# Principle and Practical Application of Microcomputer

## —— Encryption and Decryption of Caesar Cipher

QIAN Yueqi
15076103

# INTRODUCTION

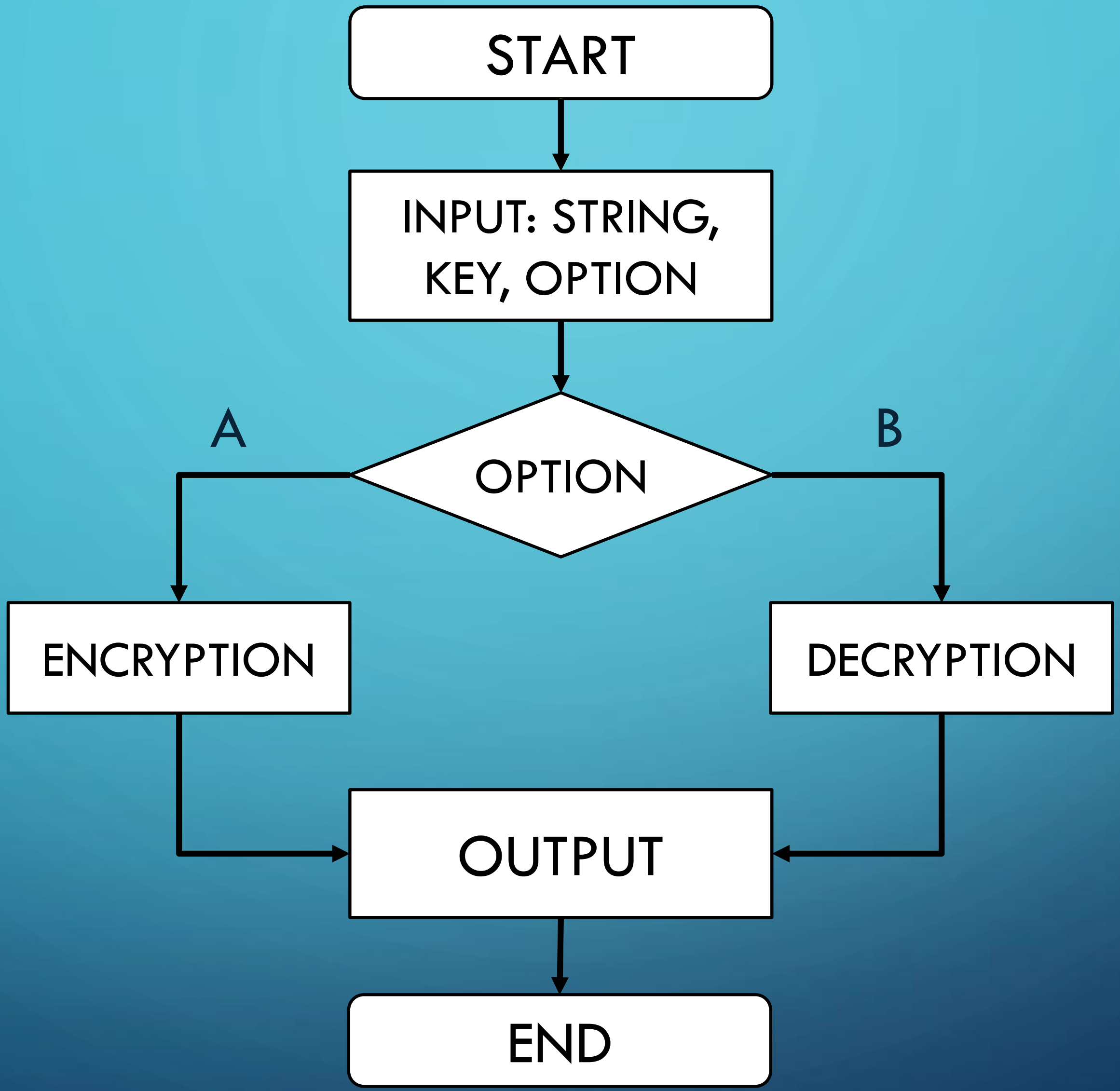
History:

Caesar, the consul in ancient Rome, used a code to communicate with his generals in military operations, which was later called the "Caesar code".

