



Principle and Practical Application of Microcomputer

—— Encryption and Decryption of Caesar Cipher

QIAN Yueqi
15076103

INTRODUCTION

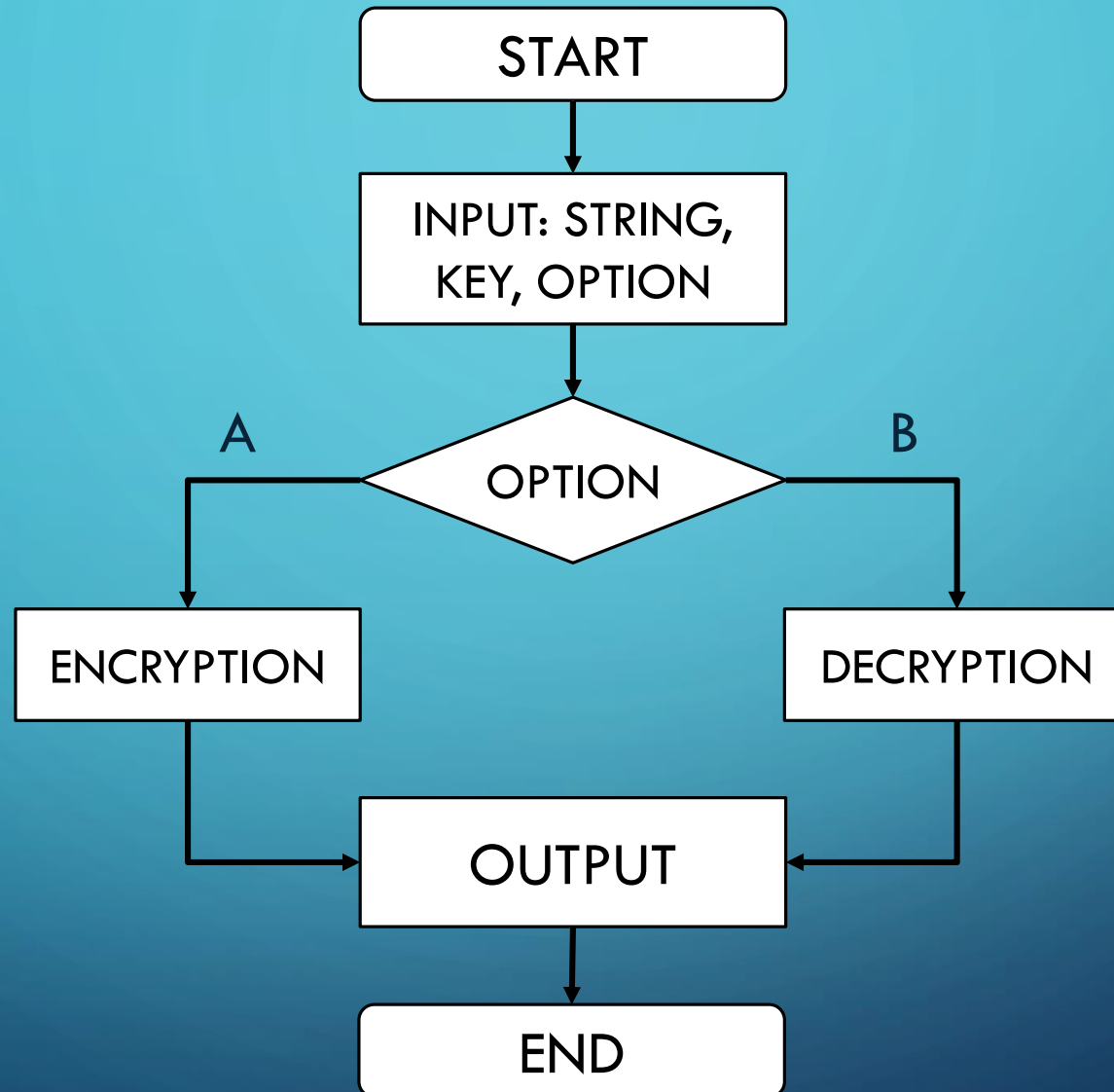
History:

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



MAIN

```
134 START:  MOV AX,DATA
135         MOV DS,AX
136
137         MOV CX,2
138     MAIN:
139         CALL INTERFACE
140         CALL OPERATION
141         CALL OUTCOME
142
143         LOOP MAIN
144
145 CODE     ENDS
146         END      START
```

