

บรรณานุกรม



Andy Johnson, The Programmer's CP/M Handbook, Osborn/
McGraw Hill

Alarn R. Miller, Mastering CP/M, Sybex Inc.

Microsoft, Softcard vol 1.

_____, Softcard vol 2.

Rodney Zaks, Programming the Z80

E Horowitz & S Sahni, Fundamentals of Data Structure,

Pitman

Tremblay Sorenson, An Introduction to Data Structures
with Application, McGraw-Hill

William Payne & Patricia Payne, Implementing BASICS,

Riston Publishing Company, Inc.

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย



ภาคผนวก ก.

คำสั่งและรหัสของคำสั่ง

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

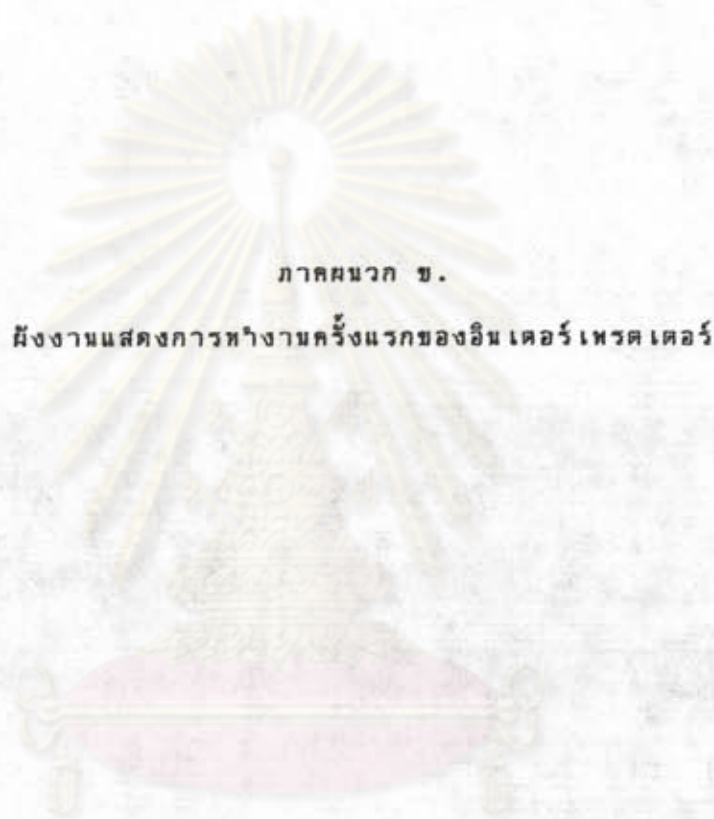
คำสั่ง	รหัสคำสั่ง	คำสั่ง	รหัสคำสั่ง
AND	F7	DATA	84
ABS	FFB6	DIM	86
ATN	FFBE	DEFSTR	A9
ASC	FF94	DEFINT	AA
AUTO	A7	DEFSNG	AB
BEEP	D4	DEFDBL	AC
CLOSE	BC	DEF	96
CONT	98	DEL	A6
CLEAR	92	END	81
CINT	FF9B	ELSE	9E
CSNG	FF9C	ERASE	A2
CDBL	FF9D	EDIT	A3
CVI	FFAA	ERROR	A4
CVS	FFAB	ERL	E5
CVD	FFAC	ERR	E6
COS	FF8C	EXP	FF8B
CHRS	FF95	EOF	FFAE
CALL	B1	EQV	FA
COMMON	B3	FOR	82
CHAIN	B4	FIELD	B9
COLOR	CD	FILES	BF
CREATE	D1	FN	E2



คำสั่ง	รหัสคำสั่ง	คำสั่ง	รหัสคำสั่ง
FRE	FF8F	LINE	AD
FIX	FF9E	LOAD	BD
GOTO	89	LSET	C2
GO TO	89	LPRINT	9B
GOSUB	8D	LLIST	9C
GET	BA	LPOS	FF9A
GR	CC	LIST	93
HOME	C7	LOC	FFAF
HLIN	CE	LOG	FF8A
HTAB	C9	LEN	FF91
HSCRN	ED	LEFT\$	FF81
HEX\$	FF99	LOF	FFB0
INPUT	85	MERGE	BE
IF	8B	MOD	FC
INSTR	E9	MKI\$	FFB1
INT	FF85	MKS\$	FFB2
IMP	FB	MKD\$	FFB3
INKEY\$	EK	MID\$	FF83
INVERSE	CA	NEXT	83
INSERT	D2	NORMAL	CB
KILL	C1	NOTRACE	AO
LET	88	NAME	CO

คำสั่ง	รหัสคำสั่ง	คำสั่ง	รหัสคำสั่ง
NEW	94	RIGHT\$	FF82
NOT	E4	RND	FF88
ON	95	RENUM	A8
OPEN	B8	RESET	C5
OR	F8	RANDOMIZE	B6
OCT\$	FF98	STOP	90
OPTION	B5	SWAP	A1
PUT	BB	SAVE	C4
POKE	97	SPC(E3
PRINT	91	STEP	E0
POS	FF90	SGN	FF84
PEEK	FF96	SQR	FF87
PLOT	D0	SIN	FF89
POP	AE	STR\$	FF92
REMOVE	D3	STRING\$	E7
READ	87	SPACE\$	FF97
RUN	8A	SYSTEM	B7
RESTORE	8C	SEARCH	D6
RETURN	8E	TRACE	9F
REM	8F	TAB(DF
RESUME	A5	TO	DD
RSET	C3	THEN	DE

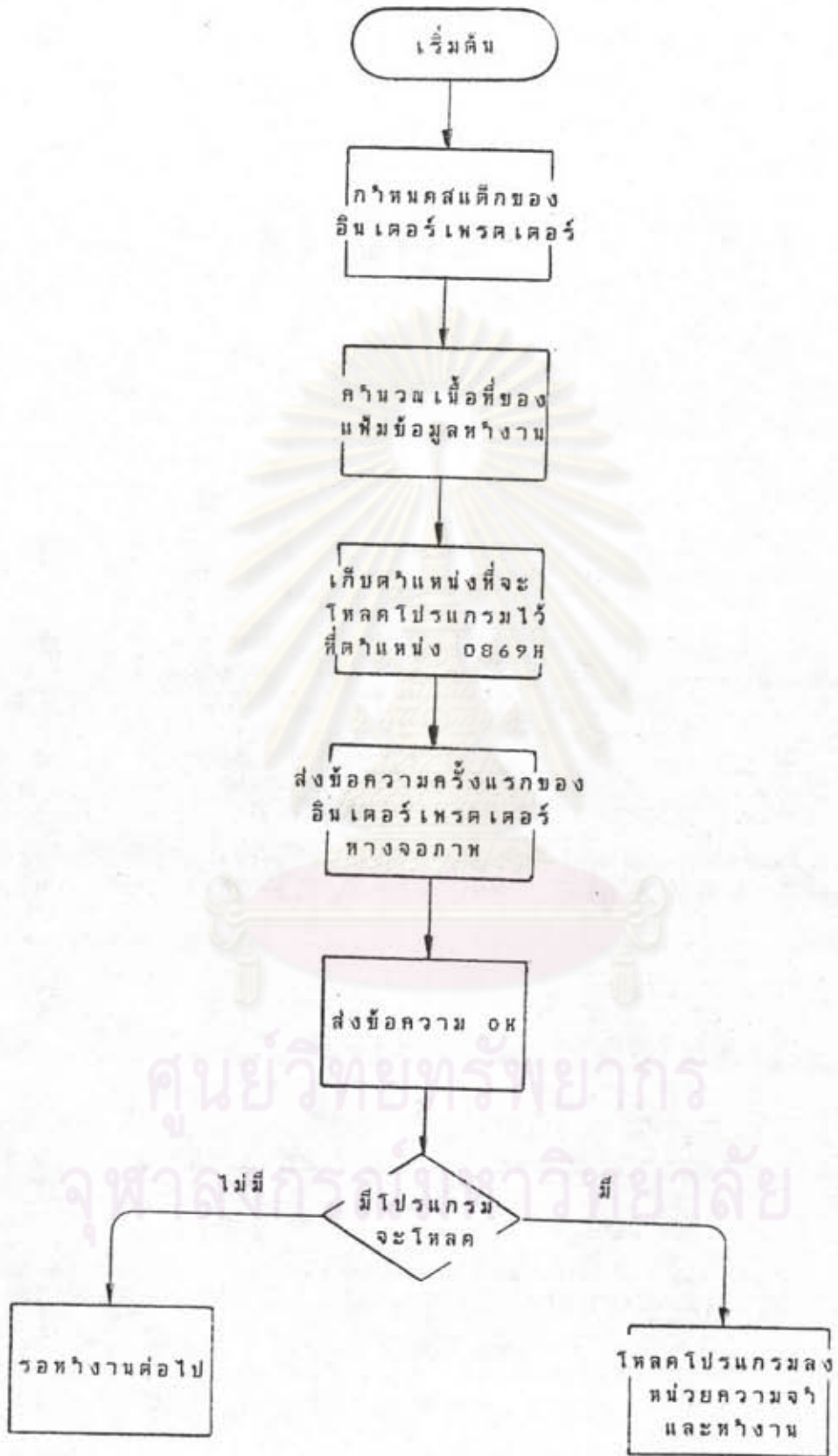
คำสั่ง	รหัสคำสั่ง	คำสั่ง	รหัสคำสั่ง
TAN	FF8D	~	F6
TEXT	C6	\	FD
USING	E8	'	EA
USR	E1	>	EF
UPDATE	D7		
VAL	FF93		
VARPTR	EB		
VLIN	CF		
VTAB	CB		
VPOS	FFB4		
WIDTH	9D		
WAIT	D5		
WHILE	AF		
WEND	B0		
WRITE	B2		
XOR	F9		
=	F0		
<	F1		
+	F2		
-	F3		
*	F4		
/	F5		



ภาคผนวก ข.