Formula:

$$c = ( m + k ) \bmod 26$$

C: Ciphertext, k: Key ( Cipher ), M: Plaintext



START

INPUT: STRING, KEY, OPTION

OPTION

A

B

ENCRYPTION

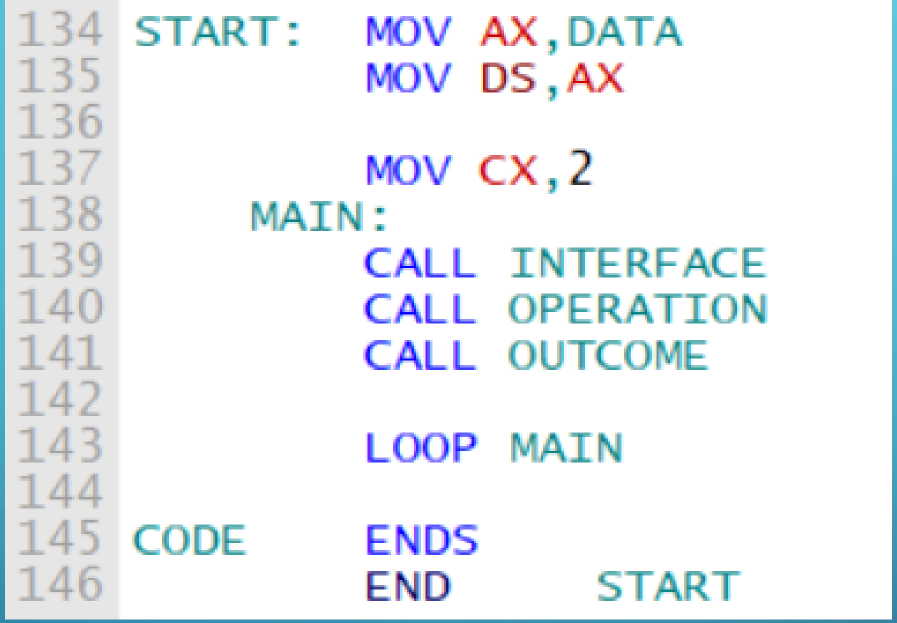
DECRYPTION

OUTPUT

END

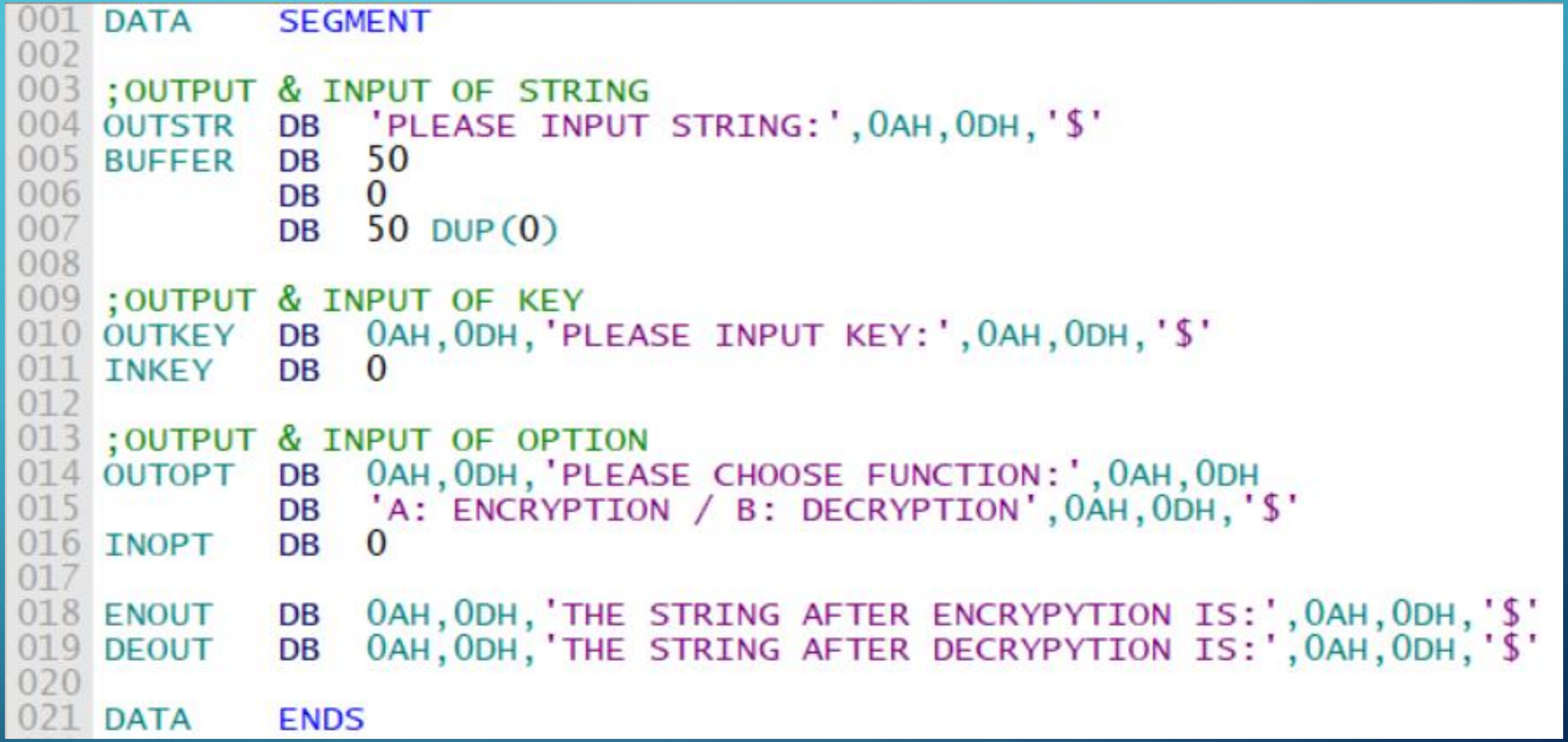
QIAN YUEQI
2019/05/24
HANGZHOU DIANZI UNIVERSITY

