

*Alexandria University,*  
*Faculty of Engineering,*  
*Computer and Systems Engineering Dept.*  
*CS222: System programming*

## **Lab (2) Report**

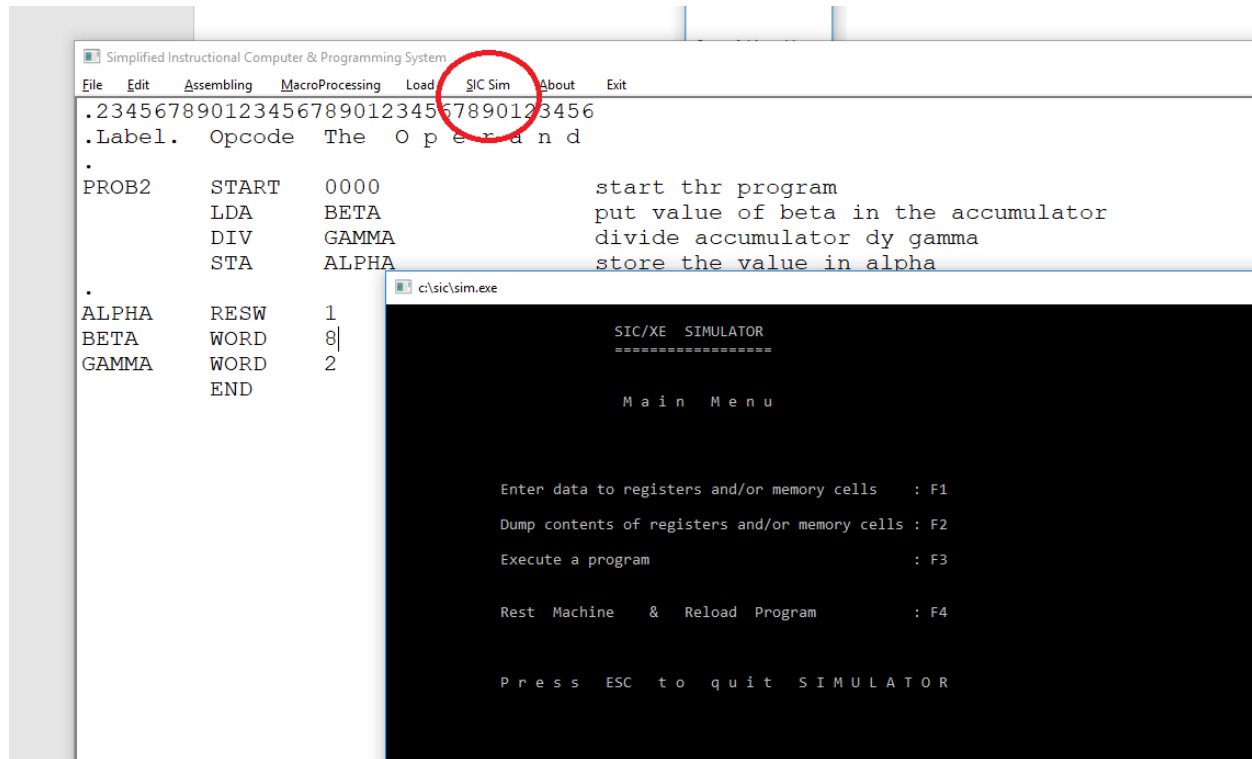
**Name:**

Abdel-Fatah Mohamed Abdel-Fatah. (24)

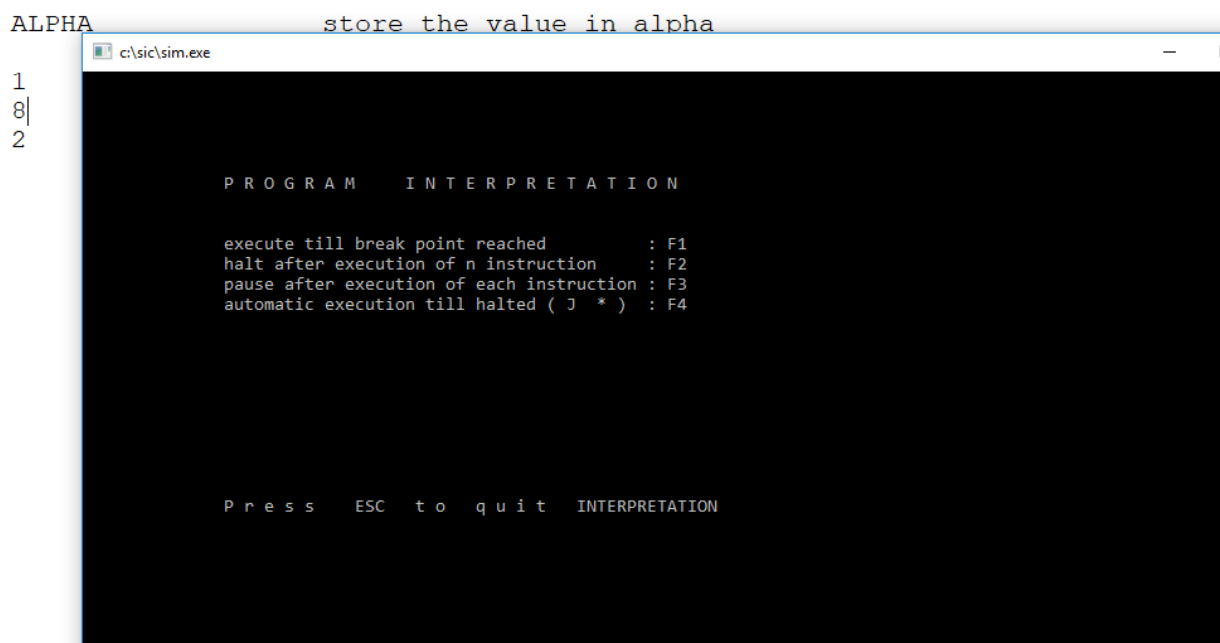
## Content:

- 1) Simulator.
- 2) PROGRAMS.
  - a. Write a program that searches for a certain byte in a string, if found, put the address of the byte in register A else, put 0xFFFFF.
  - b. Read a string from an input device and print it reversed to an output device.
  - c. Read a string from an input device and print the string to an output device after converting it to UPPER case.
  - d. Implement the bubble sort to sort characters in a string.

