
  MOV SBUF, #'1'
  ACALL SEND_CHAR
  MOV SBUF, #'8'
  ACALL SEND_CHAR
  MOV SBUF, #'5'
  ACALL SEND_CHAR
  MOV SBUF, #'1'
  ACALL SEND_CHAR
  SJMP MAIN      ; Continue receiving characters

NOT_R:
  CJNE A, #'N', MAIN ; If character is not 'N', jump to MAIN
  ; If character is 'N', transmit "RAHUL" to PC
  MOV SBUF, #'R'
  ACALL SEND_CHAR
  MOV SBUF, #'A'
  ACALL SEND_CHAR
  MOV SBUF, #'H'
  ACALL SEND_CHAR
  MOV SBUF, #'U'
  ACALL SEND_CHAR
  MOV SBUF, #'L'
  ACALL SEND_CHAR
  SJMP MAIN      ; Continue receiving characters

SEND_CHAR:
  JNB TI, SEND_CHAR ; Wait for transmit interrupt flag
  CLR TI         ; Clear transmit interrupt flag
  RET
```

Screenshots:

(a)Program



(b) Output



Inference:

Thus, the data is being received and based on the received value, the value is being transmitted.

5. LED

Question:

Write a assembly language program to configure 8051 microcontrollers to display the message on the LCD based on external interrupt received.

- Whenever an interrupt 0 (INT0) triggered display the message "(Your Reg. no)" (example for 21BEC1001 display 21BEC1001) on the first line of the LCD

- Whenever an interrupt 1 (INT1) triggered display the message "(your name)" (example Aswin Kumar) on the second line of the LCD

Algorithm:

- We are writing Command and Data for MYDATA1 and MYDATA2.
- We are writing command subroutine by clearing P2.0, P2.1 and setting P2.1 and then clearing P2.2.
- We are writing data writing subroutine by setting P2.0, P2.2 and clearing P2.1 and then clearing P2.2.
- We are writing delay subroutine by using 2 inner loops.
- We are writing command values and Data values using DB.
- We are triggering INTR0 and INTR1 and moving MYDATA1 and MYDATA2 in DPTR and returning.

Program:

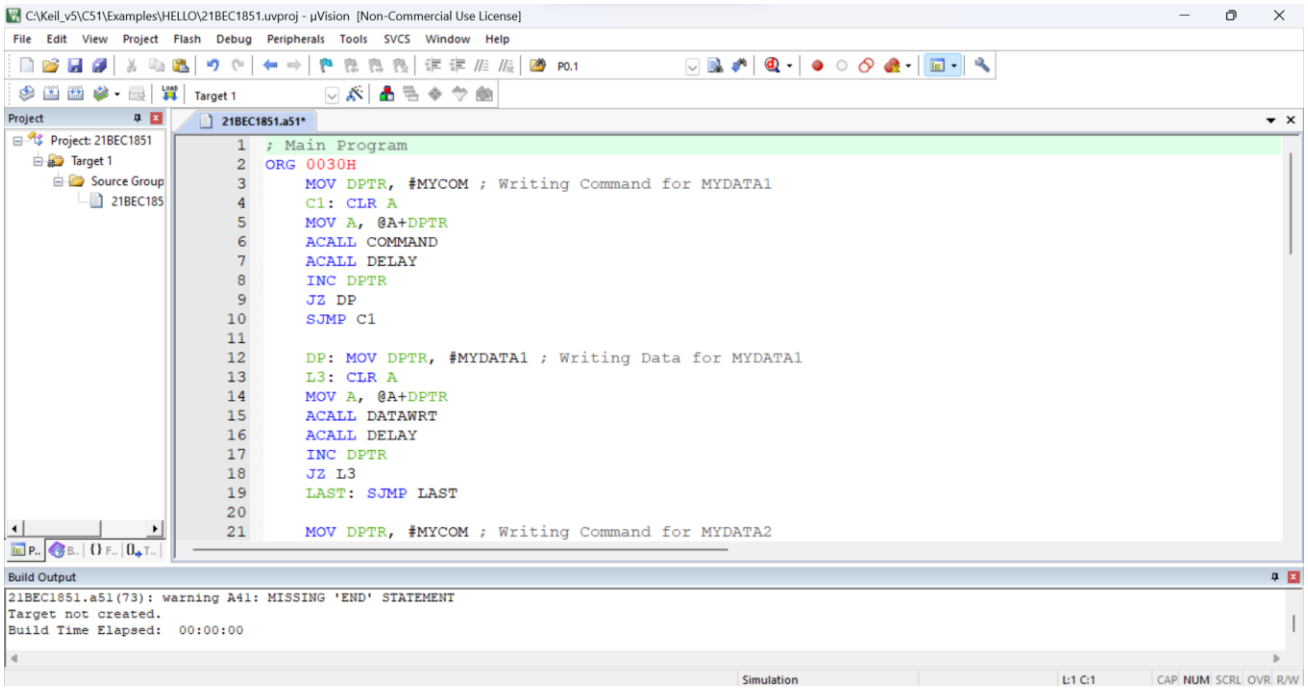
```
; Main Program
ORG 0030H
        MOV DPTR, #MYCOM ; Writing Command for MYDATA1
        C1: CLR A
        MOV A, @A+DPTR
        ACALL COMMAND
        ACALL DELAY
        INC DPTR
        JZ DP
        SJMP C1