# MAIN



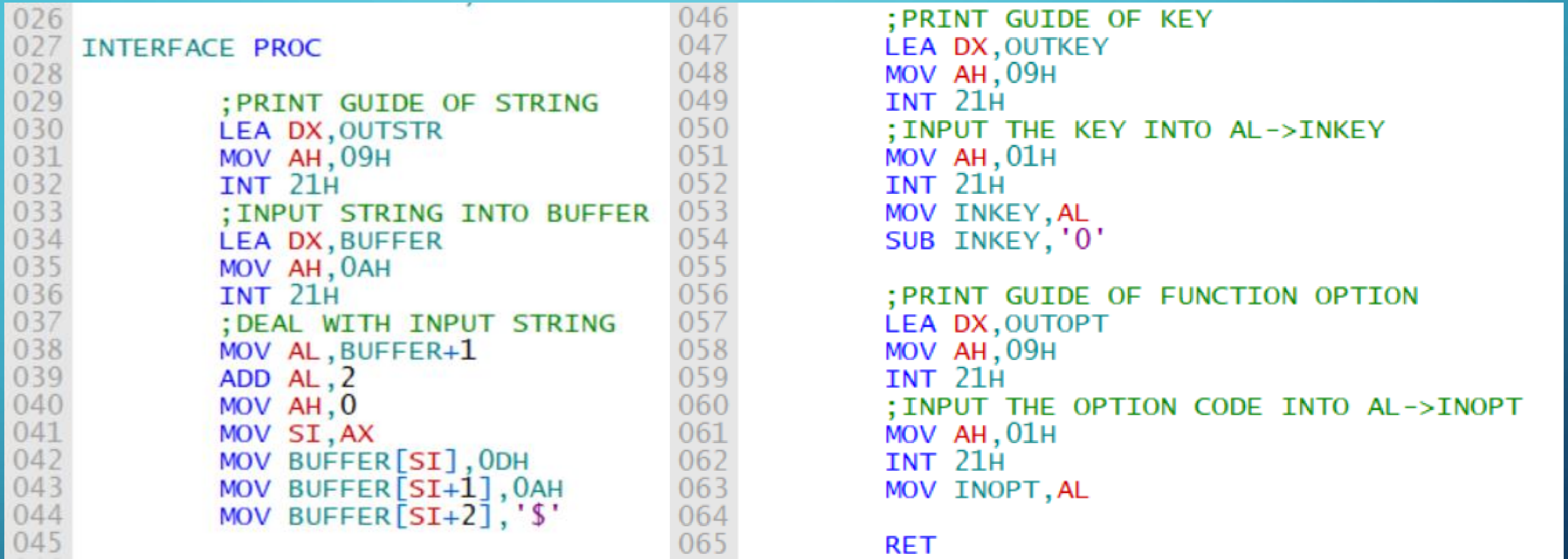
```
START:  MOV AX,DATA
        MOV DS,AX

        MOV CX,2
    MAIN:
        CALL INTERFACE
        CALL OPERATION
        CALL OUTCOME

        LOOP MAIN

CODE    ENDS
        END     START
```

# DATA SEGMENT



```
DATA    SEGMENT

;OUTPUT & INPUT OF STRING
OUTSTR  DB  'PLEASE INPUT STRING:',0AH,0DH,'$'
BUFFER  DB  50
        DB  0
        DB  50 DUP(0)

;OUTPUT & INPUT OF KEY
OUTKEY  DB  0AH,0DH,'PLEASE INPUT KEY:',0AH,0DH,'$'
INKEY   DB  0

;OUTPUT & INPUT OF OPTION
OUTOPT  DB  0AH,0DH,'PLEASE CHOOSE FUNCTION:',0AH,0DH
        DB  'A: ENCRYPTION / B: DECRYPTION',0AH,0DH,'$'
INOPT   DB  0

ENOUT   DB  0AH,0DH,'THE STRING AFTER ENCRYPYTION IS:',0AH,0DH,'$'
DEOUT   DB  0AH,0DH,'THE STRING AFTER DECRYPYTION IS:',0AH,0DH,'$'

DATA    ENDS
```