DATA SEGMENT

```
001 DATA SEGMENT
002
003 ;OUTPUT & INPUT OF STRING
004 OUTSTR DB 'PLEASE INPUT STRING:',0AH,0DH,'$'
005 BUFFER DB 50
006         DB 0
007         DB 50 DUP(0)
008
009 ;OUTPUT & INPUT OF KEY
010 OUTKEY DB 0AH,0DH,'PLEASE INPUT KEY:',0AH,0DH,'$'
011 INKEY DB 0
012
013 ;OUTPUT & INPUT OF OPTION
014 OUTOPT DB 0AH,0DH,'PLEASE CHOOSE FUNCTION:',0AH,0DH
015         DB 'A: ENCRYPTION / B: DECRYPTION',0AH,0DH,'$'
016 INOPT DB 0
017
018 ENOUT DB 0AH,0DH,'THE STRING AFTER ENCRYPYTION IS:',0AH,0DH,'$'
019 DEOUT DB 0AH,0DH,'THE STRING AFTER DECRYPTION IS:',0AH,0DH,'$'
020
021 DATA ENDS
```

INTERFACE

```
026
027 INTERFACE PROC
028
029     ;PRINT GUIDE OF STRING
030     LEA DX,OUTSTR
031     MOV AH,09H
032     INT 21H
033     ;INPUT STRING INTO BUFFER
034     LEA DX,BUFFER
035     MOV AH,0AH
036     INT 21H
037     ;DEAL WITH INPUT STRING
038     MOV AL,BUFFER+1
039     ADD AL,2
040     MOV AH,0
041     MOV SI,AX
042     MOV BUFFER[SI],0DH
043     MOV BUFFER[SI+1],0AH
044     MOV BUFFER[SI+2],'$'
045
046     ;PRINT GUIDE OF KEY
047     LEA DX,OUTKEY
048     MOV AH,09H
049     INT 21H
050     ;INPUT THE KEY INTO AL->INKEY
051     MOV AH,01H
052     INT 21H
053     MOV INKEY,AL
054     SUB INKEY,'0'
055
056     ;PRINT GUIDE OF FUNCTION OPTION
057     LEA DX,OUTOPT
058     MOV AH,09H
059     INT 21H
060     ;INPUT THE OPTION CODE INTO AL->INOPT
061     MOV AH,01H
062     INT 21H
063     MOV INOPT,AL
064
065     RET
```


OPTION

```
069 OPERATION PROC
070
071     PUSH CX
072
073     MOV CL,[BUFFER+1]
074     MOV CH,0
075
076     LEA SI,BUFFER+2
077     MOV AL,INKEY
078
079     CMP INOPT,'A'
080     JZ ENCRY
081
082     CMP INOPT,'B'
083     JZ DECRY
084
085     JMP RETURN
086
```

```
118
119     RETURN:
120         POP CX
121         RET
122
123 OPERATION ENDP
124
```

ENCRYPTION

```
087      ENCRY:
088          ADD [SI],AL
089          CMP [SI],'Z'
090          JNA NEXTENCRY
091          SUB [SI],1AH
092      NEXTENCRY:
093          INC SI
094          LOOP ENCRY
095
096          LEA DX,ENOUT
097          MOV AH,09H
098          INT 21H
099
100          JMP RETURN
```

DECRYPTION

```
104      DECRY:
105          SUB [SI],AL
106          CMP [SI],'A'
107          JNB NEXTDECRY
108          ADD [SI],1AH
109      NEXTDECRY:
110          INC SI
111          LOOP DECRY
112
113          LEA DX,ENOUT
114          MOV AH,09H
115          INT 21H
116
117          JMP RETURN
```


new open examples save compile emulate calculator convertor options

001 DATA SEGMENT

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

	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 NI	ADD [BX + SI], AL
072B1: 00 000 NI	ADD [BX + SI], AL
072B2: 00 000 NI	ADD [BX + SI], AL
072B3: 00 000 NI	ADD [BX + SI], AL
072B4: 00 000 NI	ADD [BX + SI], AL
072B5: 00 000 NI	ADD [BX + SI], AL
072B6: 00 000 NI	ADD [BX + SI], AL
072B7: 00 000 NI	ADD [BX + SI], AL
072B8: 00 000 NI	...

screen source reset aux vars debug stack flags

```

036 INT 21H
037 ;DEAL WITH INPUT STRING
038 MOV AL,BUFFER+1
039 ADD AL,2
040 MOV AH,0
041 MOV SI,AX
042 MOV BUFFER[SI],0DH
043 MOV BUFFER[SI+1],0AH
044 MOV BUFFER[SI+2],'$'
045
046 ;PRINT GUIDE OF KEY
047 LEA DX,OUTKEY
048 MOV AH,09H
049 INT 21H
050 ;INPUT THE KEY INTO AL
051 MOV AH,01H
052 INT 21H
053 MOV INKEY,AL
054 SUB INKEY,'0'

```

message

emulator halted successfully.