        DP: MOV DPTR, #MYDATA1 ; Writing Data for MYDATA1
        L3: CLR A
        MOV A, @A+DPTR
        ACALL DATAWRT
```

```
ACALL DELAY
INC DPTR
JZ L3
LAST: SJMP LAST

MOV DPTR, #MYCOM ; Writing Command for MYDATA2
C2: CLR A
MOV A, @A+DPTR
ACALL COMMAND
ACALL DELAY
INC DPTR
JZ DP2
SJMP C2

DP2: MOV DPTR, #MYDATA2 ; Writing Data for MYDATA2
L2: CLR A
MOV A, @A+DPTR
ACALL DATAWRT
ACALL DELAY
INC DPTR
JZ L2
LAST2: SJMP LAST2

COMMAND: ; Writing the Command Subroutine
MOV P1, A
CLR P2.0
CLR P2.1
SETB P2.2
CLR P2.2
RET

DELAY: ; Writing the Delay Subroutine
MOV R0, #0FFH
```

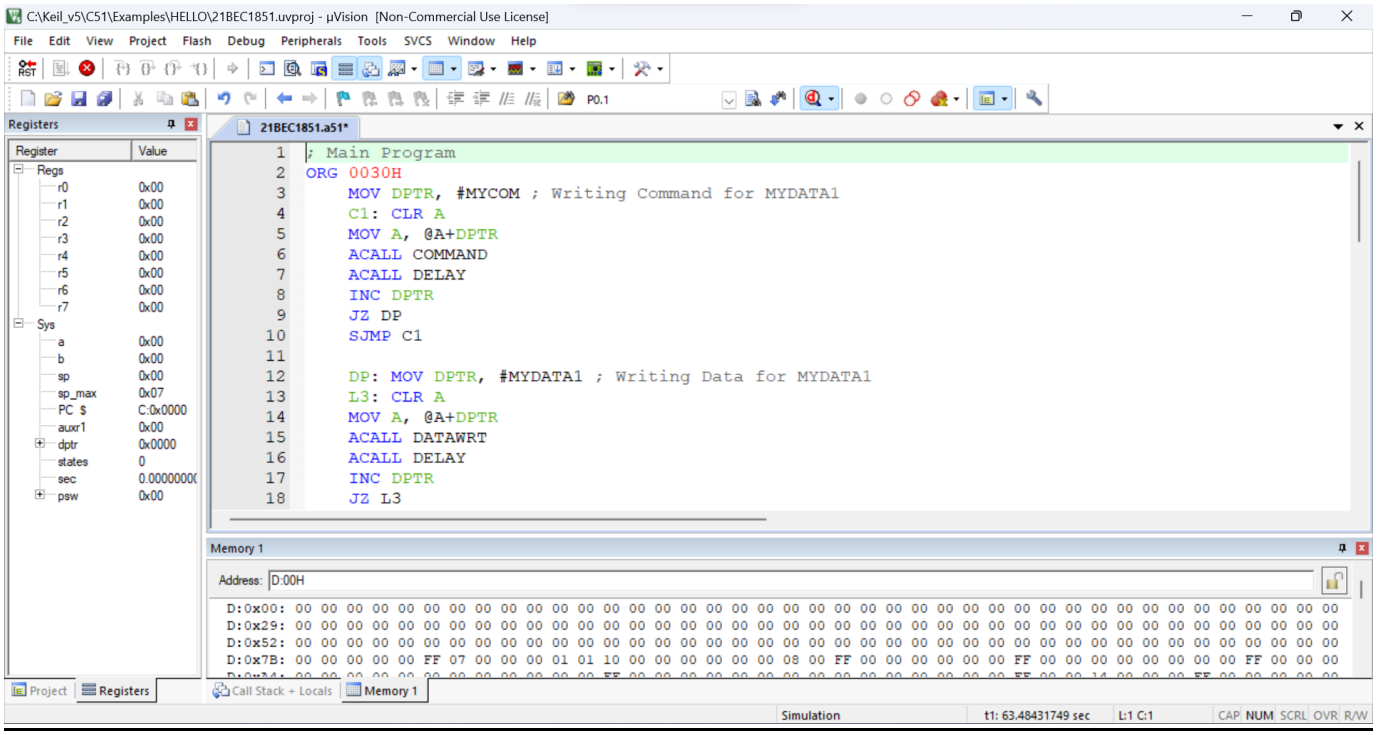
```
L2: MOV R1, #0FFH
L1: DJNZ R1, L1
DJNZ R0, L2
RET

DATAWRT: ; Writing the Data writing Subroutine
MOV P1, A
SETB P2.0
CLR P2.1
SETB P2.2
CLR P2.2
RET

ORG 0300H ; Writing Command values and Data values in 0300H memory locations
        MYCOM DB 38, 0E, 01, 06, 83H, 0
        MYDATA1 DB "21BEC1851", 0
        MYDATA2 DB "RAHUL KARTHIK S", 0

ORG 0003H ; MYDATA1 is displayed when INTR0 is triggered
        MOV DPTR, #MYDATA1
        RETI

ORG 0013H ; MYDATA2 is displayed when INTR1 is triggered
        MOV DPTR, #MYDATA2
        RETI
```

Screenshots:

(a) Program



(b) Output



Inference:

Thus, the respective message is displayed in LED when INTR0 and INTR1 is trigged respectively.