\*\*\*Simulator:  
(before all examples).



a. Execute a program (F3).



b. Automatic execution till halted (F4):

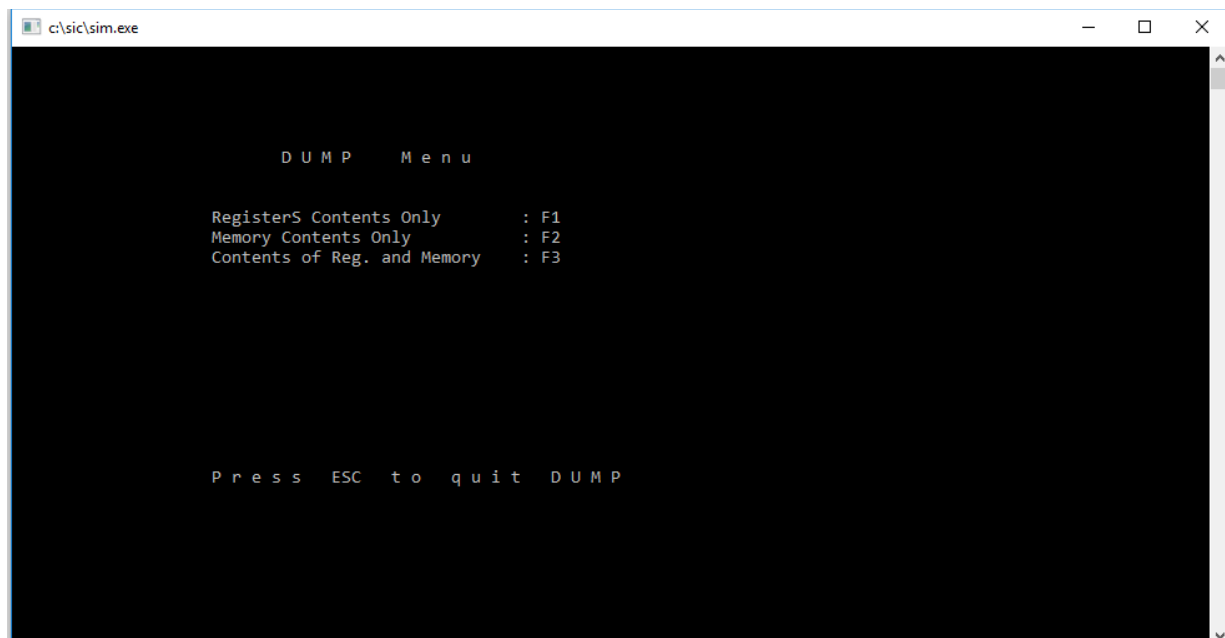
And press any button to execute instruction till the end.



C. Dump contents of registers and/or memory cells (F2):

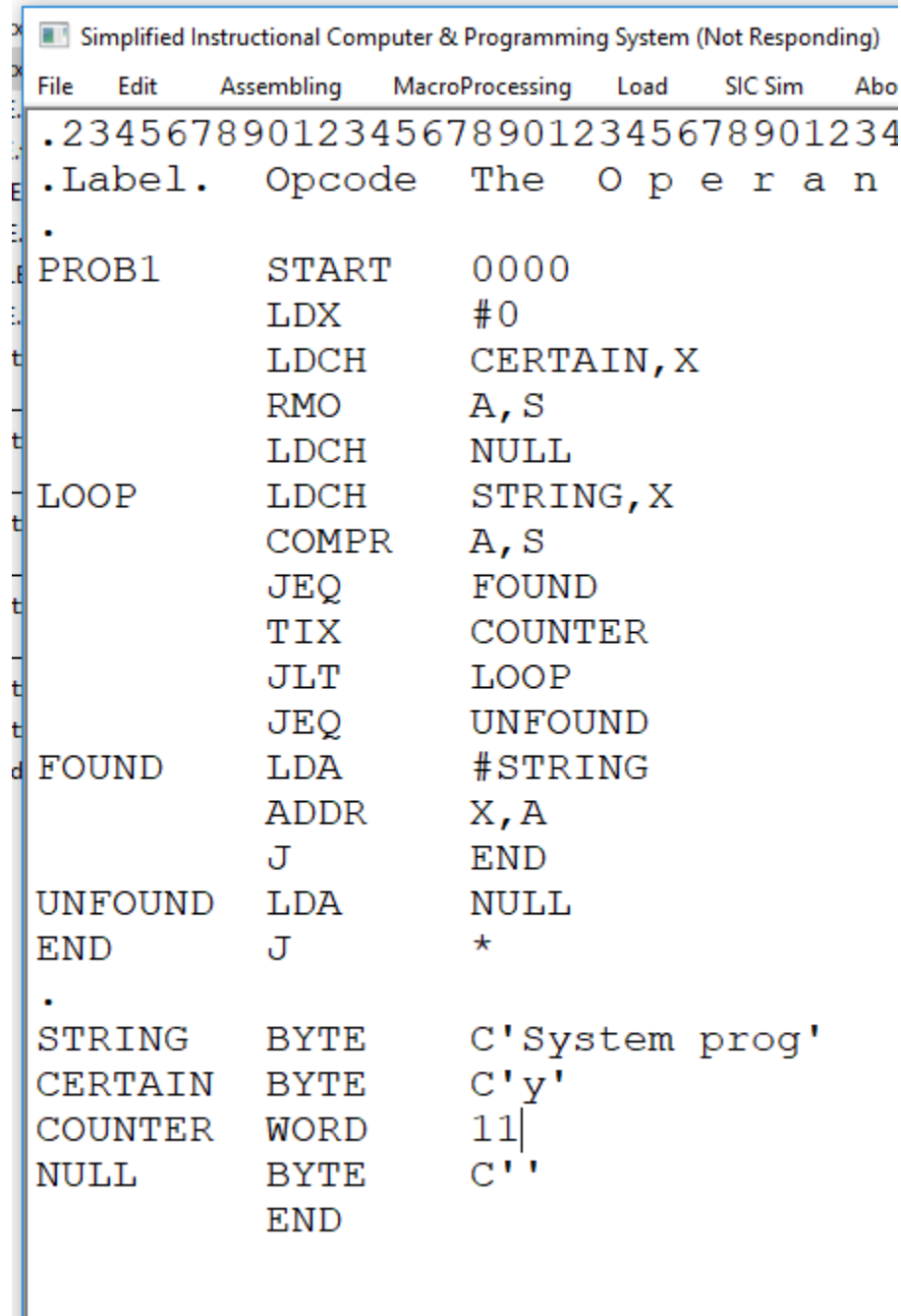


## D. Contents of Reg. and Memory (F3):



- a. Write a program that searches for a certain byte in a string, if found, put the address of the byte in register A else, put 0xFFFF:

### 1) Code:

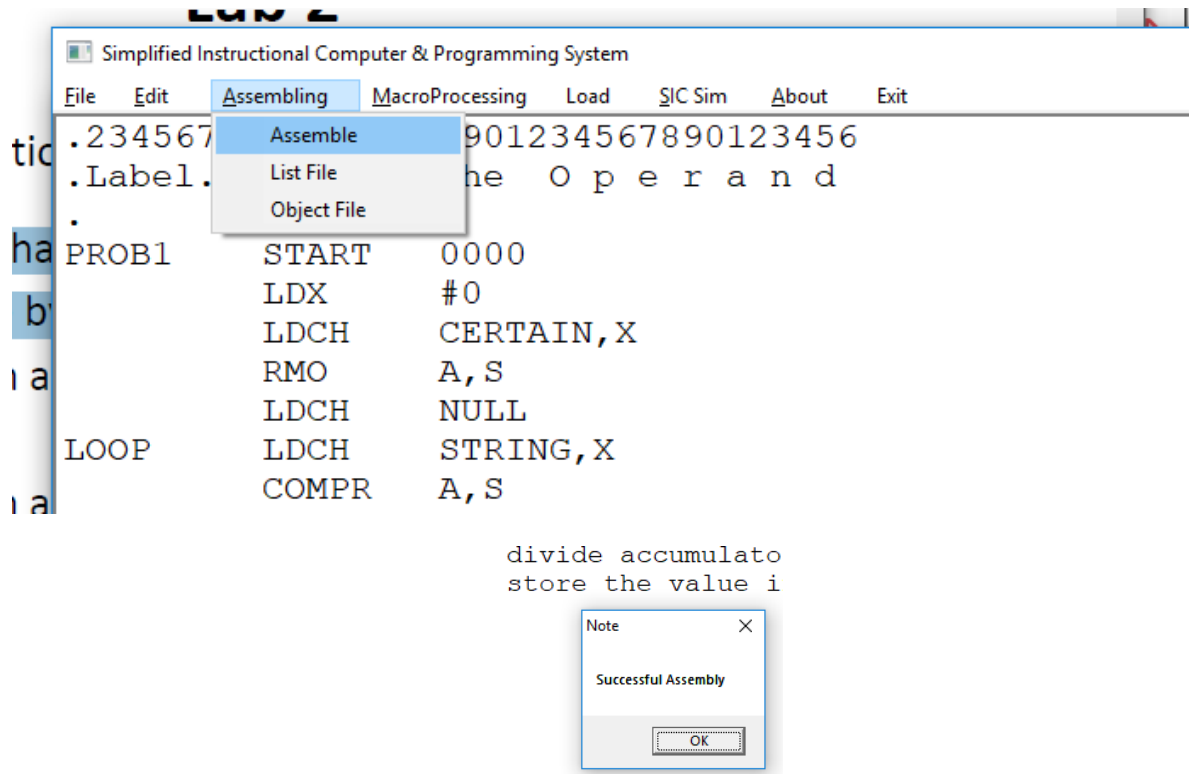


The screenshot shows the SIC interface with a menu bar (File, Edit, Assembling, MacroProcessing, Load, SIC Sim, Abo) and a text area containing the following assembly code:

```

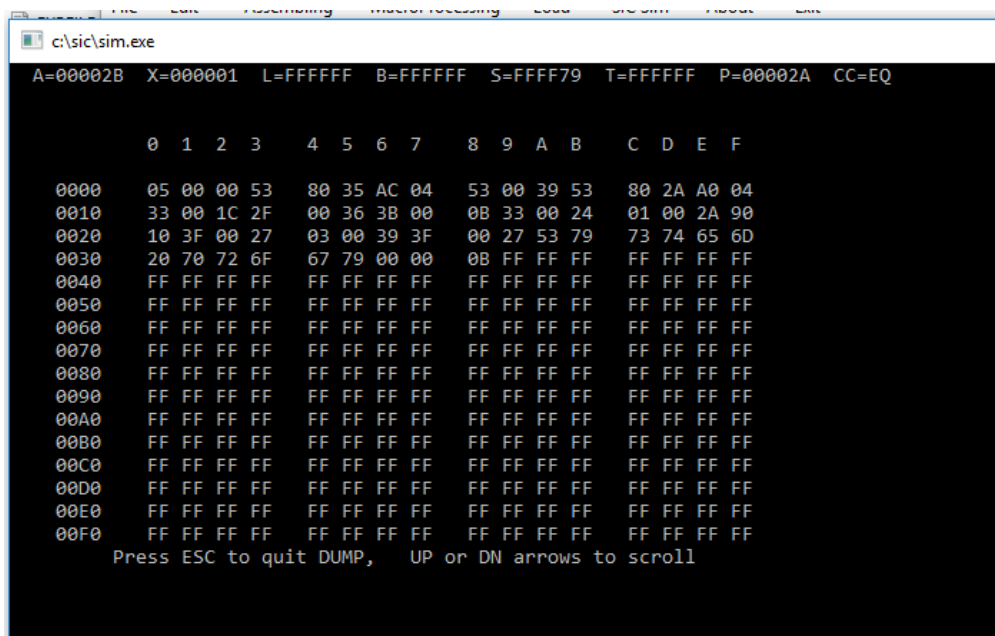
.234567890123456789012345678901234
.Label.  Opcode  The  O p e r a n
.
PROB1    START    0000
          LDX      #0
          LDCH     CERTAIN,X
          RMO      A,S
          LDCH     NULL
LOOP     LDCH     STRING,X
          COMPR    A,S
          JEQ      FOUND
          TIX      COUNTER
          JLT      LOOP
          JEQ      UNFOUND
FOUND    LDA      #STRING
          ADDR     X,A
          J        END
UNFOUND  LDA      NULL
END      J        *
.
STRING   BYTE     C'System prog'
CERTAIN  BYTE     C'y'
COUNTER  WORD     11|
NULL     BYTE     C''
          END
  
