MARCH 23, 2018

LAB EXERCISE 3

PROF. JIMMY MATHEW | CSE 2006 MPI B1 SLOT | 16BCE0783 DAKSH

Question 1:

A password requires 4 alphabets and two numbers. It should start with a number. Number of characters must be six. Read a password string from the user using interrupt, and check if it is a valid password combination. Print "Yes, valid" or "No, Not valid" on the screen.

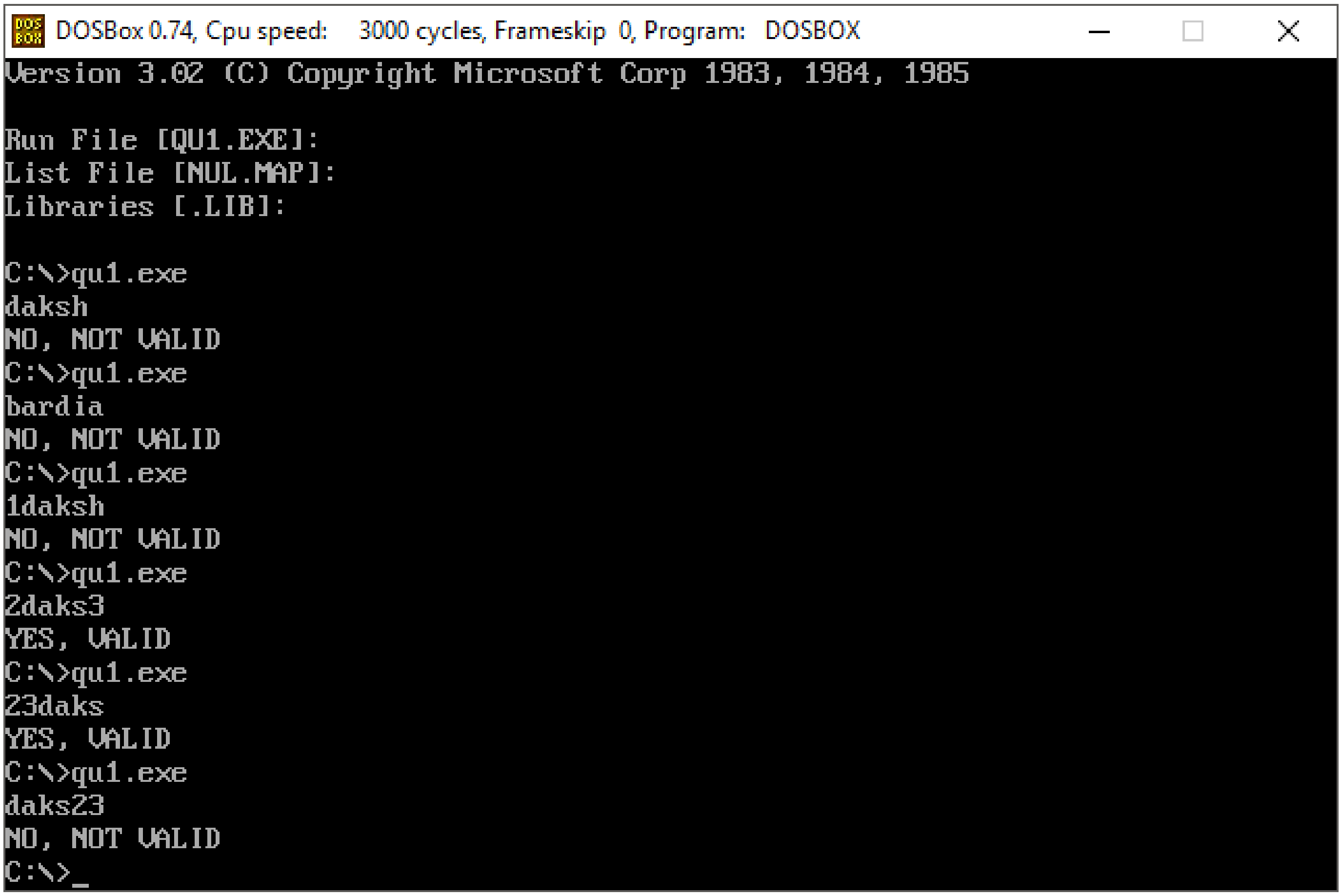
Algorithm:

Just checking for the three conditions one by one.

