

Exercises:

1. Write an assembly program where the user can input their name and save it into an array. Set aside 15 elements for your array (not including the termination element) starting at memory location 40. Set up your code to exit early when the user presses enter.

```

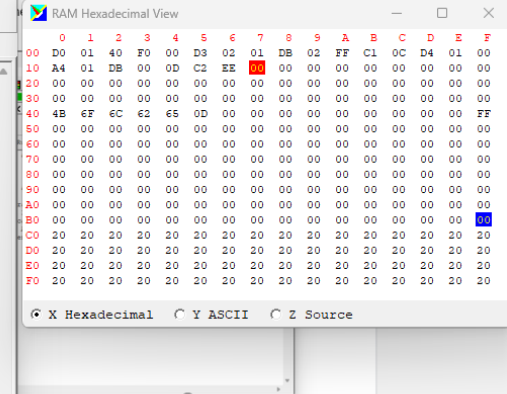
T:\Architecture Labs\Lab5\Lab5.ASM
File Edit View Examples Help
AL 00001101 0D +013 IP 00010111 17 +023 Assemble Slower Continue
BL 01000110 46 +070 SP 10111111 BF -065 Step Faster Cpu Reset
CL 00000000 00 +000 SR 00000010 02 +002 Run F8 STOP Show Ram
DL 00000000 00 +000 IS02

Write Run Log Log Assembler Activity
Source Code List File Configuration Tokens Run Log

MOV BL, 40; Initialize BL to starting address of the array
loop:
    IN 00; input stored in AL
    MOV CL,[BL]
    CMP CL, FF; Compare index with end of array
    JZ end
    MOV [BL], AL; Inputed value stored in location of BL
    INC BL; Increment BL to go to next location in the array
    CMP AL, 0D; CMP input with ASCII for enter
    JNZ loop
end:
    HALT

ORG 40; Declare array
DB 00
DB 00
DB 00
DB 00
DB 00
DB 00
DB 00
DB 00
DB 00
DB 00
DB 00
DB 00
DB 00
DB 00
DB 00
DB FF
END
  
```

where the user can input a number and the program
the ASCII input to an integer and save the integer



that reverses an array. Create two five element
40 and the other at location 50. Initialize the
Have your program place the elements of the
verse order.

2. Write an assembly program where the user can input a number and the program will convert the number from the ASCII input to an integer and save the integer into an array. Make a four-element array at location 50 and input your birth year.

The screenshot shows an x86 assembly editor window titled 'TAArchitecture Labs\Lab5\Lab5.ASM'. The editor contains the following assembly code:

```

AL 00001101 0D +013 IF 00011101 1D +029
BL 01010100 54 +084 SF 10111111 BF -065
CL 11111111 FF -001 SR 00000010 02 +002
DL 00000000 00 +000 ISOZ

loop:
    MOV BL, 50; Initialize BL to starting address of the array
    IN 00; input stored in AL
    MOV CL, [BL]
    CMP CL, FF; Compare index with end of array
    JZ end
    SUB AL, 30; Convert from ASCII by subtracting 48 (30 in hex)
    MOV [BL], AL; Inputed value stored in location of BL
    INC BL; Increment BL to go to next location in the array
    ADD AL, 30; Add 48 (30 in hex) back to AL for comparison for enter key to stop program
    CMP AL, 0D; CMP input with ASCII for enter
    JNZ loop
end:
    HALT

ORG 50; Declare array
DB 00
DB 00
DB 00
DB 00
DB FF
END
  
```

The RAM Hexadecimal View window shows a memory dump starting at address 00. The value at address 50 is 00, and the value at address 54 is 00. The value at address 58 is 00, and the value at address 5C is 00. The value at address 60 is 00, and the value at address 64 is 00. The value at address 68 is 00, and the value at address 6C is 00. The value at address 70 is 00, and the value at address 74 is 00. The value at address 78 is 00, and the value at address 7C is 00. The value at address 80 is 00, and the value at address 84 is 00. The value at address 88 is 00, and the value at address 8C is 00. The value at address 90 is 00, and the value at address 94 is 00. The value at address 98 is 00, and the value at address 9C is 00. The value at address A0 is 00, and the value at address A4 is 00. The value at address A8 is 00, and the value at address AC is 00. The value at address B0 is 00, and the value at address B4 is 00. The value at address B8 is 00, and the value at address BC is 00. The value at address C0 is 20, and the value at address C4 is 20. The value at address C8 is 20, and the value at address CC is 20. The value at address D0 is 20, and the value at address D4 is 20. The value at address D8 is 20, and the value at address DC is 20. The value at address E0 is 20, and the value at address E4 is 20. The value at address E8 is 20, and the value at address EC is 20. The value at address F0 is 20, and the value at address F4 is 20. The value at address F8 is 20, and the value at address FC is 20.

[illegible]

The screenshot displays the TASM (Turbo Assembler) interface. The title bar reads "T:\Architecture Labs\Lab5\Lab5.ASM". The menu bar includes "File", "Edit", "View", "Examples", and "Help".

The main window is divided into several sections:

- Registers and Memory:** A table showing the state of registers and memory locations.

| | | | | | | | | |
|---------------------|---------------------|--|--|--|--|----------|--------|-----------|
| AL 01000110 46 +070 | IP 00010110 16 +022 | | | | | Assemble | Slower | Continue |
| BL 01001110 4E +078 | SP 10111111 BF -065 | | | | | Step | Faster | Cpu Reset |
| CL 11111111 FF -001 | SR 00000010 02 +002 | | | | | Run F9 | STOP | Show Ram |
| DL 00000000 00 +000 | IS02 | | | | | | | |
- Status Bar:** A yellow bar at the top of the main window displays the message: "END Program has halted." Below it, there are checkboxes for "Write Run Log" and "Log Assembler Activity".
- Source Code:** A tabbed interface with "Source Code" selected. The code is as follows:


```

MOV AL, 40; Initialize BL to starting address of the first array
MOV BL, 54; Initialize CL to ending address of the second array

loop:
MOV CL, [AL]
MOV [BL], CL; Copy value in location of AL into end of BL
INC AL; Increment AL to go to next location in the array
SUB BL, 1; Decrement BL to go to next location in the array
CMP CL, FF; Compare index with end of array
JNZ loop
HALT

ORG 40; Declare first array
DB 05
DB 04
DB 03
DB 02
DB 01
DB FF

ORG 50; Declare Second array
DB 00
DB 00
DB 00
DB 00
DB 00
DB 00
DB FF
END
      
```