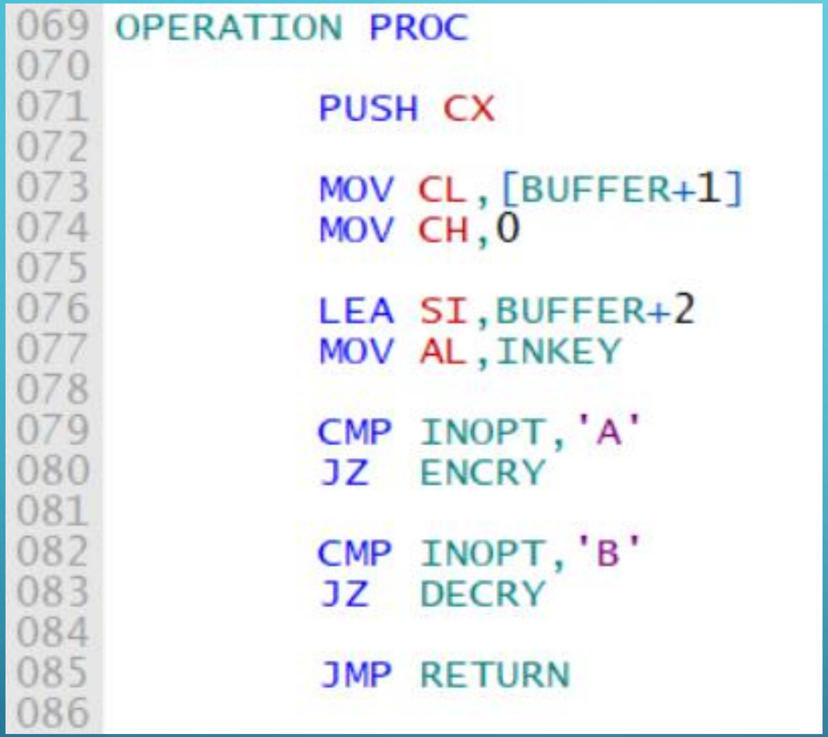
# INTERFACE



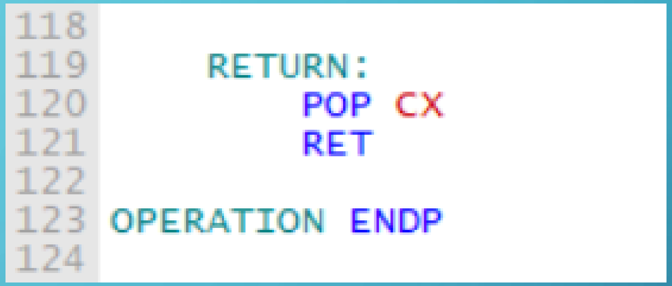
```
INTERFACE PROC

        ;PRINT GUIDE OF STRING
        LEA DX,OUTSTR
        MOV AH,09H
        INT 21H
        ;INPUT STRING INTO BUFFER
        LEA DX,BUFFER
        MOV AH,0AH
        INT 21H
        ;DEAL WITH INPUT STRING
        MOV AL,BUFFER+1
        ADD AL,2
        MOV AH,0
        MOV SI,AX
        MOV BUFFER[SI],0DH
        MOV BUFFER[SI+1],0AH
        MOV BUFFER[SI+2],'$'

        ;PRINT GUIDE OF KEY
        LEA DX,OUTKEY
        MOV AH,09H
        INT 21H
        ;INPUT THE KEY INTO AL->INKEY
        MOV AH,01H
        INT 21H
        MOV INKEY,AL
        SUB INKEY,'0'

        ;PRINT GUIDE OF FUNCTION OPTION
        LEA DX,OUTOPT
        MOV AH,09H
        INT 21H
        ;INPUT THE OPTION CODE INTO AL->INOPT
        MOV AH,01H
        INT 21H
        MOV INOPT,AL

        RET
```