Code:

```
.MODEL SMALL

.STACK 64

.DATA

        PASS DB 100 DUP('$')

        SUCCESS DB 'YES, VALID$'

        FAIL DB 'NO, NOT VALID$'

        LEN DB 6

.CODE


START:
        MOV AX, @DATA
        MOV DS, AX

        MOV SI, OFFSET PASS

L1:
        MOV AH, 01H
        INT 21H
        CMP AL, 13
        JE IPTAKEN
        MOV [SI], AL
        INC SI
        JMP L1

IPTAKEN:

        ;MOV DX, OFFSET PASS
        ;MOV AH, 09H
        ;INT 21H

        MOV BL, 00H
        LEA SI, PASS

CHECKLEN:
        MOV AL, '$'
        CMP [SI], AL
        JE CHECKLEN2
        INC SI
        ADD BL, 01H
        JMP CHECKLEN

CHECKLEN2:
        CMP BL, 06H
        JNE FMSG

CHECKNUMBER:
          LEA SI, PASS
          MOV DL, [SI]
          MOV AL, 031H
          CMP DL, AL
          JL FMSG
          MOV AL, 039H
          CMP DL, AL
          JG FMSG

          MOV CH, 00H
          MOV CL, 00H
          LEA SI, PASS

CHECKNUM:
          MOV DL, [SI]
          MOV AL, '$'
          CMP DL, AL
          JE FINALCHECK
          MOV AL, 039H
          CMP DL, AL
          JLE NUMNUM
          CMP DL, AL
          JG NUMALPHA

NUMALPHA:
          ADD CH, 01H
          INC SI
          JMP CHECKNUM

NUMNUM:
          ADD CL, 01H
          INC SI
          JMP CHECKNUM

FINALCHECK:
          CMP CH, 04H
          JNE FMSG
          CMP CL, 02H
          JNE FMSG

SMSG:
        MOV DX, OFFSET SUCCESS
        MOV AH, 09H

        INT 21H


        MOV AH, 4CH

        INT 21H


FMSG:

        MOV DX, OFFSET FAIL

        MOV AH, 09H

        INT 21H


        MOV AH, 4CH

        INT 21H


END START

.END
```

Screenshots:



DOSBox 0.74, Cpu speed: 3000 cycles, Frameskip 0, Program: DOSBOX

```
Version 3.02 (C) Copyright Microsoft Corp 1983, 1984, 1985

Run File [QU1.EXE]:
List File [NUL.MAP]:
Libraries [.LIB]:

C:\>qu1.exe
daksh
NO, NOT VALID
C:\>qu1.exe
bardia
NO, NOT VALID
C:\>qu1.exe
1daksh
NO, NOT VALID
C:\>qu1.exe
2daks3
YES, VALID
C:\>qu1.exe
23daks
YES, VALID
C:\>qu1.exe
daks23
NO, NOT VALID
C:\>_
```

Question 2:

Prompt the user for name, age, reg. no from the keyboard and display it on the screen. You must clear the entire screen before the display.

Algorithm:

Just Input and Output. Screen is cleared using INT 10H and 0003H in AX.

Code:

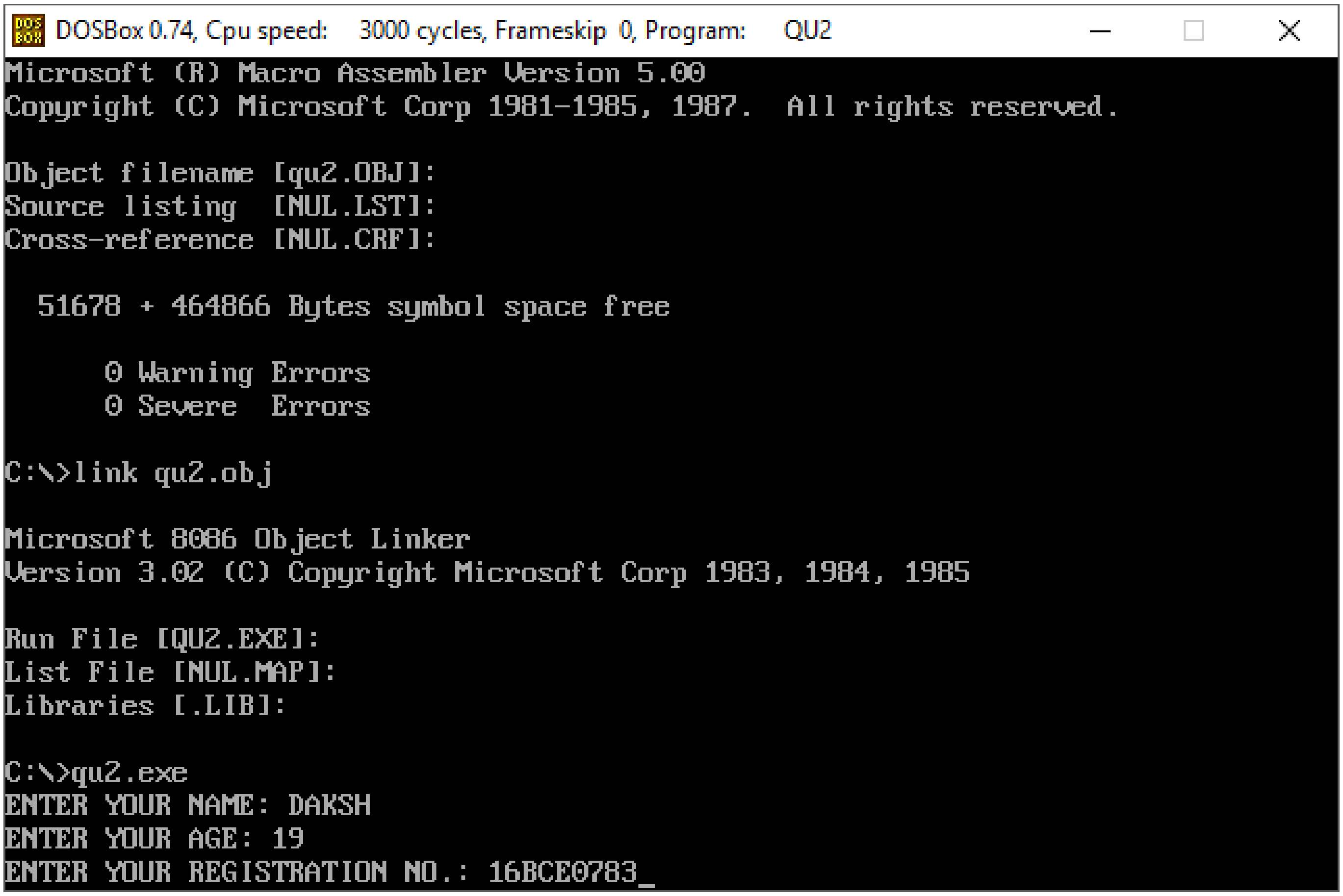
```
.MODEL SMALL

.STACK 64

.DATA


        Q_NAME DB 'ENTER YOUR NAME: $'

        Q_AGE DB 'ENTER YOUR AGE: $'

        Q_REGNO DB 'ENTER YOUR REGISTRATION NO.: $'


        NME DB 20 DUP('$')

        AGE DB 20 DUP('$')

        REGNO DB 20 DUP('$')


        A_NAME DB 'YOUR NAME IS: $'

        A_AGE DB 'YOUR AGE IS: $'

        A_REGNO DB 'YOUR REGISTRATION NO. IS: $'


.CODE


START:

        MOV AX, @DATA

        MOV DS, AX


          MOV DX, OFFSET Q_NAME

        MOV AH, 09H
```

```
    INT 21H

     MOV SI, OFFSET NME

     ;13 IS THE CODE FOR ENTER KEY
GETNAME:
    MOV AH, 01H
    INT 21H
    CMP AL, 13
    JE GETAGE
    MOV [SI], AL
    INC SI
    JMP GETNAME

GETAGE:
     MOV DX, OFFSET Q_AGE
    MOV AH, 09H
    INT 21H
     MOV SI, OFFSET AGE
GETAGE2:
    MOV AH, 01H
    INT 21H
    CMP AL, 13
    JE GETREGNO
    MOV [SI], AL
    INC SI
```

```
        JMP GETAGE2


GETREGNO:

          MOV DX, OFFSET Q_REGNO

        MOV AH, 09H

        INT 21H

          MOV SI, OFFSET REGNO

GETREGNO2:

        MOV AH, 01H

        INT 21H

        CMP AL, 13

        JE PROGRAMEND

        MOV [SI], AL

        INC SI

        JMP GETREGNO2


PROGRAMEND:


          ; CLEARING THE SCREEN

        MOV AX, 0003H

        INT 10H


        MOV DX, OFFSET A_NAME

        MOV AH, 09H

        INT 21H

          MOV DX, OFFSET NME
        MOV AH, 09H
        INT 21H
          ; NEW LINE
          MOV DL,   13
          MOV AH,   02H
          INT 21H
          MOV DL,   10
          MOV AH,   02H
          INT 21H

          MOV DX, OFFSET A_AGE
        MOV AH, 09H
        INT 21H
          MOV DX, OFFSET AGE
        MOV AH, 09H
        INT 21H
          MOV DL,   13
          MOV AH,   02H
          INT 21H
          MOV DL,   10
          MOV AH,   02H
          INT 21H

          MOV DX, OFFSET A_REGNO
        MOV AH, 09H
        INT 21H
```