OK

```

094 LOOP ENCRY
095
096 LEA DX,ENOUT
097 MOV AH,09H
098 INT 21H
099
100 JMP RETURN
101
102 .....
103
104 DECRY:
105 SUB [SI],AL
106 CMP [SI],'A'
107 JNB NEXTDECRY
108 ADD [SI],1AH
109 NEXTDECRY:
110 INC SI
111 LOOP DECRY
112
113 LEA DX,ENOUT
114 MOV AH,09H
115 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

```

001	DATA	SEGMENT	051	MOV AH,01H	101	
002			052	INT 21H	102	;;;;;;;;;
003	;OUTPUT & INPUT OF STRING		053	MOV INKEY,AL	103	
004	OUTSTR DB 'PLEASE INPUT STRING:',0AH,0DH,'\$'		054	SUB INKEY,'0'	104	DECRY:
005	BUFFER DB 50		055		105	SUB [SI],AL
006	DB 0		056		106	CMP [SI],'A'
007	DB 50 DUP(0)		057	;PRINT GUIDE OF FUNCTION OPTION	107	JNB NEXTDECY
008			058	LEA DX,OUTOPT	108	ADD [SI],1AH
009	;OUTPUT & INPUT OF KEY		059	MOV AH,09H	109	NEXTDECY:
010	OUTKEY DB 0AH,0DH,'PLEASE INPUT KEY:',0AH,0DH,'\$'		060	INT 21H	110	INC SI
011	INKEY DB 0		061	;INPUT THE OPTION CODE INTO AL->INOPT	111	LOOP DECRY
012			062	MOV AH,01H	112	
013	;OUTPUT & INPUT OF OPTION		063	INT 21H	113	LEA DX,ENOUT
014	OUTOPT DB 0AH,0DH,'PLEASE CHOOSE FUNCTION:',0AH,0DH		064	MOV INOPT,AL	114	MOV AH,09H
015	DB 'A: ENCRYPTION / B: DECRYPTION',0AH,0DH,'\$'		065		115	INT 21H
016	INOPT DB 0		066	RET	116	
017			067	INTERFACE ENDP	117	JMP RETURN
018	ENOUT DB 0AH,0DH,'THE STRING AFTER ENCRYPYTION IS:',0AH,0DH,'\$'		068		118	
019	DEOUT DB 0AH,0DH,'THE STRING AFTER DECRYPYTION IS:',0AH,0DH,'\$'		069	OPERATION PROC	119	RETURN:
020			070		120	POP CX
021	DATA ENDS		071	PUSH CX	121	RET
022			072		122	
023			073	MOV CL,[BUFFER+1]	123	OPERATION ENDP
024	CODE SEGMENT		074	MOV CH,0	124	
025	ASSUME CS:CODE, DS:DATA		075		125	
026			076	LEA SI,BUFFER+2	126	OUTCOME PROC
027	INTERFACE PROC		077	MOV AL,INKEY	127	
028			078		128	LEA DX,BUFFER+2
029	;PRINT GUIDE OF STRING		079	CMP INOPT,'A'	129	MOV AH,09H
030	LEA DX,OUTSTR		080	JZ ENCRY	130	INT 21H
031	MOV AH,09H		081		131	
032	INT 21H		082	CMP INOPT,'B'	132	RET
033	;INPUT STRING INTO BUFFER		083	JZ DECRY	133	
034	LEA DX,BUFFER		084		134	OUTCOME ENDP
035	MOV AH,0AH		085	JMP RETURN	135	
036	INT 21H		086		136	
037	;DEAL WITH INPUT STRING		087	ENCRY:	137	START: MOV AX,DATA
038	MOV AL,BUFFER+1		088	ADD [SI],AL	138	MOV DS,AX
039	ADD AL,2		089	CMP [SI],'Z'	139	
040	MOV AH,0		090	JNA NEXTENCRY	140	MOV CX,2
041	MOV SI,AX		091	SUB [SI],1AH	141	MAIN:
042	MOV BUFFER[SI],0DH		092	NEXTENCRY:	142	CALL INTERFACE
043	MOV BUFFER[SI+1],0AH		093	INC SI	143	CALL OPERATION
044	MOV BUFFER[SI+2],'\$'		094	LOOP ENCRY	144	CALL OUTCOME
045			095		145	
046	;PRINT GUIDE OF KEY		096	LEA DX,ENOUT	146	LOOP MAIN
047	LEA DX,OUTKEY		097	MOV AH,09H	147	
048	MOV AH,09H		098	INT 21H	148	CODE ENDS
049	INT 21H		099		149	END START
050	;INPUT THE KEY INTO AL->INKEY		100	JMP RETURN		

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