ผังงานแสดงการทำงานครั้งแรกของอินเตอร์เน็ตเตอร์

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย



ศูนย์วิทยากร
จุฬาลงกรณ์มหาวิทยาลัย



ภาคผนวก ก.

โปรแกรมทำการย้ายส่วนการหางานครั้งแรกของอินเตอร์เน็ตเตอร์

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

```

; ***** db
; *
; * PROGRAM ID : MOVECODE.MAC *
; * AUTHOR : Prasert Fungwanit *
; * DATE WRITTEN : 18 April 1987 *
; * DATE REVISION : 23 April 1987 *
; *
; *****
; I
; I FUNCTION :- I
; I - Move memory content of BASIC Initialization Routine which I
; I locates at address '5E51H' - '60FFH' to address '6867H' - I
; I '6E65H' I
; I - Add some necessary memory content that is address between I
; I '5E51H' and '60FFH' with displacement of movement I
; I
; *****

```

```

0000' MAIN: ORG 8000H
8000' CD 804A' MOVE_COMM_AWAY: CALL MOVE_AWAY
8003' C3 0000 JP 0000H
8006' CD 8015' MOVE_BASIC: CALL INIT_DATA
8009' CD 801E' CALL CHG_ADDR
800C' CD 803C' CALL MOVE_MEM
800F' CD 8057' MOVE_COMM_BACK: CALL MOVE_BACK
8012' C3 0000 JP 0000H

```

; This subroutine initializes
; Register B with no. of address that will be added with
; displacement of movement & Register pair IX with starting
; address of changed address table

```

8015' 3A 8064' INIT_DATA: LD A,(NO_CHG_ADDR)
8018' 47 LD B,A
8019' DD 21 8067' LD IX,CHG_ADDR_TABL
801D' C9 RET

```

; This subroutine adds memory content whose address appears
; in changed address table with displacement of movement

```

801E' DD 5E 00 CHG_ADDR: LD E,(IX) ; Load DE with next changed
8021' DD 56 01 LD D,(IX+1) ; address.
8024' D5 PUSH DE ; Save current changed addr.
8025' 1A LD A,(DE) ;
8026' 6F LD L,A ; Load content of changed addr
8027' 13 INC DE ; into HL.
8028' 1A LD A,(DE) ;
8029' 67 LD H,A ; Now HL contains content.
802A' ED 58 8065' LD DE,(DISPLACEMENT) ;
802E' 19 ADD HL,DE ; Add HL with movement dispat.
802F' D1 POP DE ; Restore current changed addr.
8030' 7D LD A,L ;

```

```

8031' 12          LD (DE),A          ; Load added address back to DE
8032' 13          INC DE              ; memory content.
8033' 7C          LD A,H              ;
8034' 12          LD (DE),A          ; Now memory content is new addr.
8035' DD 23       INC IX              ;
8037' DD 23       INC IX              ; Skip forward to next chg. addr.
8039' 10 E3       DJNZ CHG_ADDR      ;
803B' C9          RET

```

```

-----
; This subroutine move memory content of address '5E51H' - '60FFH'
; to address '68B7H' - '6E65H'
-----

```

```

803C' 2A 8091'    MOVE_MEM:        LD HL,(BEG_SOURCE_ADDR)
803F' ED 48 8093' LD BC,(NO_ADDR_MOVE)
8043' ED 58 8095' LD DE,(BEG_DESTIN_ADDR)
8047' ED 80       LDIR
8049' C9         RET

```

```

804A' 21 0180     MOVE_AWAY:       LD HL,0180H
804D' ED 48 8097' LD BC,(COMM_LEN)
8051' 11 9000     LD DE,9000H
8054' ED 80       LDIR
8056' C9         RET

```

```

8057' 21 9000     MOVE_BACK:       LD HL,9000H
805A' ED 48 8097' LD BC,(COMM_LEN)
805E' 11 5E51     LD DE,5E51H
8061' ED 80       LDIR
8063' C9         RET

```

```

8064' 15         NO_CHG_ADDR:      DEFB 15H
8065' 0D66       DISPLACEMENT:     DEFW 0D66H
8067' 5E45       CHG_ADDR_TBL:     DEFW 5E45H
8069' 5E52       DEFW 5E52H
806B' 5E01       DEFW 5E01H
806D' 5F28       DEFW 5F28H
806F' 5F2E       DEFW 5F2EH
8071' 5F31       DEFW 5F31H
8073' 5F35       DEFW 5F35H
8075' 5F39       DEFW 5F39H
8077' 5F41       DEFW 5F41H
8079' 5F44       DEFW 5F44H
807B' 5F4F       DEFW 5F4FH
807D' 5F55       DEFW 5F55H
807F' 5F5F       DEFW 5F5FH
8081' 5F69       DEFW 5F69H
8083' 5F77       DEFW 5F77H
8085' 5F86       DEFW 5F86H
8087' 5F8A       DEFW 5F8AH
8089' 5FE1       DEFW 5FE1H
808B' 6054       DEFW 6054H
808D' 605E       DEFW 605EH
808F' 0101       DEFW 0101H
8091' 5E51       BEG_SOURCE_ADDR:  DEFW 5E51H

```

8093'	02AF	NO_ADDR_MOVE:	DEFW 02AFH	๗๔
8095'	6BB7	BEG_DESTIN_ADDR:	DEFW 5E51H + 0D66H	
8097'	0D66	COMM_LEN:	DEFW 0D66H	; No. of index comm. byte
			END MAIN	



ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

Macros:

๗๔

Symbols:

8095'	BEG_DESTIN_ADDR	8091'	BEG_SOURCE_ADDR	801E'	CHG_ADDR
8067'	CHG_ADDR_TABL	8097'	COMM_LEN	8065'	DISPLACEMENT
8015'	INIT_DATA	0000'	MAIN	804A'	MOVE_AWAY
8057'	MOVE_BACK	8006'	MOVE_BASIC	8000'	MOVE_COMM_AWAY
800F'	MOVE_COMM_BACK	803C'	MOVE_MEM	8093'	NO_ADDR_MOVE
8064'	NO_CHG_ADDR				

No fatal error(s)



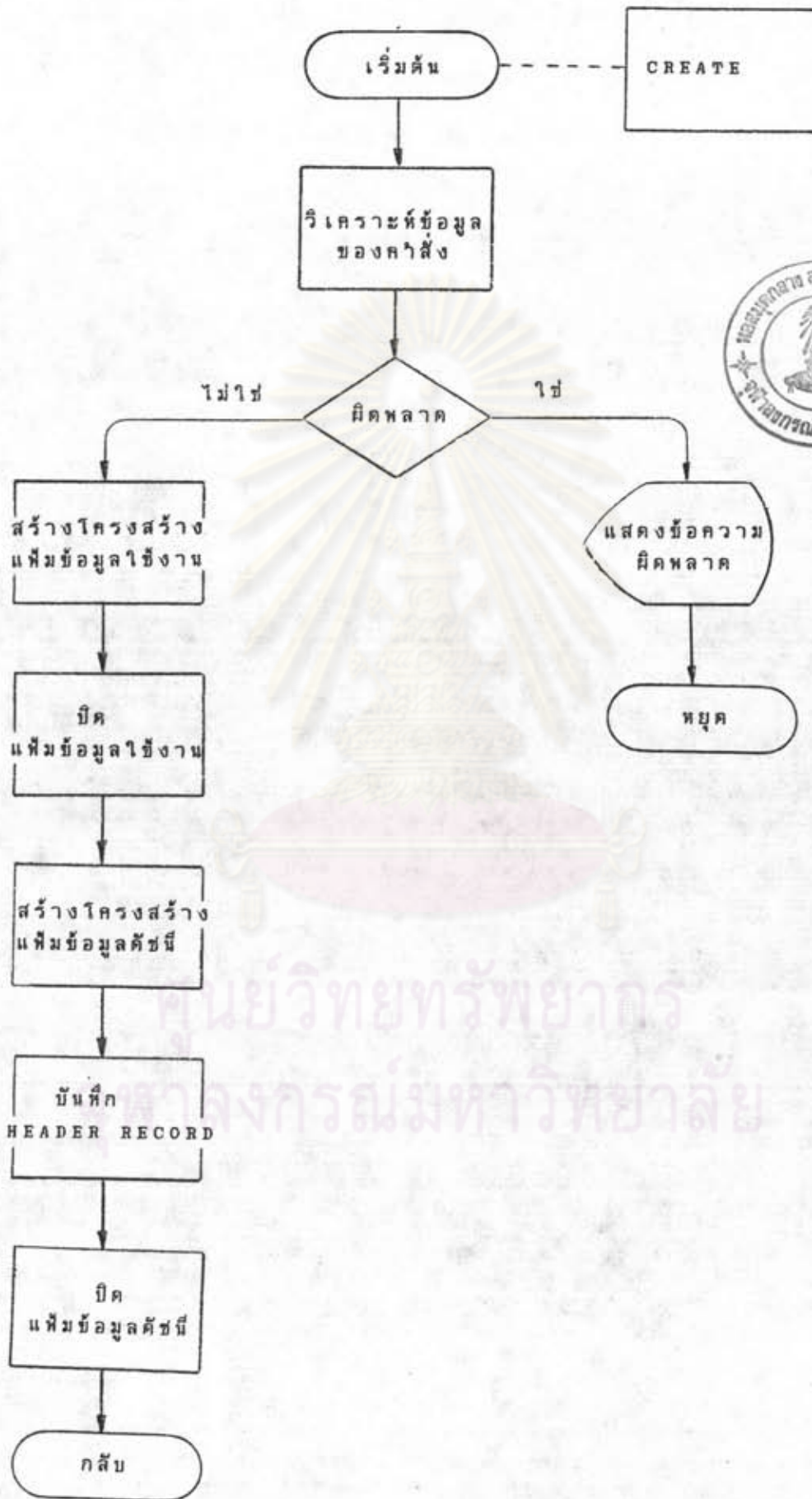
ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย



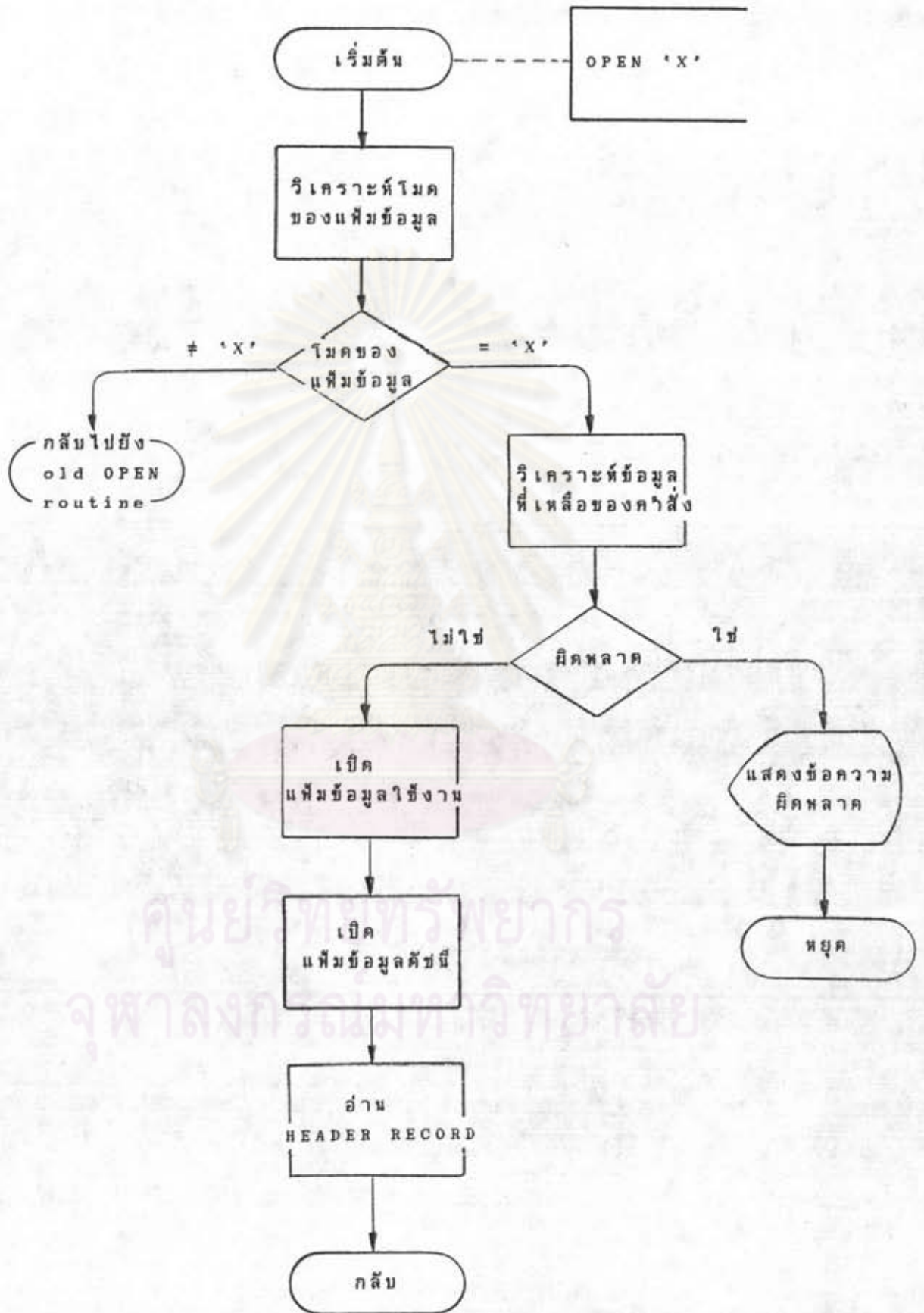
ภาคผนวก ง.

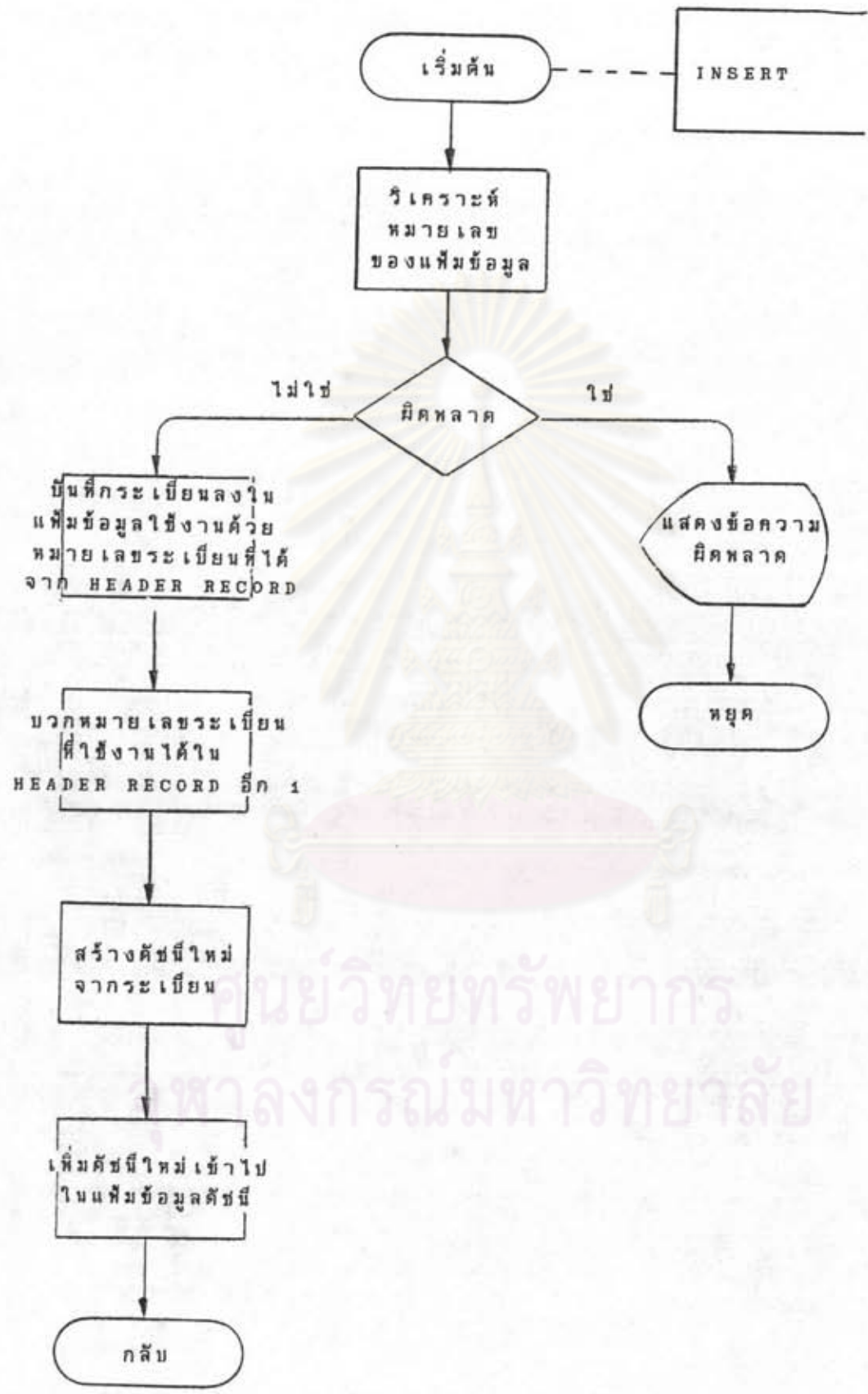
ผังงานแสดงการทำงานของคำสั่งการจัดเก็บแฟ้มข้อมูลแบบดัชนี