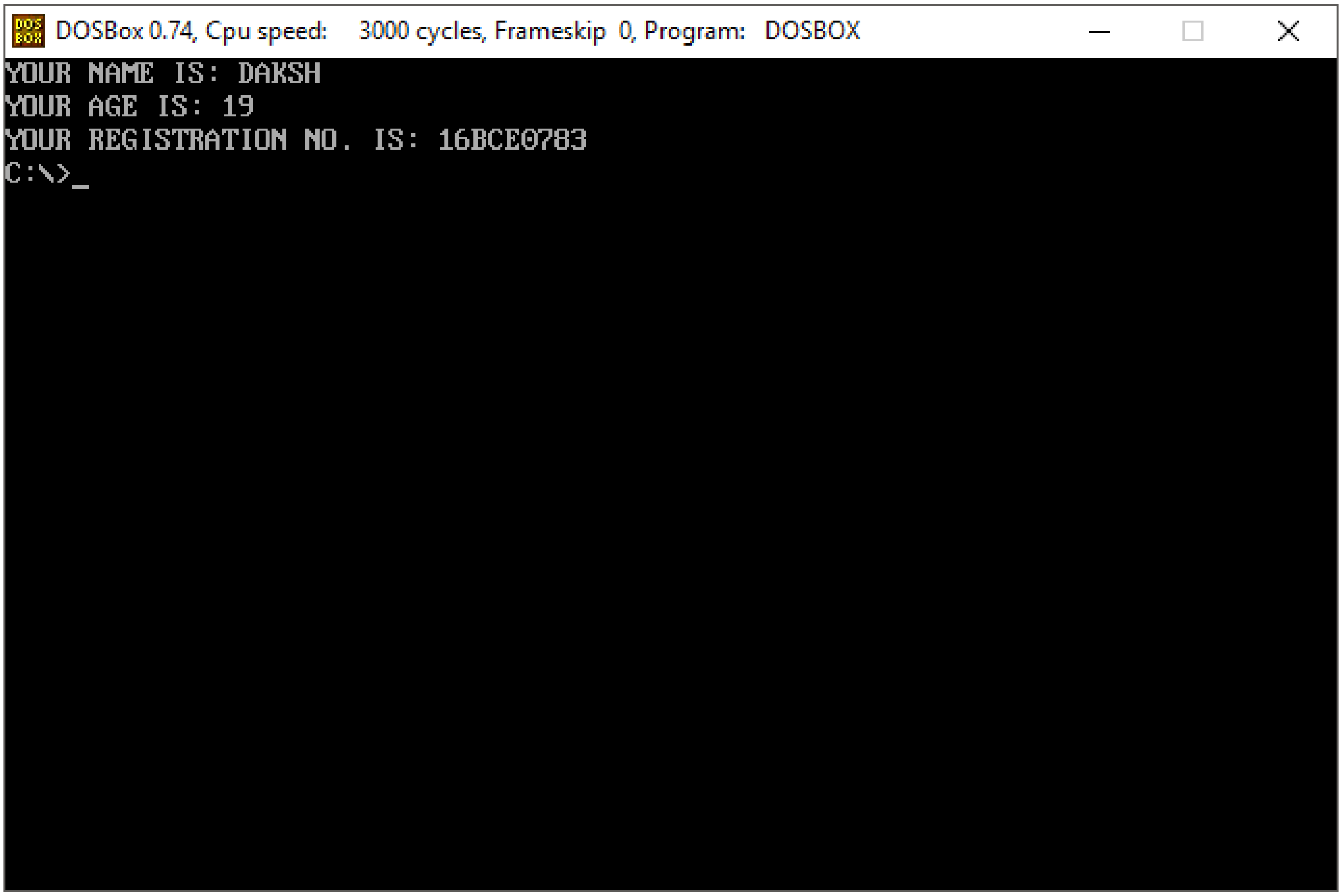
```
          MOV DX, OFFSET REGNO

        MOV AH, 09H

        INT 21H


        MOV AH, 4CH

        INT 21H


END START

.END
```