QIAN YUEQI
2019/05/24
HANGZHOU DIANZI UNIVERSITY

# OPTION

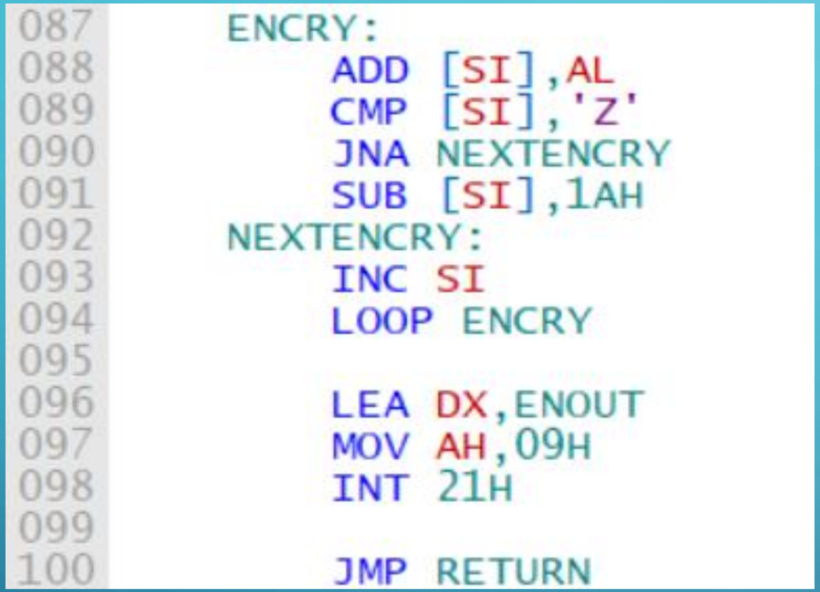


```
OPERATION PROC

        PUSH CX

        MOV CL,[BUFFER+1]
        MOV CH,0

        LEA SI,BUFFER+2
        MOV AL,INKEY

        CMP INOPT,'A'
        JZ  ENCRY

        CMP INOPT,'B'
        JZ  DECRY

        JMP RETURN

```



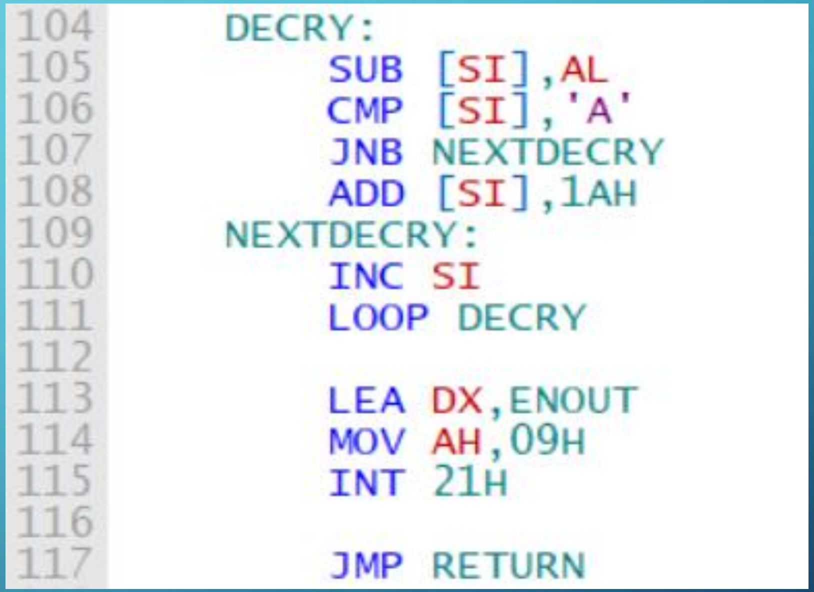
```

    RETURN:
        POP CX
        RET

OPERATION ENDP