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย



ศูนย์วิทยทรัพยากร
มหาวิทยาลัยสุโขทัย





ศูนย์วิทยทรัพยากร
ศาลงกรณ์มหาวิทยาลัย

SEARCH

เริ่มต้น

วิเคราะห์
หมายเลข
ของแฟ้มข้อมูล

ผิดพลาด

ไม่ใช่

ใช่

สร้างคีย์จากระเบียน

แสดงข้อความผิดพลาด

ค้นหาคีย์ที่ตรงกัน
ในแฟ้มข้อมูลคีย์นี้

หยุด

ไม่ใช่

ใช่

พบ

ส่งรหัส
ข้อผิดพลาด

ได้หมายเลข
ทะเบียนข้อมูล
จากคีย์นี้ของคีย์นั้น

กลับ

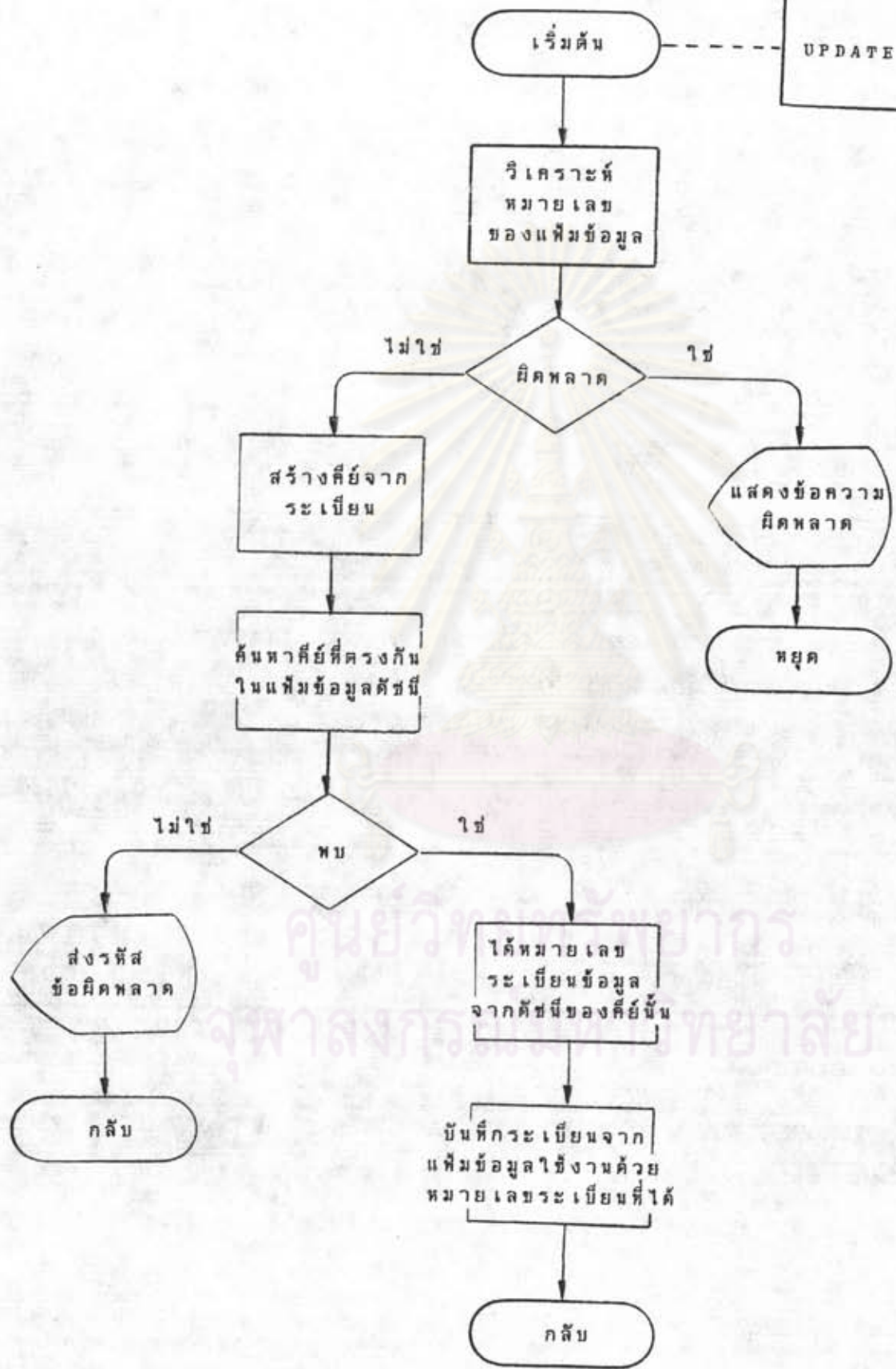
อ่านระเบียนจาก
แฟ้มข้อมูลใช้งานด้วย
หมายเลขทะเบียนที่ได้