Screenshots:



DOSBox 0.74, Cpu speed: 3000 cycles, Frameskip 0, Program: QU2

```
Microsoft (R) Macro Assembler Version 5.00
Copyright (C) Microsoft Corp 1981-1985, 1987.  All rights reserved.

Object filename [qu2.OBJ]:
Source listing  [NUL.LST]:
Cross-reference [NUL.CRF]:

  51678 + 464866 Bytes symbol space free

      0 Warning Errors
      0 Severe  Errors

C:\>link qu2.obj

Microsoft 8086 Object Linker
Version 3.02 (C) Copyright Microsoft Corp 1983, 1984, 1985

Run File [QU2.EXE]:
List File [NUL.MAP]:
Libraries [.LIB]:

C:\>qu2.exe
ENTER YOUR NAME: DAKSH
ENTER YOUR AGE: 19
ENTER YOUR REGISTRATION NO.: 16BCE0783_
```


DOSBox 0.74, Cpu speed: 3000 cycles, Frameskip 0, Program: DOSBOX

```
YOUR NAME IS: DAKSH
YOUR AGE IS: 19
YOUR REGISTRATION NO. IS: 16BCE0783
C:\>_
```

Question 3:

Read a string from user using interrupts. Use Caesar's cypher to encrypt the given string. Encryption key (displacement) should be read from the user. Display both encrypted and decrypted string.

Algorithm:

Caesar's Encryption Algorithm (Addition).