```

# ENCRYPTION



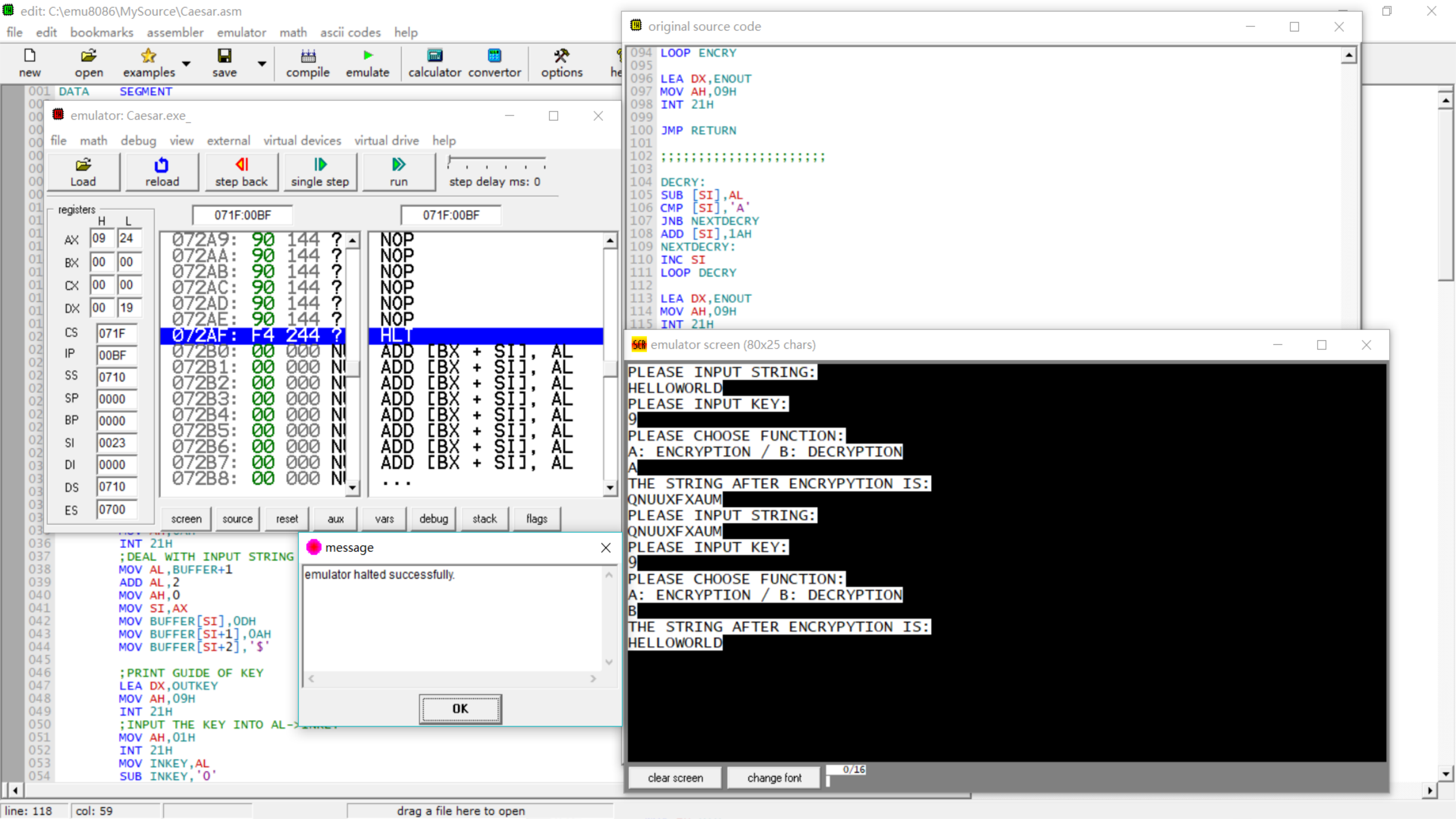
```
        ENCRY:
                ADD  [SI],AL
                CMP  [SI],'Z'
                JNA  NEXTENCRY
                SUB  [SI],1AH
        NEXTENCRY:
                INC  SI
                LOOP  ENCRY

                LEA  DX,ENOUT
                MOV  AH,09H
                INT  21H

                JMP  RETURN
```

# DECRYPTION



```
        DECRY:
                SUB  [SI],AL
                CMP  [SI],'A'
                JNB  NEXTDECRY
                ADD  [SI],1AH
        NEXTDECRY:
                INC  SI
                LOOP  DECRY

                LEA  DX,ENOUT
                MOV  AH,09H
                INT  21H

                JMP  RETURN
```

QIAN YUEQI
2019/05/24
HANGZHOU DIANZI UNIVERSITY


edit: C:\emu8086\MySource\Caesar.asm
file edit bookmarks assembler emulator math ascii codes help
new open examples save compile emulate calculator convertor options
emulator: Caesar.exe_
file math debug view external virtual devices virtual drive help
Load reload step back single step run step delay ms: 0
registers
screen source reset aux vars debug stack flags
original source code
emulator screen (80x25 chars)
clear screen change font 0/16
message
OK

```
DATA    SEGMENT
```

```
        INT 21H
        ;DEAL WITH INPUT STRING
        MOV AL,BUFFER+1
        ADD AL,2
        MOV AH,0
        MOV SI,AX
        MOV BUFFER[SI],0DH
        MOV BUFFER[SI+1],0AH
        MOV BUFFER[SI+2],'$'

        ;PRINT GUIDE OF KEY
        LEA DX,OUTKEY
        MOV AH,09H
        INT 21H
        ;INPUT THE KEY INTO AL->INKEY
        MOV AH,01H
        INT 21H
        MOV INKEY,AL
        SUB INKEY,'0'
```

|  | H | L |
|---|---|---|
| AX | 09 | 24 |
| BX | 00 | 00 |
| CX | 00 | 00 |
| DX | 00 | 19 |
| CS | 071F |  |
| IP | 00BF |  |
| SS | 0710 |  |
| SP | 0000 |  |
| BP | 0000 |  |
| SI | 0023 |  |
| DI | 0000 |  |
| DS | 0710 |  |
| ES | 0700 |  |