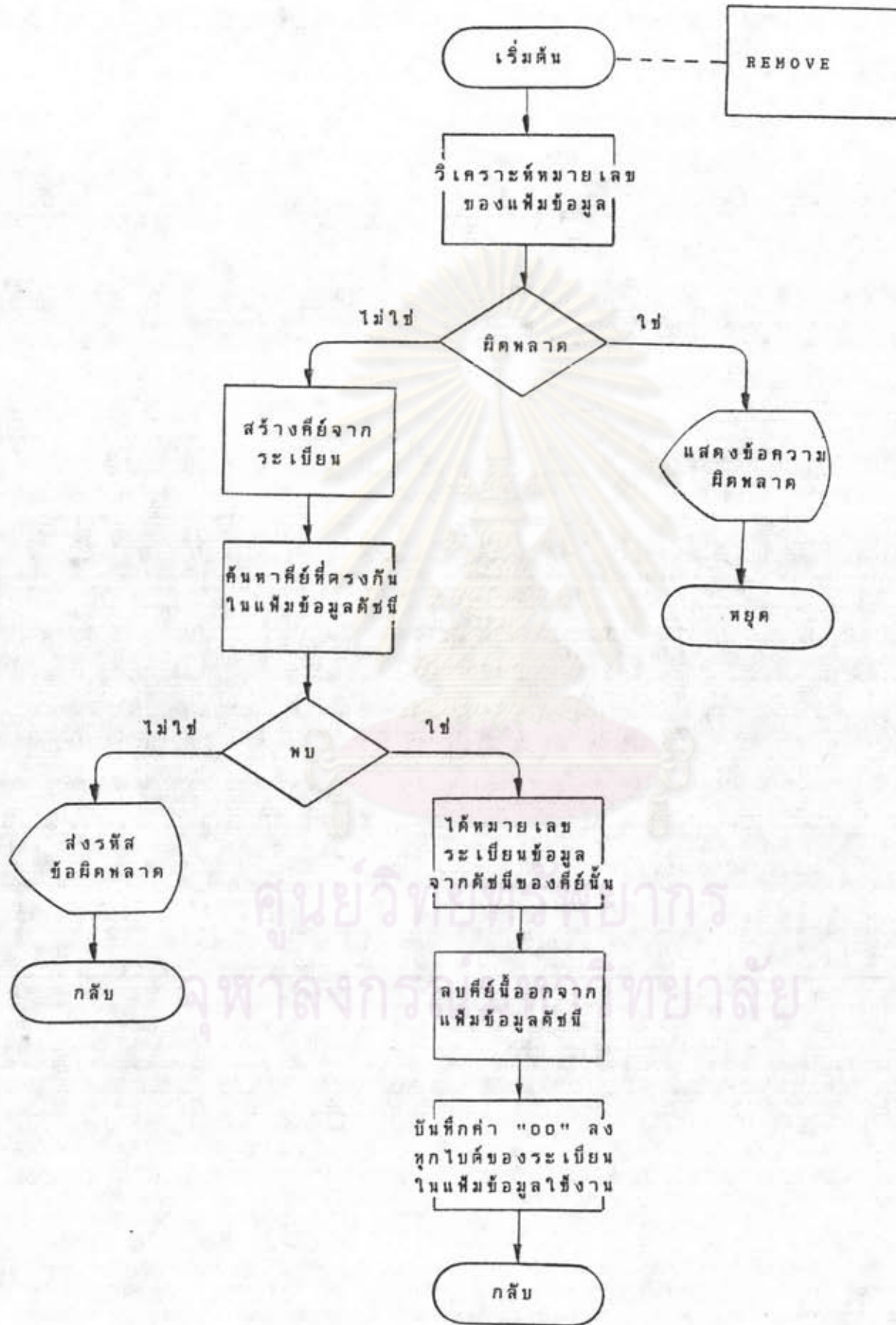
กลับ

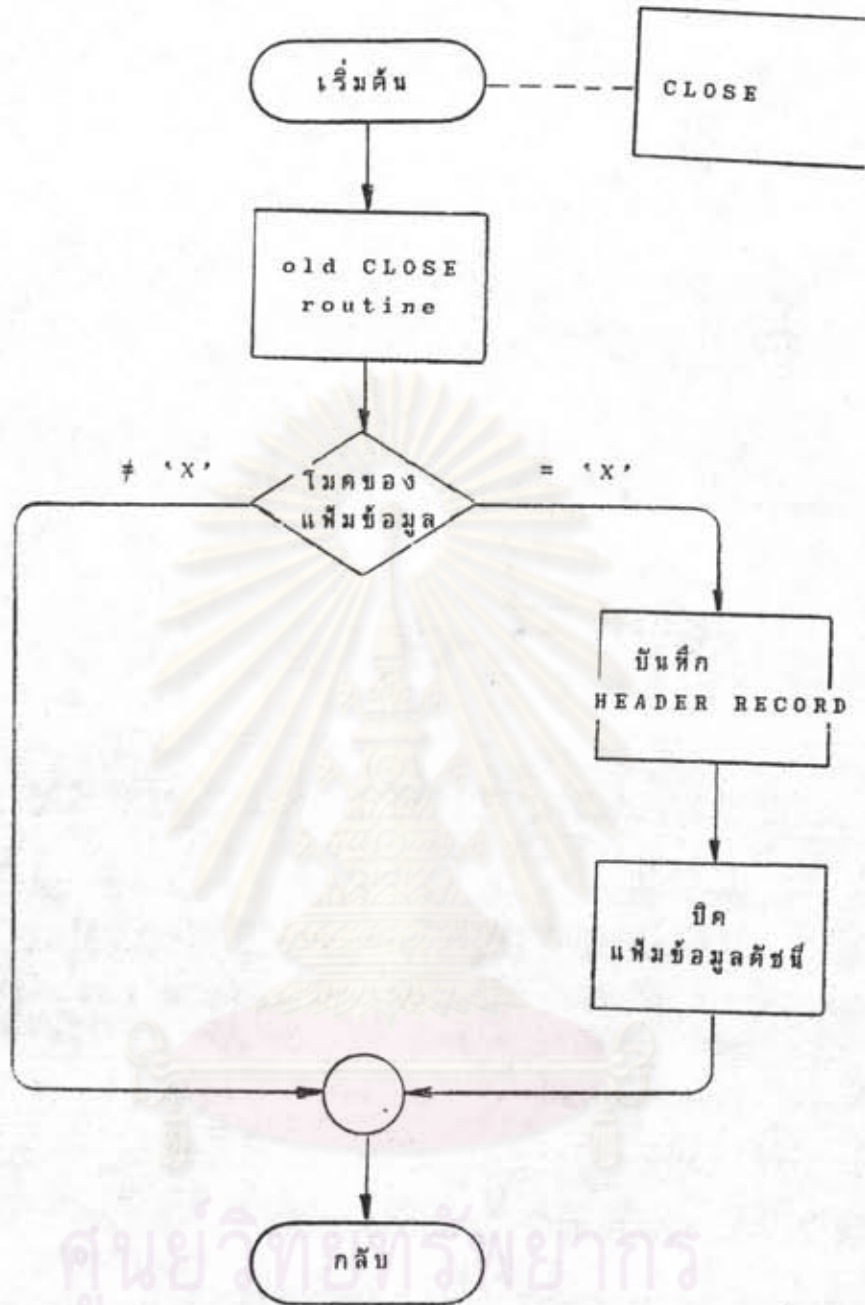


ศูนย์วิจัยระบบสารบรรณ
จุฬาลงกรณ์มหาวิทยาลัย

UPDATE







ศูนย์วิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย



ภาคผนวก จ.

โปรแกรมการทำงานของคำสั่งการจัดเก็บแฟ้มข้อมูลแบบดัชนี

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

```

C
; *****
; *   PROGRAM ID   : INDEX.MAC   *
; *   AUTHOR      : Prasert Fungwanit *
; *   DATE WRITTEN : 3 May 1987 *
; *   DATE REVISION : 3 May 1987 *
; *****
; I   FUNCTION :-                I
; I       - Command jump vector  I
; *****

```

```

0000'
SE51' C3 5E66'
SE54' C3 5F7E'
SE57' C3 6028'
SE5A' C3 619B'
SE5D' C3 6119'
SE60' C3 615A'
SE63' C3 63EC'

```

```

INDEX:
      ORG  SES1H
      JP   CREATE
      JP   OPENX
      JP   INSERT
      JP   REMOVE
      JP   SEARCH
      JP   UPDATE
      JP   CLOSEX

```

```

C   INCLUDE CREATE.MAC
C   PAGE

```

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

๘๖

```

C ; *****
C ; *   PROGRAM ID   : CREATE.MAC   *
C ; *   AUTHOR      : Prasert Fungwanit   *
C ; *   DATE WRITTEN : 15 August 1986   *
C ; *   DATE REVISION : 1 May 1987     *
C ; *****
C ; I   FUNCTION :-                       J
C ; I   - Create file structure of index file by           I
C ; I     Create null data file                           I
C ; I     Create null index file                           I
C ; I   - Add header record to index file                   I
C ; *****
C
SE66' CD 5E72' C CREATE:      CALL CREATE_INIT
SE69' CD 5E8B' C              CALL CREATE_SYNT_ANAL
SE6C' CD 5E9E' C              CALL CREATE_EXEC
SE6F' C3 6454' C              JP   COMM_EXIT
C
C ;
C ;           This subroutine initialize working variable of CREATE command
C ;           by move blank to string variable and move zero to numeric variable
C ;
C
SE72' E5      C CREATE_INIT:  PUSH HL
SE73' 21 68A3' C              LD   HL,BLANK_CHAR
SE76' 01 0008' C              LD   BC,0008H
SE79' 11 68E7' C              LD   DE,ANAL_FILE_NAME
SE7C' ED B0   C              LDIR
SE7E' 21 68A3' C              LD   HL,BLANK_CHAR
SE81' 01 0003' C              LD   BC,0003H
SE84' 11 68EF' C              LD   DE,ANAL_FILE_TYPE
SE87' ED B0   C              LDIR
SE89' E1      C              POP  HL
SE8A' C9      C              RET
C
C ;
C ;           This subroutine scans CREATE command line in program text area
C ;           and analyzes individual field. If analysis is complete, command tail
C ;           of CREATE command will be kept at working variable, otherwise it
C ;           will output error message.
C ;
C ;
SE88' CD 648D' C CREATE_SYNT_ANAL: CALL CHK_FILE_NO
SE8E' CD 64C2' C              CALL CHK_FILE_SPEC
SE91' CD 6535' C              CALL CHK_REC_LEN
SE94' CD 6566' C              CALL CHK_KEY_POSI
SE97' CD 6597' C              CALL CHK_KEY_LEN
SE9A' 22 68E4' C              LD   (NEXT_COMM_ADDR),HL
SE9D' C9      C              RET
C
C ;
C ;           This subroutine executes CREATE command by
C ;           - Open & close data file that will create null data file

```



```

C ; - Open index file, write header record to index file &
C ; close index file that will create null index file
C ;
C ;
SE9E' C CREATE_EXEC:
C
C ; ----- Open & Close Data File -----
C
SE9E' 3E 00 C LD A,00H
SEA0' 32 0858 C LD (0858H),A
SEA3' 21 68E6' C LD HL,ANAL_FILE_INFO
SEA6' 11 68FB' C LD DE,WORK_FILE_INFO
SEA9' 01 0015 C LD BC,0015H
SEAC' ED 80 C LDIR
SEAE' CD 65D0' C CALL OPEN_RANDOM_FILE
SEB1' CD 6607' C CALL CLOSE_RANDOM_FILE
C
C ; ----- Open Index File -----
C
SEB4' 21 6912' C LD HL,STRING_NDX
SEB7' 11 6904' C LD DE,WORK_FILE_TYPE
SEBA' 01 0003 C LD BC,0003H
SEBD' ED 80 C LDIR
SEBF' ED 48 6908' C LD BC,(WORK_FILE_NO)
SEC3' 03 C INC BC
SEC4' ED 43 6908' C LD (WORK_FILE_NO),BC
SEC8' 01 0080 C LD BC,0080H
SECB' ED 43 690A' C LD (WORK_REC_LEN),BC
SECF' CD 65D0' C CALL OPEN_RANDOM_FILE
C
C ; ----- Set Up Header Record of Index File -----
C ; Then Close Index File
C
SED2' ED 48 68F5' C LD BC,(ANAL_REC_LEN) ; Set up header record with
SED6' ED 43 6A41' C LD (SAVE_REC_LEN),BC ; information analyzed from
SEDA' ED 48 68F7' C LD BC,(ANAL_KEY_POSI) ; command tail & initialize
SEDE' ED 43 6A43' C LD (SAVE_KEY_ADDR),BC ; file information as
SEE2' ED 48 68F9' C LD BC,(ANAL_KEY_LEN) ;
SEE6' ED 43 6A45' C LD (SAVE_KEY_LEN),BC ; Data record no = 1
SEEA' 01 0001 C LD BC,0001H ;
SEED' ED 43 6A47' C LD (SAVE_DATA_REC),BC ;
SEF1' ED 43 6A4F' C LD (SAVE_ROOT_BYTE),BC ; Root record no = 2
SEF5' 01 0002 C LD BC,0002H ; Root byte no = 1
SEF8' ED 43 6A4D' C LD (SAVE_ROOT_REC),BC ; No of active rec = 0
SEFC' 01 0000 C LD BC,0000H ;
SEFF' ED 43 6A51' C LD (SAVE_NO_REC),BC ;
C
SF03' ED 48 6908' C LD BC,(WORK_FILE_NO) ;
SF07' CD 6831' C CALL CAL_FILE_ADDR ;
SF0A' 21 00B2 C LD HL,00B2H ;
SF0D' 09 C ADD HL,BC ;
SF0E' 22 6A53' C LD (SAVE_INDX_ADDR),HL ;
SF11' 01 0002 C LD BC,0002H ; Write header node to index
SF14' ED 43 6A57' C LD (WORK_INDX_REC),BC ; file.
SF18' 01 0001 C LD BC,0001H ;
SF1B' ED 43 6A59' C LD (WORK_INDX_BYTE),BC