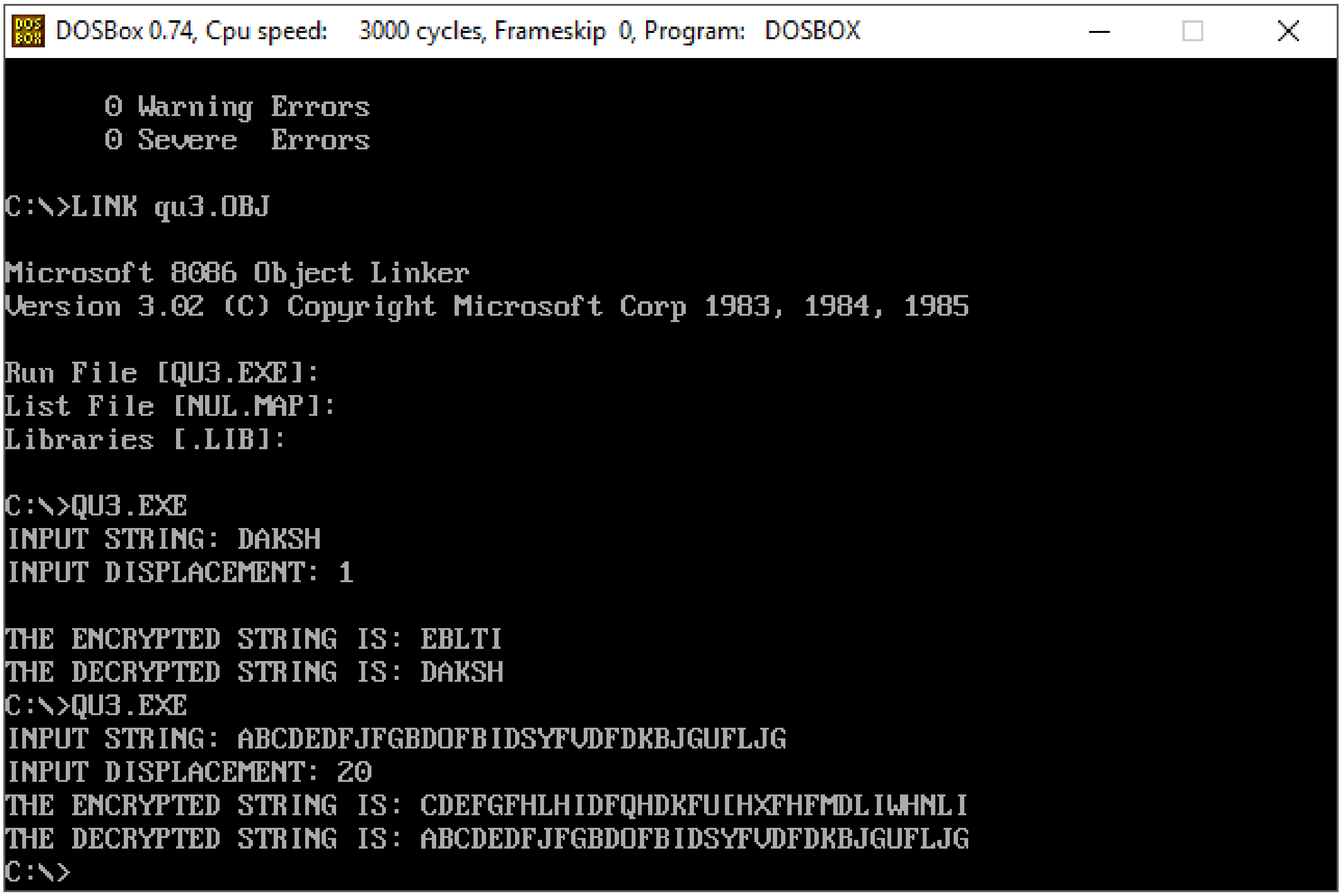
Code:

```
.MODEL SMALL

.STACK 64
```

```
.DATA

        MSGSTR DB 'INPUT STRING: $'

        MSGDIS DB 'INPUT DISPLACEMENT: $'

        MSG1 DB 'THE ENCRYPTED STRING IS: $'

        MSG2 DB 'THE DECRYPTED STRING IS: $'

        STR DB 100 DUP('$')

        ENCSTR DB 100 DUP('$')

        DECSTR DB 100 DUP('$')

        DISPLACE DB 00H

.CODE


START:

        MOV AX, @DATA

        MOV DS, AX


        MOV DX, OFFSET MSGSTR

        MOV AH, 09H

        INT 21H

        MOV SI, OFFSET STR


L1:

        MOV AH, 01H

        INT 21H

        CMP AL, 13

        JE TAKEDISPLACE

        MOV [SI], AL
        INC SI
        JMP L1

TAKEDISPLACE:
        MOV DX, OFFSET MSGDIS
        MOV AH, 09H
        INT 21H
        MOV AL, 00H
        MOV AH, 01H
        INT 21H
        MOV [DISPLACE], AL
        LEA SI, STR
        LEA DI, ENCSTR
        MOV AL, [DISPLACE]
        SUB AL, 48

NEXT:
        MOV DL, [SI]
        CMP DL, '$'
        JE DECRYPT
        ADD DL, AL
        MOV [DI], DL
        INC SI
        INC DI
        JMP NEXT
```

```
DECRYPT:
          LEA SI, ENCSTR
        LEA DI, DECSTR
        MOV AL, [DISPLACE]
        SUB AL, 48

NEXT2:
          MOV DL, [SI]
        CMP DL, '$'
        JE PROGRAMEND
        SUB DL, AL
        MOV [DI], DL
        INC SI
        INC DI
        JMP NEXT2

PROGRAMEND:


        MOV DL, 13
        INT 21H
        MOV DL, 10
        INT 21H

        MOV DX, OFFSET MSG1
        MOV AH, 09H
        INT 21H
        MOV DX, OFFSET ENCSTR

        MOV AH, 09H

        INT 21H

        MOV DL, 13

        MOV AH, 02H

        INT 21H

        MOV DL, 10

        MOV AH, 02H

        INT 21H


        MOV DX, OFFSET MSG2

        MOV AH, 09H

        INT 21H

        MOV DX, OFFSET DECSTR

        MOV AH, 09H

        INT 21H


        MOV AH, 4CH

        INT 21H


END START

.END
```

Screenshots:


DOSBox 0.74, Cpu speed: 3000 cycles, Frameskip 0, Program: DOSBOX

```
      0 Warning Errors
      0 Severe  Errors

C:\>LINK qu3.OBJ

Microsoft 8086 Object Linker
Version 3.02 (C) Copyright Microsoft Corp 1983, 1984, 1985

Run File [QU3.EXE]:
List File [NUL.MAP]:
Libraries [.LIB]:

C:\>QU3.EXE
INPUT STRING: DAKSH
INPUT DISPLACEMENT: 1

THE ENCRYPTED STRING IS: EBLTI
THE DECRYPTED STRING IS: DAKSH
C:\>QU3.EXE
INPUT STRING: ABCDEDFJFGBDOFBIDSYFVDFDKBJGUFLJG
INPUT DISPLACEMENT: 20
THE ENCRYPTED STRING IS: CDEFGFHLHIDFQHDKFU[HXFHFMDLIWHNLI
THE DECRYPTED STRING IS: ABCDEDFJFGBDOFBIDSYFVDFDKBJGUFLJG
C:\>
```