071F:00BF    071F:00BF

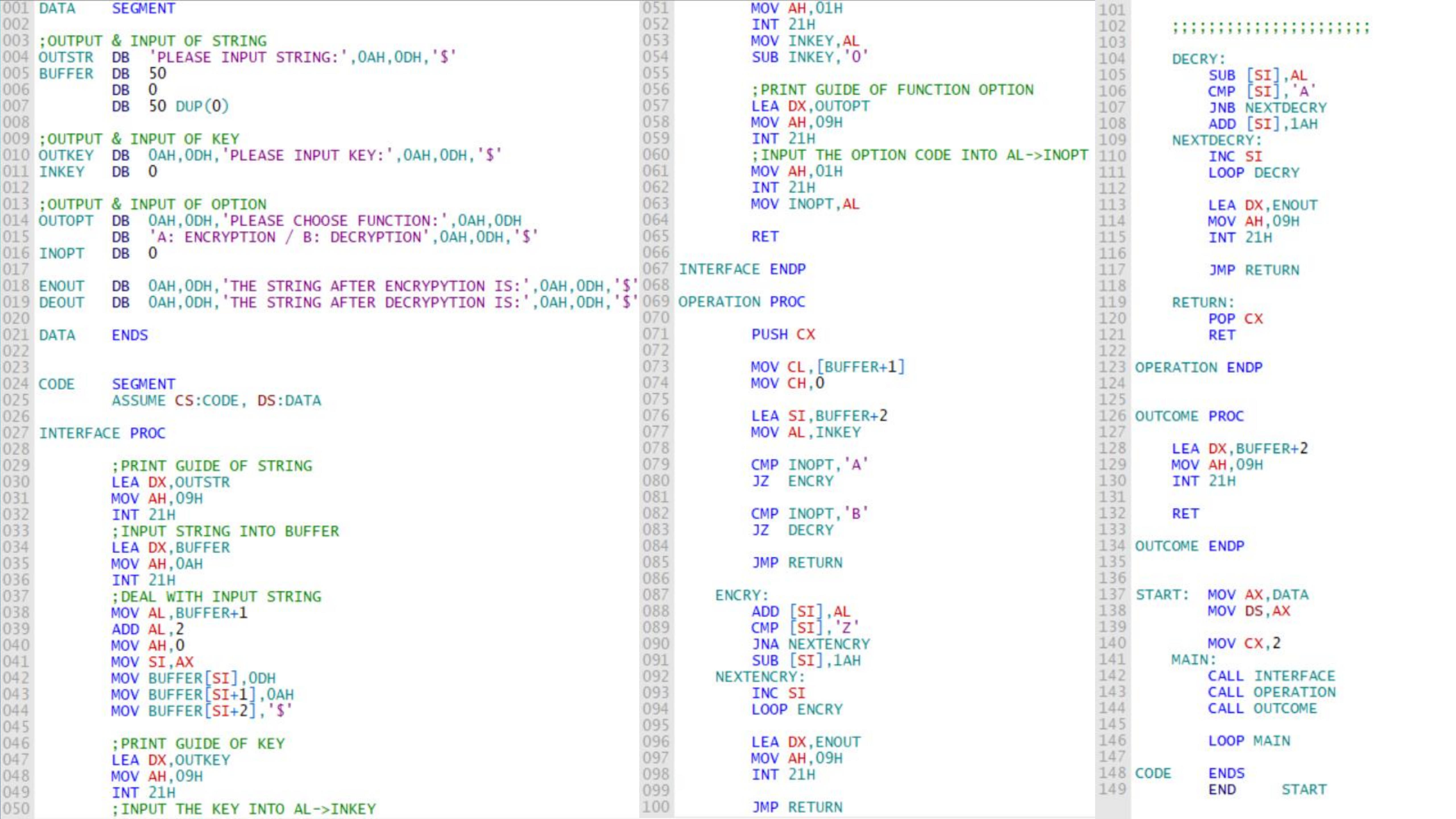
```
072A9: 90 144 ?     NOP
072AA: 90 144 ?     NOP
072AB: 90 144 ?     NOP
072AC: 90 144 ?     NOP
072AD: 90 144 ?     NOP
072AE: 90 144 ?     NOP
072AF: F4 244 ?     HLT
072B0: 00 000 NU    ADD [BX + SI], AL
072B1: 00 000 NU    ADD [BX + SI], AL
072B2: 00 000 NU    ADD [BX + SI], AL
072B3: 00 000 NU    ADD [BX + SI], AL
072B4: 00 000 NU    ADD [BX + SI], AL
072B5: 00 000 NU    ADD [BX + SI], AL
072B6: 00 000 NU    ADD [BX + SI], AL
072B7: 00 000 NU    ADD [BX + SI], AL
072B8: 00 000 NU    ...
```

```
LOOP ENCRY

LEA DX,ENOUT
MOV AH,09H
INT 21H

JMP RETURN

;;;;;;;;;;;;;;;;;;;;;;

DECRY:
SUB [SI],AL
CMP [SI],'A'
JNB NEXTDECRY
ADD [SI],1AH
NEXTDECRY:
INC SI
LOOP DECRY

LEA DX,ENOUT
MOV AH,09H
INT 21H
```

```
PLEASE INPUT STRING:
HELLOWORLD
PLEASE INPUT KEY:
9
PLEASE CHOOSE FUNCTION:
A: ENCRYPTION / B: DECRYPTION
A
THE STRING AFTER ENCRYPYTION IS:
QNUUXFXAUM
PLEASE INPUT STRING:
QNUUXFXAUM
PLEASE INPUT KEY:
9
PLEASE CHOOSE FUNCTION:
A: ENCRYPTION / B: DECRYPTION
B
THE STRING AFTER ENCRYPYTION IS:
HELLOWORLD
```

emulator halted successfully.