```

๘๘

```

SF1F' ED 43 6A5B' C LD (WORK_MOVE_POSI),BC
SF23' 3A 6A45' C LD A,(SAVE_KEY_LEN)
SF26' 3C C INC A
SF27' 47 C LD B,A
SF28' C6 08 C ADD A,08H
SF2A' 32 6A5D' C LD (WORK_MOVE_LEN),A
SF2D' 3E FF C LD A,OFFH
SF2F' 32 6A5E' C LD (WORK_INDX_FLAG),A
SF32' 21 6A5F' C LD HL,WORK_INDX_MODE
SF35' 36 FF C FILL_HIGHEST_KEY: LD (HL),OFFH
SF37' 23 C INC HL
SF38' 10 FB C DJNZ FILL_HIGHEST_KEY
SF3A' 06 08 C LD B,08H
SF3C' 36 00 C FILL_ZERO_LINK: LD (HL),00H
SF3E' 23 C INC HL
SF3F' 10 FB C DJNZ FILL_ZERO_LINK
SF41' CD 6721' C CALL WRITE_INDX_NODE
C
SF44' ED 4B 6A57' C LD BC,(WORK_INDX_REC) ;
SF48' ED 43 6A49' C LD (SAVE_INDX_REC),BC ;
SF4C' ED 4B 6A59' C LD BC,(WORK_INDX_BYTE) ;
SF50' ED 43 6A4B' C LD (SAVE_INDX_BYTE),BC ;
SF54' ED 4B 6908' C LD BC,(WORK_FILE_NO) ; Calculate index buffer address
SF58' CD 6831' C CALL CAL_FILE_ADDR ;
SF5B' 21 00B2 C LD HL,00B2H ;
SF5E' 09 C ADD HL,BC ; Now HL point to index buff addr
SF5F' 11 68E7' C LD DE,ANAL_FILE_NAME ;
SF62' EB C EX DE,HL ;
SF63' 01 0008 C LD BC,0008H ;
SF66' ED 80 C LDIR ; Move file specification
SF68' 21 6A41' C LD HL,SAVE_FILE_INFO ;
SF6B' 01 0012 C LD BC,0012H ;
SF6E' ED 80 C LDJR ; Move other file information.
SF70' 01 0001 C LD BC,0001H ; Set up header record no.
SF73' ED 43 6910' C LD (WORK_REC_NO),BC ; for writing.
SF77' CD 6612' C CALL WRITE_RANDOM_REC ;
SF7A' CD 6607' C CALL CLOSE_RANDOM_FILE ; Now close index file.
SF7D' C9 C RET
C INCLUDE OPENX.MAC
C PAGE

```

```

C ;
C ; *****
C ; *   PROGRAM ID   : OPENX.MAC   *
C ; *   AUTHOR      : Prasert Fungwanit *
C ; *   DATE WRITTEN : 20 August 1986 *
C ; *   DATE REVISION : 1 May 1987 *
C ; *****
C ; I   FUNCTION :-                I
C ; I       - Open index file which consist of 2 random file           I
C ; I           First file is data file                                 I
C ; I           Second file is index file                               I
C ; I       - Read header record from index file                         I
C ; *****
C
5F7E' CD 5F8A' C OPENX:          CALL OPEN_INIT
5F81' CD 5FA3' C                  CALL OPEN_SYNT_ANAL
5F84' CD 5FB3' C                  CALL OPEN_EXEC
5F87' C3 6454' C                  JP   COMM_EXIT
C
C ;
C ;           This subroutine initialize working variable of OPEN command
C ;           by move blank to string variable and move zero to numeric variable
C ;
C
5F8A' E5          C OPEN_INIT:    PUSH HL
5F8B' 21 6BA3'   C                  LD   HL,BLANK_CHAR
5F8E' 01 0000'   C                  LD   BC,0000H
5F91' 11 68E7'   C                  LD   DE,ANAL_FILE_NAME
5F94' ED 80      C                  LDIR
5F96' 21 6BA3'   C                  LD   HL,BLANK_CHAR
5F99' 01 0003'   C                  LD   BC,0003H
5F9C' 11 68EF'   C                  LD   DE,ANAL_FILE_TYPE
5F9F' ED 80      C                  LDIR
5FA1' E1          C                  POP  HL
5FA2' C9          C                  RET
C
C ;
C ;           This subroutine scans OPEN command line in program text area
C ;           and analyzes individual field. If analysis is complete, command tail
C ;           of OPEN command will be kept at working variable, otherwise it
C ;           will output error message.
C ;
C
5FA3' CD 645B'   C OPEN_SYNT_ANAL:  CALL CHK_FILE_MODE
5FA6' CD 648D'   C                  CALL CHK_FILE_NO
5FA9' CD 64C2'   C                  CALL CHK_FILE_SPEC
5FAC' CD 6535'   C                  CALL CHK_REC_LEN
5FAF' 22 68E4'   C                  LD   (NEXT_COMM_ADDR),HL
5FB2' C9          C                  RET
C
C ;
C ;           This subroutine executes OPEN command by
C ;           - Open data file & index file
C ;           - Read header record of index file

```

```

C ;
C
5FB3' C OPEN_EXEC:
C
C ; ----- Open Data File -----
C
5FB3' 3E 00 C LD A,00H
5FB5' 32 0858 C LD (0858H),A
5FB6' 21 68E6' C LD HL,ANAL_FILE_INFO
5FB8' 11 68F8' C LD DE,WORK_FILE_INFO
5FBE' 01 0015 C LD BC,0015H
5FC1' ED 80 C LDIR
5FC3' CD 65D0' C CALL OPEN_RANDOM_FILE
C
C ; ----- Open Index File -----
C
5FC6' 21 6912' C LD HL,STRING_HDX
5FC9' 11 6904' C LD DE,WORK_FILE_TYPE
5FCC' 01 0003 C LD BC,0003H
5FCF' ED 80 C LDIR
5FD1' ED 4B 6908' C LD BC,(WORK_FILE_NO)
5FD5' 03 C INC BC
5FD6' ED 43 6908' C LD (WORK_FILE_NO),BC
5FDA' 01 0080 C LD BC,0080H
5FDD' ED 43 690A' C LD (WORK_REC_LEN),BC
5FE1' CD 65D0' C CALL OPEN_RANDOM_FILE
C
C ; ----- Read Header Record from Index File -----
C ; & Set Up File Info. Table
C
5FE4' 01 0001 C LD BC,0001H ; Setup record no for reading
5FE7' ED 43 6910' C LD (WORK_REC_NO),BC ; header record from index
5FEB' CD 6634' C CALL READ_RANDOM_REC ; file.
C
5FEE' ED 4B 6908' C LD BC,(WORK_FILE_NO) ; Find index file buffer address
5FF2' CD 6831' C CALL CAL_FILE_ADDR ; of current file no.
5FF5' 21 00B2 C LD HL,00B2H ;
5FF8' 09 C ADD HL,BC ;
5FF9' 22 6A53' C LD (SAVE_INDX_ADDR),HL ;
5FFC' 01 0008 C LD BC,0008H ; Skip file spec to file info byte.
5FFF' 09 C ADD HL,BC ; Now HL points to file info byte.
6000' 11 6A41' C LD DE,SAVE_FILE_INFO ;
6003' 01 0012 C LD BC,0012H ;
6006' ED 80 C LDIR ; Move file info to save area.
6008' ED 4B 68F3' C LD BC,(ANAL_FILE_NO) ; Find data file buffer address
600C' CD 6831' C CALL CAL_FILE_ADDR ; of current file no.
600F' 21 00B1 C LD HL,00B1H ;
6012' 09 C ADD HL,BC ;
6013' ED 4B 6A43' C LD BC,(SAVE_KEY_ADDR) ;
6017' 09 C ADD HL,BC ;
6018' 22 6A43' C LD (SAVE_KEY_ADDR),HL ; Change content to absolute addr.
601B' ED 4B 6A45' C LD BC,(SAVE_KEY_LEN) ; Increment key length to actual
601F' 03 C INC BC ; key length.
6020' ED 43 6A45' C LD (SAVE_KEY_LEN),BC ;
C
6024' CD 681D' C CALL STORE_FILE_INFO ; Save file info into table