```

## 2) Assembling:

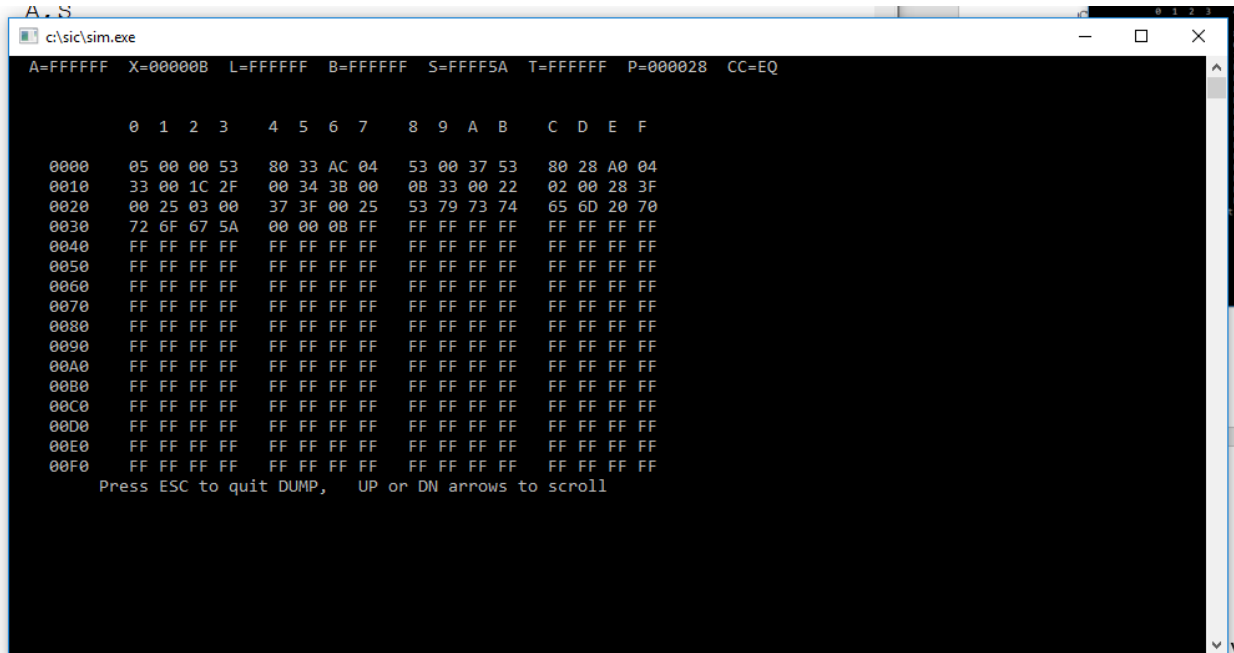


## 3) Sample run:

String = "System prog ", Certain = 'y',



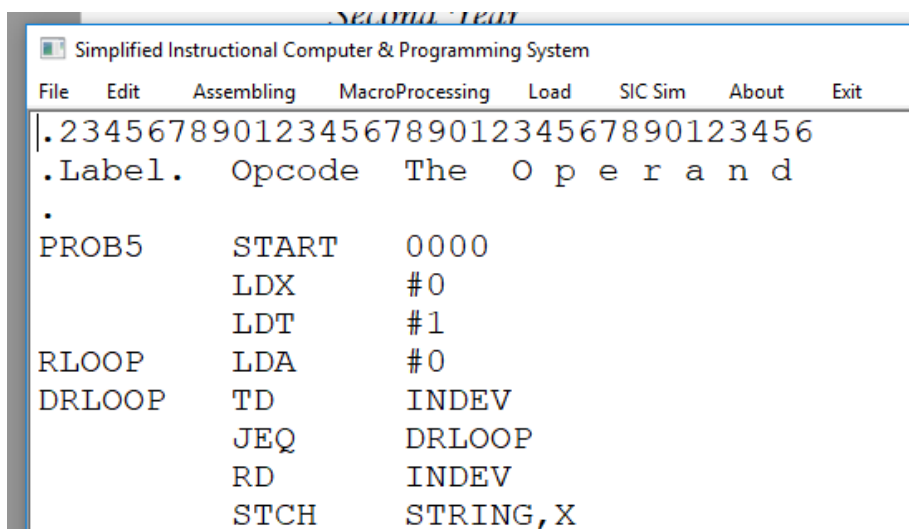
String = "System prog", Certain = 'Z',



```
A.S.  
c:\sic\sim.exe  
A=FFFFFF X=00000B L=FFFFFF B=FFFFFF S=FFFF5A T=FFFFFF P=000028 CC=EQ  
  
  0  1  2  3  4  5  6  7  8  9  A  B  C  D  E  F  
0000  05 00 00 53 80 33 AC 04 53 00 37 53 80 28 A0 04  
0010  33 00 1C 2F 00 34 3B 00 0B 33 00 22 02 00 28 3F  
0020  00 25 03 00 37 3F 00 25 53 79 73 74 65 6D 20 70  
0030  72 6F 67 5A 00 00 0B FF FF FF FF FF FF FF FF  
0040  FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF  
0050  FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF  
0060  FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF  
0070  FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF  
0080  FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF  
0090  FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF  
00A0  FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF  
00B0  FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF  
00C0  FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF  
00D0  FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF  
00E0  FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF  
00F0  FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF  
Press ESC to quit DUMP, UP or DN arrows to scroll
```

b. Read a string from an input device and print it reversed to an output device:

1) Code:



```
Simplified Instructional Computer & Programming System  
File Edit Assembling MacroProcessing Load SIC Sim About Exit  
0123456789012345678901234567890123456  
.Label. Opcode The Operand  
.  
PROB5 START 0000  
LDX #0  
LDT #1  
RLOOP LDA #0  
DRLOOP TD INDEV  
JEQ DRLOOP  
RD INDEV  
STCH STRING,X
```



```

        COMP      #4
        ADDR      T,X
        JGT        RLOOP
        JLT        RLOOP
        SUBR       T,X
        SUBR       T,X
        LDS        #0
WLOOP   LDA        #0
DWLOOP  TD         OUTDEV
        JEQ        DWLOOP
        LDCH       STRING,X
        WD         OUTDEV
        SUBR       T,X
        COMPR      S,X
        JLT        DWLOOP
        JEQ        DWLOOP
        J          *

        .
INDEV   BYTE       X'F1'
OUTDEV  BYTE       X'04'
STRING  RESB       100
        END

```

C:\SIC\files\prog\_2\_2.txt

## 2) Assembling:

0000

```

#0
#1
#0
INI |.234567
DRI |.Label.
INI |.
STF PROB5      START 0000
#4             LDX    #0
T,X            LDT    #1
RLO RLOOP      LDA    #0
-- DWLOOP      TD     INDEV

```

Simplified Instructional Computer & Programming System

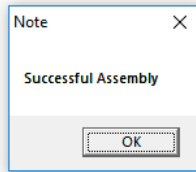
File Edit **Assembling** MacroProcessing Load SIC Sim About Exit

Assemble  
List File  
Object File

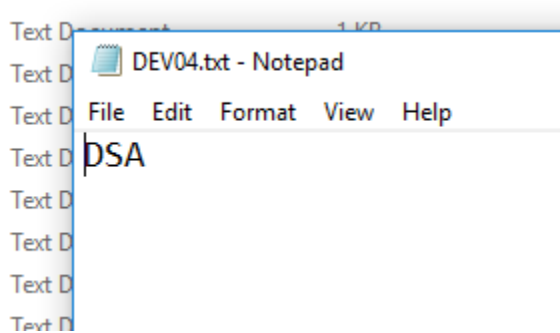
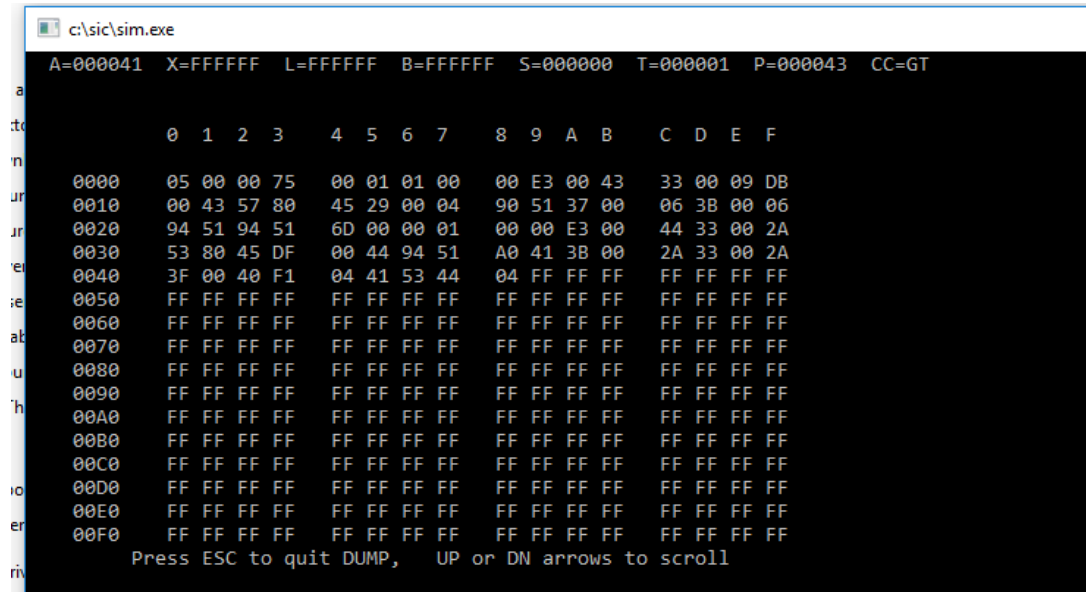
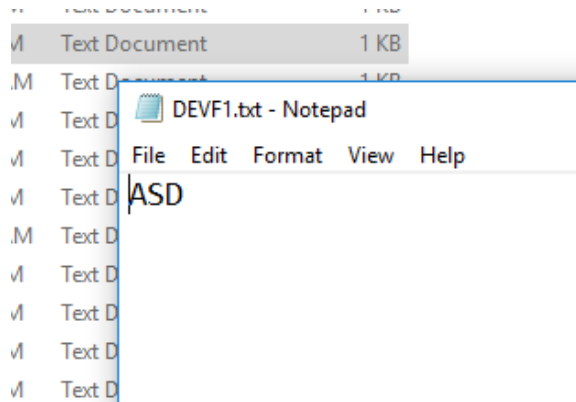
901234567890123456

the Operand

```
divide accumulator
store the value i
```



### 3) Sample run:



xt D  
xt D  
xt D  
xt D  
xt D  
xt D  
xt D

DEVF1.txt - Notepad

File Edit Format View Help

System programming

c:\sic\sim.exe

A=000053 X=FFFFFF L=FFFFFF B=FFFFFF S=000000 T=000001 P=000043 CC=GT

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0000	05	00	00	75	00	01	01	00	00	E3	00	43	33	00	09	DB
0010	00	43	57	80	45	29	00	04	90	51	37	00	06	3B	00	06
0020	94	51	94	51	6D	00	00	01	00	00	E3	00	44	33	00	2A
0030	53	80	45	DF	00	44	94	51	A0	41	3B	00	2A	33	00	2A
0040	3F	00	40	F1	04	53	79	73	74	65	6D	20	70	72	6F	67
0050	72	61	6D	6D	69	6E	67	04	FF	FF	FF	FF	FF	FF	FF	FF
0060	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
0070	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
0080	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
0090	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
00A0	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
00B0	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
00C0	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
00D0	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
00E0	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
00F0	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF

Press ESC to quit DUMP, UP or DN arrows to scroll

DEV04.txt - Notepad

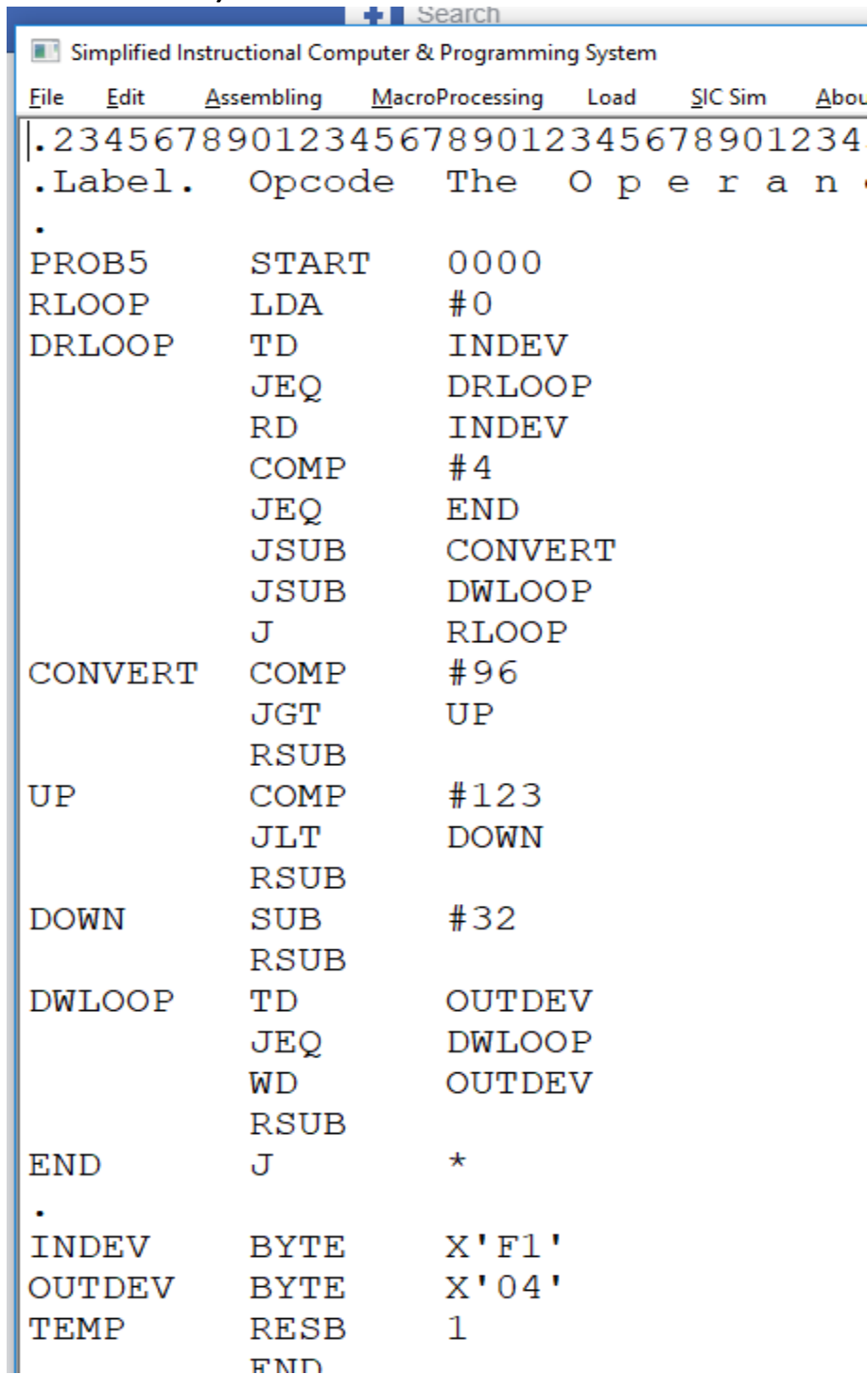
File Edit Format View Help

gnimmargorp metsys

0  
0  
0  
0  
6  
F  
F

c. Read a string from an input device and print the string to an output device after converting it to UPPER case:

1) Code:

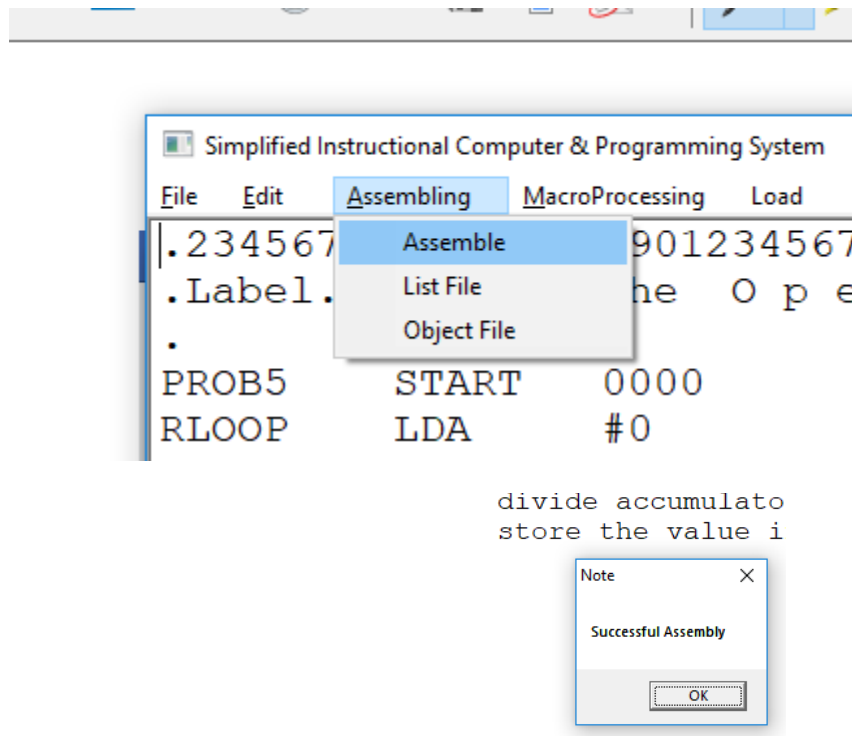


The screenshot shows the SIC editor window with the following assembly code:

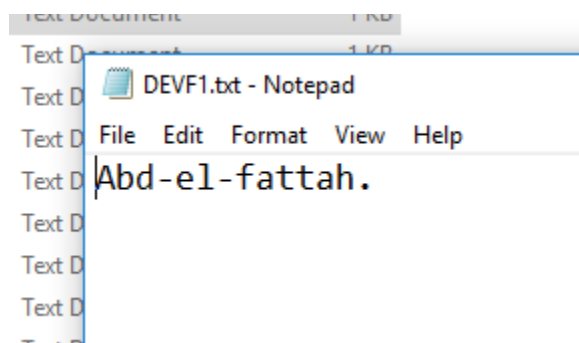
```

.234567890123456789012345678901234
.Label.  Opcode  The  O p e r a n d
.
PROB5    START    0000
RLOOP    LDA      #0
DRLOOP    TD      INDEV
          JEQ      DRLOOP
          RD      INDEV
          COMP     #4
          JEQ      END
          JSUB     CONVERT
          JSUB     DWLOOP
          J        RLOOP
CONVERT   COMP     #96
          JGT      UP
          RSUB
UP         COMP     #123
          JLT      DOWN
          RSUB
DOWN      SUB      #32
          RSUB
DWLOOP    TD      OUTDEV
          JEQ      DWLOOP
          WD      OUTDEV
          RSUB
END       J        *
.
INDEV     BYTE     X'F1'
OUTDEV    BYTE     X'04'
TEMP      RESB     1
          END
  
```

## 2) Assembling:



## 3) Sample run:





## D.Implement the bubble sort to sort characters in a string:

### 1) Code:

```
Simplified Instructional Computer & Programming System
File Edit Assembling MacroProcessing Load SIC Sim About Exit
|.23456789012345678901234567890123456
.Label. Opcode The Operand
.
PROB1 START 0000
      LDX #0
      LDT LENGTH
LOOP  LDCH STRING,X
      LDS NULL
      RMO A,S
      LDCH NULL
      TIX LENGTH
      COMPR X,T
      JLT F
      JSUB FUNC
      J LOOP
F     LDCH STRING,X
      COMPR A,S
      JLT SWAP
      J LOOP
SWAP  STCH TEMP
      RMO S,A
      STCH STRING,X
      LDA #1
      SUBR A,X
      LDA NULL
      LDCH TEMP
      STCH STRING,X
      LDA #1
      ADDR A,X
      LDA NULL
      J LOOP
FUNC  LDX #1
      SUBR X,T
      LDX #0
      COMPR X,T
      JEQ END
      RSUB
      LDX #0
      LDT #1
      LDA #0
END   TD OUTDEV
DWLOOP JEQ DWLOOP
```

```

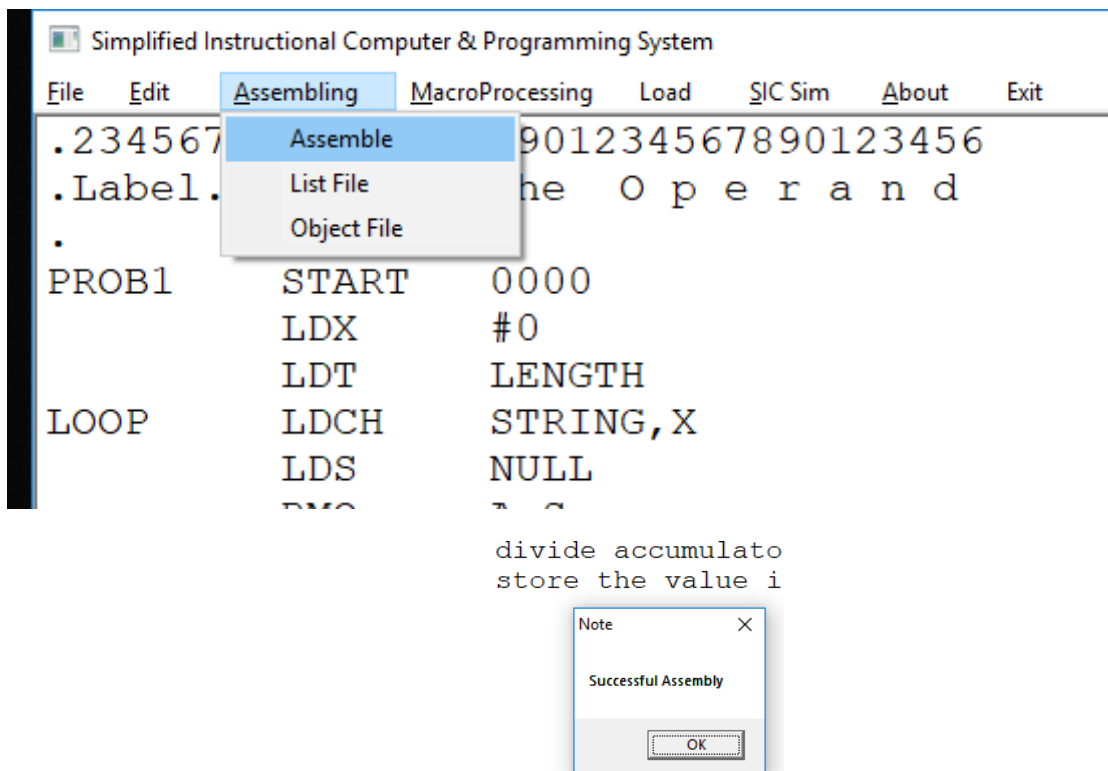
END      LDA      #0
DWLOOP   TD       OUTDEV
         JEQ      DWLOOP
         LDCH     STRING,X
         WD       OUTDEV
         TIX      LENGTH
         JLT      DWLOOP
         J        *

.
OUTDEV   BYTE     X'04'
STRING   BYTE     C'ECDBFA'
TEMP     RESB     1
LENGTH   WORD     6
NULL     BYTE     C''
         END