line: 118    col: 59    drag a file here to open



```
DATA    SEGMENT

;OUTPUT & INPUT OF STRING
OUTSTR  DB  'PLEASE INPUT STRING:',0AH,0DH,'$'
BUFFER  DB  50
        DB  0
        DB  50 DUP(0)

;OUTPUT & INPUT OF KEY
OUTKEY  DB  0AH,0DH,'PLEASE INPUT KEY:',0AH,0DH,'$'
INKEY   DB  0

;OUTPUT & INPUT OF OPTION
OUTOPT  DB  0AH,0DH,'PLEASE CHOOSE FUNCTION:',0AH,0DH
        DB  'A: ENCRYPTION / B: DECRYPTION',0AH,0DH,'$'
INOPT   DB  0

ENOUT   DB  0AH,0DH,'THE STRING AFTER ENCRYPYTION IS:',0AH,0DH,'$'
DEOUT   DB  0AH,0DH,'THE STRING AFTER DECRYPYTION IS:',0AH,0DH,'$'

DATA    ENDS


CODE    SEGMENT
        ASSUME CS:CODE, DS:DATA

INTERFACE PROC

        ;PRINT GUIDE OF STRING
        LEA DX,OUTSTR
        MOV AH,09H
        INT 21H
        ;INPUT STRING INTO BUFFER
        LEA DX,BUFFER
        MOV AH,0AH
        INT 21H
        ;DEAL WITH INPUT STRING
        MOV AL,BUFFER+1
        ADD AL,2
        MOV AH,0
        MOV SI,AX
        MOV BUFFER[SI],0DH
        MOV BUFFER[SI+1],0AH
        MOV BUFFER[SI+2],'$'

        ;PRINT GUIDE OF KEY
        LEA DX,OUTKEY
        MOV AH,09H
        INT 21H
        ;INPUT THE KEY INTO AL->INKEY
        MOV AH,01H
        INT 21H
        MOV INKEY,AL
        SUB INKEY,'0'

        ;PRINT GUIDE OF FUNCTION OPTION
        LEA DX,OUTOPT
        MOV AH,09H
        INT 21H
        ;INPUT THE OPTION CODE INTO AL->INOPT
        MOV AH,01H
        INT 21H
        MOV INOPT,AL

        RET

INTERFACE ENDP

OPERATION PROC

        PUSH CX

        MOV CL,[BUFFER+1]
        MOV CH,0

        LEA SI,BUFFER+2
        MOV AL,INKEY

        CMP INOPT,'A'
        JZ  ENCRY

        CMP INOPT,'B'
        JZ  DECRY

        JMP RETURN

    ENCRY:
        ADD [SI],AL
        CMP [SI],'Z'
        JNA NEXTENCRY
        SUB [SI],1AH
    NEXTENCRY:
        INC SI
        LOOP ENCRY

        LEA DX,ENOUT
        MOV AH,09H
        INT 21H

        JMP RETURN


    ;;;;;;;;;;;;;;;;;;;;;;

    DECRY:
        SUB [SI],AL
        CMP [SI],'A'
        JNB NEXTDECRY
        ADD [SI],1AH
    NEXTDECRY:
        INC SI
        LOOP DECRY

        LEA DX,ENOUT
        MOV AH,09H
        INT 21H

        JMP RETURN

    RETURN:
        POP CX
        RET

OPERATION ENDP


OUTCOME PROC

    LEA DX,BUFFER+2
    MOV AH,09H
    INT 21H

    RET

OUTCOME ENDP


START:  MOV AX,DATA
        MOV DS,AX

        MOV CX,2
    MAIN:
        CALL INTERFACE
        CALL OPERATION
        CALL OUTCOME

        LOOP MAIN

CODE    ENDS
        END     START
```

# SUMMARY

## Strengths:

- Implement Encryption and Decryption of Caesar Cipher
- Support Multiple Input and Output of Strings
- Boundary Inspection

## Weaknesses:

For me, the value of “key” can only be 0-9 temporarily.

For EMU8086, the usage of ‘?’ is not supported.

# THANK YOU FOR YOUR LISTENING !

# TIME FOR Q&A

QIAN YUEQI
2019/05/24
HANGZHOU DIANZI UNIVERSITY