```

6027' C9

C
C
C
C

RET
INCLUDE INSERT.MAC
PAGE

๗๑



ศูนย์วิจัยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

```

C ; *****
C ; *      PROGRAM ID      : INSERT.MAC      *
C ; *      AUTHOR         : Prasert Fungwanit *
C ; *      DATE WRITTEN    : 25 September 1986 *
C ; *      DATE REVISION   : 2 May 1987      *
C ; *****
C ; I      FUNCTION :-                          I
C ; I      - Write record (from record buffer) to data file I
C ; I      with record no. by available record from index file I
C ; I      - Update header record to index file I
C ; I      on available record I
C ; I      - Create key from record I
C ; I      - Add key to index file I
C ; *****
C
6028' CD 6031' C INSERT: CALL INSERT_SYNT_ANAL
6028' CD 6038' C          CALL INSERT_EXEC
602E' C3 6454' C          JP COMM_EXIT
C
C ;
C ;      This subroutine scans INSERT command line in program text area
C ;      and analyzes individual field. If analysis is complete, command tail
C ;      of INSERT command will be kept at working variable, otherwise it
C ;      will output error message.
C ;
6031' CD 648D' C INSERT_SYNT_ANAL: CALL CHK_FILE_NO
6034' 22 68E4' C          LD (NEXT_COMM_ADDR),HL
6037' C9          C          RET
C
C ;
C ;      This subroutine executes INSERT command by
C ;      - Write record to data file
C ;      - Write key to index file
C ;
6038' C INSERT_EXEC:
C ; ----- Check Duplicated Key -----
6038' CD 680A' C          CALL LOAD_FILE_INFO
6038' CD 6656' C          CALL SEARCH_KEY
603E' 7B          C          LD A,E
603F' FE 00       C          CP 00H
6041' 20 06       C          JR NZ,ADJUST_INDEX
6043' 3E 48       C DUP_KEY: LD A,48H ; Find duplicated key
6045' 32 0858     C          LD (0858H),A
6048' C9          C          RET
C ; ----- Adjust Index of Index File -----
6049' 3E 00       C ADJUST_INDEX: LD A,00H
604B' 32 0858     C          LD (0858H),A

```

```

604E' ED 4B 6B67' C LD BC,(PARENT_INDY_REC)
6052' ED 43 6B84' C LD (ADJUST_INDY_REC),BC
6056' ED 4B 6B69' C LD BC,(PARENT_INDY_BYTE)
605A' ED 43 6B86' C LD (ADJUST_INDY_BYTE),BC
605E' 3A 6B68' C LD A,(PARENT_LINK_FLAG)
6061' 32 6B88' C LD (ADJUST_LINK_FLAG),A
6064' ED 4B 6A49' C LD BC,(SAVE_INDY_REC)
6068' ED 43 6B89' C LD (ADJUST_LINK_REC),BC
606C' ED 4B 6A4B' C LD BC,(SAVE_INDY_BYTE)
6070' ED 43 6B8B' C LD (ADJUST_LINK_BYTE),BC
6074' CD 66D4' C CALL ADJUST_LR_LINK
C
C ; ----- Write Record to Data File -----
C
6077' ED 4B 68F3' C LD BC,(ANAL_FILE_NO)
607B' ED 43 6908' C LD (WORK_FILE_NO),BC
607F' ED 4B 6A47' C LD BC,(SAVE_DATA_REC) ; Setup record no for writing
6083' ED 43 6910' C LD (WORK_REC_NO),BC
6087' CD 6612' C CALL WRITE_RANDOM_REC
C
C ; ----- Write Index to Index File -----
C
608A' ED 4B 6A49' C LD BC,(SAVE_INDY_REC) ; Set record no & byte no for
608E' ED 43 6A57' C LD (WORK_INDY_REC),BC ; new index of inserted record.
6092' ED 4B 6A4B' C LD BC,(SAVE_INDY_BYTE) ;
6096' ED 43 6A59' C LD (WORK_INDY_BYTE),BC ;
609A' 01 0001 C LD BC,0001H ; Set move position = 1, to start
609D' ED 43 6A5B' C LD (WORK_MOVE_POSI),BC ; from key.
60A1' 3A 6A45' C LD A,(SAVE_KEY_LEN) ; Set move length to move key,
60A4' C6 08 C ADD A,08H ; data rec no & link.
60A6' 32 6A5D' C LD (WORK_MOVE_LEN),A ;
60A9' 3E FF C LD A,OFFH ; Set index flag 'FF' to indicate
60AB' 32 6A5E' C LD (WORK_INDY_FLAG),A ; new index.
60AE' 2A 6A43' C LD HL,(SAVE_KEY_ADDR) ; Build index of inserted record.
60B1' 11 6A5F' C LD DE,WORK_INDY_NODE ;
60B4' ED 4B 6A45' C LD BC,(SAVE_KEY_LEN) ;
60B8' ED 80 C LDIR ; Move key to working index node.
60BA' 21 6A47' C LD HL,SAVE_DATA_REC ;
60BD' 01 0002 C LD BC,0002H ;
60C0' ED 80 C LDIR ; Set data file record no.
60C2' EB C EX DE,HL ; Fill null link to left link
60C3' 06 06 C LD B,06H ; & right link.
60C5' 36 00 C FILL_NULL_LINK: LD (HL),00H ;
60C7' 23 C INC HL ;
60C8' 10 FB C DJNZ FILL_NULL_LINK ;
C
60CA' CD 6721' C CALL WRITE_INDY_NODE ; Write new node to index file.
C
C ; ----- Store File Information to Table -----
C ; & Update Header Record of Index File
C
60CD' ED 4B 68F3' C LD BC,(ANAL_FILE_NO) ; Read header record from
60D1' 03 C INC BC ; index file to prepare for
60D2' ED 43 6908' C LD (WORK_FILE_NO),BC ; updating.
60D6' 01 0001 C LD BC,0001H ;
60D9' ED 43 6910' C LD (WORK_REC_NO),BC ;

```

```

60DD' CD 6634' C          CALL READ_RANDOM_REC ;
C
60E0' ED 4B 6A47' C      LD BC,(SAVE_DATA_REC) ; Update file information after
60E4' 03 C                INC BC ; inserting new record.
60E5' ED 43 6A47' C      LD (SAVE_DATA_REC),BC ; Increment next data file rec no.
60E9' ED 4B 6A57' C      LD BC,(WORK_INDX_REC) ;
60ED' ED 43 6A49' C      LD (SAVE_INDX_REC),BC ; Adjust next index file rec no.
60F1' ED 4B 6A59' C      LD BC,(WORK_INDX_BYTE) ;
60F5' ED 43 6A4B' C      LD (SAVE_INDX_BYTE),BC ; Adjust next index file byte no.
60F9' ED 4B 6A51' C      LD BC,(SAVE_NO_REC) ;
60FD' 03 C                INC BC ;
60FE' ED 43 6A51' C      LD (SAVE_NO_REC),BC ; Increment no rec of index file.
6102' CD 681D' C          CALL STORE_FILE_INFO ; Store file info back to table.
C
6105' 2A 6A53' C          LD HL,(SAVE_INDX_ADDR) ; Move file information to index
6108' 01 0011 C          LD BC,0011H ; file buffer to prepare for
610B' 09 C                ADD HL,BC ; updating.
610C' 11 6A47' C          LD DE,SAVE_DATA_REC ;
610F' E8 C                EX DE,HL ;
6110' 01 000C C          LD BC,000CH ;
6113' ED 80 C            LDIR ;
6115' CD 6612' C          CALL WRITE_RANDOM_REC ; Rewrite header rec of index file.
C
6118' C9 C                RET
C                INCLUDE SEARCH.MAC
C                PAGE

```

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย


```

C
C ; *****
C ; *   PROGRAM ID   : SEARCH.MAC   *
C ; *   AUTHOR      : Prasert Fungwanit   *
C ; *   DATE WRITTEN : 12 October 1986   *
C ; *   DATE REVISION : 3 May 1987      *
C ; *****
C ; I   FUNCTION :-                   I
C ; I   - Create key from record       I
C ; I   - Read record no. from index file I
C ; I       on that key                 I
C ; I   - Read record from data file   I
C ; I       on that record no.         I
C ; *****
C
6119' CD 6122' C SEARCH:      CALL SEARCH_SYNT_ANAL
611C' CD 6129' C              CALL SEARCH_EXEC
611F' C3 6454' C              JP   COMM_EXIT
C
C
C ;
C ;       This subroutine scans SEARCH command line in program text area
C ;       and analyzes individual field. If analysis is complete, command tail
C ;       of SEARCH command will be kept at working variable, otherwise it
C ;       will output error message.
C ;
C
6122' CD 648D' C SEARCH_SYNT_ANAL: CALL CHK_FILE_NO
6125' 22 68E4' C              LD   (NEXT_COMM_ADDR),HL
6128' C9          C              RET
C
C ;
C ;       This subroutine executes SEARCH command by
C ;       - Create key from record
C ;       - Read record no. from index file
C ;       - Read record from data file by record no.
C ;
C
6129'          C SEARCH_EXEC:
C ; ----- Search for Existing Index -----
C
6129' CD 680A' C              CALL LOAD_FILE_INFO
612C' CD 6656' C              CALL SEARCH_KEY
612F' 78          C              LD   A,E
6130' FE 00       C              CP   00H
6132' 28 06       C              JR   Z,SEARCH_KEY_FOUND
6134' 3E 49       C              LD   A,49H
6136' 32 0858     C              LD   (0858H),A
6139' C9          C              RET
C
C ; ----- Restore Data from Data File -----
C
613A' 3E 00       C SEARCH_KEY_FOUND: LD   A,00H
613C' 32 0858     C              LD   (0858H),A