```

< C:\SIC\files\prog\_4\_2.txt

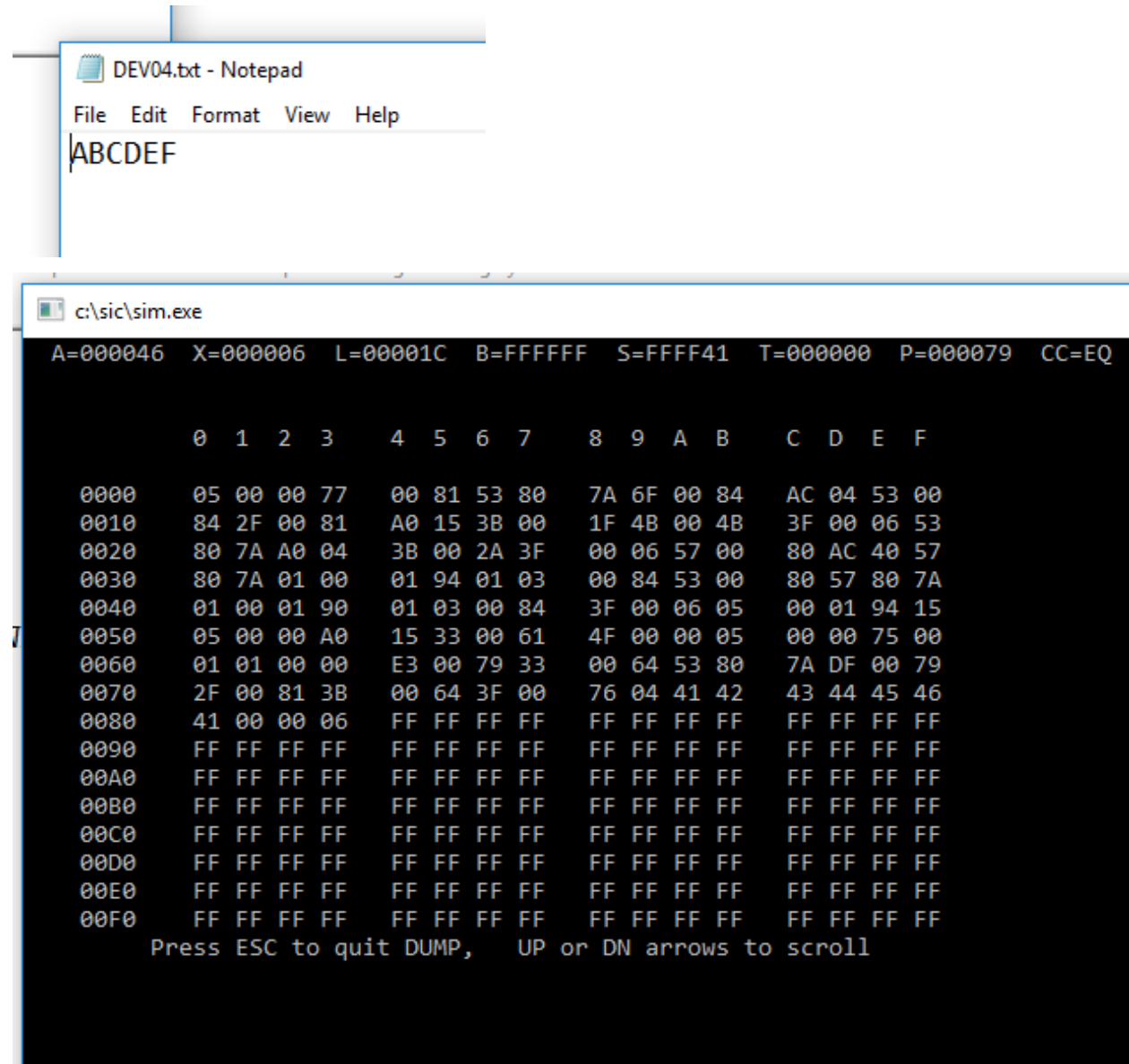
## 2) Assembling:





### 3) Sample run:

String = "ECDABF "      output = "ABCDEF"



The image shows two windows from a Windows operating system. The top window is a Notepad application titled "DEV04.txt - Notepad". It has a menu bar with "File", "Edit", "Format", "View", and "Help". The text "ABCDEF" is entered in the main editing area. The bottom window is a command prompt titled "c:\sic\sim.exe". It displays a memory dump with various registers at the top: A=000046, X=000006, L=00001C, B=FFFFFF, S=FFFF41, T=000000, P=000079, and CC=EQ. Below this is a table of memory addresses and their corresponding hexadecimal values. The addresses range from 0000 to 00F0 in increments of 10. The values are displayed in a grid format. At the bottom of the command prompt, it says "Press ESC to quit DUMP, UP or DN arrows to scroll".

```
A=000046 X=000006 L=00001C B=FFFFFF S=FFFF41 T=000000 P=000079 CC=EQ
```

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0000	05	00	00	77	00	81	53	80	7A	6F	00	84	AC	04	53	00
0010	84	2F	00	81	A0	15	3B	00	1F	4B	00	4B	3F	00	06	53
0020	80	7A	A0	04	3B	00	2A	3F	00	06	57	00	80	AC	40	57
0030	80	7A	01	00	01	94	01	03	00	84	53	00	80	57	80	7A
0040	01	00	01	90	01	03	00	84	3F	00	06	05	00	01	94	15
0050	05	00	00	A0	15	33	00	61	4F	00	00	05	00	00	75	00
0060	01	01	00	00	E3	00	79	33	00	64	53	80	7A	DF	00	79
0070	2F	00	81	3B	00	64	3F	00	76	04	41	42	43	44	45	46
0080	41	00	00	06	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
0090	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
00A0	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
00B0	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
00C0	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
00D0	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
00E0	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
00F0	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF

Press ESC to quit DUMP, UP or DN arrows to scroll