```

613F' ED 4B 68F3' C
 6143' ED 43 6908' C
 6147' 21 6ASF' C
 614A' ED 4B 6A45' C
 614E' 09 C
 614F' 4E C
 6150' 23 C
 6151' 46 C
 6152' ED 43 6910' C
 6156' CB 6634' C
 C
 6159' C9 C
 C
 C

LD BC,(ANAL_FILE_NO)
 LD (WORK_FILE_NO),BC
 LD HL,WORK_INDX_NODE
 LD BC,(SAVE_KEY_LEN)
 ADD HL,BC
 LD C,(HL)
 INC HL
 LD B,(HL)
 LD (WORK_REC_NO),BC
 CALL READ_RANDOM_REC

๙๖

RET

INCLUDE UPDATE.MAC

PAGE



ศูนย์วิทยทรัพยากร
 จุฬาลงกรณ์มหาวิทยาลัย

```

C ; *****
C ; # PROGRAM ID : UPDATE.MAC #
C ; # AUTHOR : Prasert Fungwanit #
C ; # DATE WRITTEN : 12 October 1986 #
C ; # DATE REVISION : 3 May 1987 #
C ; *****
C ; I FUNCTION :- I
C ; I - Create key from record I
C ; I - Read record no. from index file I
C ; I on that key I
C ; I - Rewrite record to data file I
C ; *****
C
615A' CD 6163' C UPDATE: CALL UPDATE_SYNT_ANAL
615D' CD 616A' C CALL UPDATE_EXEC
6160' C3 6454' C JP COMM_EXIT
C
C ;
C ; This subroutine scans UPDATE command line in program text area
C ; and analyzes individual field. If analysis is complete, command tail
C ; of UPDATE command will be kept at working variable, otherwise it
C ; will output error message.
C ;
C
6163' CD 648D' C UPDATE_SYNT_ANAL: CALL CHK_FILE_NO
6166' 22 68E4' C LD (NEXT_COMM_ADDR),HL
6169' C9 C RET
C
C ;
C ; This subroutine executes UPDATE command by
C ; - Create key from record
C ; - Read record no. from index file
C ; - Rewrite record to data file
C ;
C
616A' C UPDATE_EXEC:
C ; ----- Search for Existing Index -----
C
616A' CD 680A' C CALL LOAD_FILE_INFO
616D' CD 6656' C CALL SEARCH_KEY
6170' 7B C LD A,E
6171' FE 00 C CP 00H
6173' 28 06 C JR Z,UPDATE_KEY_FOUND
6175' 3E 49 C LD A,49H
6177' 32 0858 C LD (0858H),A
617A' C9 C RET
C
C ; ----- Rewrite Record to Data File -----
C
617B' 3E 00 C UPDATE_KEY_FOUND: LD A,00H
617D' 32 0858 C LD (0858H),A
6180' ED 4B 68F3' C LD BC,(ANAL_FILE_NO)

```

6184' ED 43 6908' C
 6188' 21 6A5F' C
 618R' ED 4B 6A45' C
 818F' 09 C
 6190' 4E C
 6191' 23 C
 6192' 46 C
 6193' ED 43 6910' C
 6197' CD 6612' C
 C
 619A' C9 C
 C
 C

LD (WORK_FILE_NO),BC
 LD HL,WORK_INDX_NODE
 LD RC,(SAVE_KEY_LEN)
 ADD HL,BC
 LD C,(HL)
 INC HL
 LD B,(HL)
 LD (WORK_REC_NO),BC
 CALL WRITE_RANDOM_REC

๗๘

RET

INCLUDE REMOVE.MAC

PAGE



ศูนย์วิทยทรัพยากร
 จุฬาลงกรณ์มหาวิทยาลัย

```

C ; *****
C ; *   PROGRAM ID   : REMOVE.MAC   *
C ; *   AUTHOR      : Prasert Fungwanit *
C ; *   DATE WRITTEN : 12 October 1986 *
C ; *   DATE REVISION : 3 May 1987 *
C ; *****
C ; I   FUNCTION :-                I
C ; I   - Create key from record    I
C ; I   - Read record no. from index I
C ; I     on that key                I
C ; I   - Rewrite zero to data record I
C ; I     of data file                I
C ; I   - Remove index from index file I
C ; I   - Update header record of index I
C ; *****
C
619B' CD 61A4' C REMOVE:          CALL REMOVE_SYNT_ANAL
619E' CD 61AB' C                  CALL REMOVE_EXEC
61A1' C3 6454' C                  JP COMM_EXIT
C
C ;
C ;   This subroutine scans REMOVE command line in program text area
C ;   and analyzes individual field. If analysis is complete, command tail
C ;   of REMOVE command will be kept at working variable, otherwise it
C ;   will output error message.
C ;
C
61A4' CD 648D' C REMOVE_SYNT_ANAL: CALL CHK_FILE_NO
61A7' CD 22 68E4' C                  LD (NEXT_COMM_ADDR),HL
61AA' C9          C                  RET
C
C ;
C ;   This subroutine executes REMOVE command by
C ;   - Create key from record
C ;   - Read record no. from index file
C ;   - Rewrite zero to data record of data file
C ;   - Remove index from index file
C ;   - Update header record of index file
C ;
C
61AB' C REMOVE_EXEC:
C ; ----- Search for Existing Index -----
C
61AB' CD 680A' C                  CALL LOAD_FILE_INFO
61AE' CD 6656' C                  CALL SEARCH_KEY
61B1' 7B          C                  LD A,E
61B2' FE 00       C                  CP 00H
61B4' 28 06       C                  JR 2,REMOVE_KEY_FOUND
61B6' 3E 49       C                  LD A,49H
61B8' 32 0858     C                  LD (0858H),A
61BB' C9          C                  RET
C
C ; ----- Rewrite '00' to Record to Data File -----

```

```

C
618C' 3E 00 C REMOVE_KEY_FOUND: LD A,00H
618E' 32 0858 C LD (0858H),A
61C1' ED 48 68F3' C LD BC,(ANAL_FILE_NO)
61C5' ED 43 6908' C LD (WORK_FILE_NO),BC
61C9' CD 6831' C CALL CAL_FILE_ADDR
61CC' 21 00B2 C LD HL,00B2H
61CF' 09 C ADD HL,BC
61D0' 01 0001 C LD BC,0001H
61D3' 36 00 C FILL_ZERO_BYTE: LD (HL),00H
61D5' 23 C INC HL
61D6' 03 C INC BC
61D7' E5 C PUSH HL
61D8' 2A 6A41' C LD HL,(SAVE_REC_LEN)
61DB' A7 C AND A
61DC' ED 42 C SBC HL,BC
61DE' 38 03 C JR C,FILL_ALL_ZERO
61E0' E1 C POP HL
61E1' 18 F0 C JR FILL_ZERO_BYTE
61E3' E1 C FILL_ALL_ZERO: POP HL
61E4' 21 6A5F' C LD HL,WORK_INDY_HODE
61E7' ED 48 6A45' C LD BC,(SAVE_KEY_LEN)
61EB' 09 C ADD HL,BC
61EC' 4E C LD C,(HL)
61ED' 23 C INC HL
61EE' 46 C LD B,(HL)
61EF' ED 43 6910' C LD (WORK_REC_NO),BC
61F3' CD 6612' C CALL WRITE_RANDOM_REC
C
C ; ----- Remove Index from Index File -----
C
61F6' CD 67E1' C CALL CREATE_INDY_LINK
61F9' 3A 686E' C CHK_INDY_LLINK: LD A,(WORK_LLINK_BYTE) ; Remove Case #1
61FC' FE 00 C CP 00H ;
61FE' 20 2C C JR NZ,CHK_INDY_RLINK ; Left link of removed index
6200' ED 48 6867' C LD BC,(PARENT_INDY_REC) ; is null.
6204' ED 43 6884' C LD (ADJUST_INDY_REC),BC ;
6208' ED 48 6869' C LD BC,(PARENT_INDY_BYTE) ;
620C' ED 43 6886' C LD (ADJUST_INDY_BYTE),BC ;
6210' 3A 6868' C LD A,(PARENT_LINK_FLAG) ;
6213' 32 6888' C LD (ADJUST_LINK_FLAG),A ;
6216' ED 48 6870' C LD BC,(WORK_RLINK_REC) ;
621A' ED 43 6889' C LD (ADJUST_LINK_REC),BC ;
621E' ED 48 6872' C LD BC,(WORK_RLINK_BYTE) ;
6222' ED 43 6888' C LD (ADJUST_LINK_BYTE),BC ;
6226' CD 66D4' C CALL ADJUST_LR_LINK ;
6229' C3 63B9' C JP ADJUST_HEADER ;
C
622C' 3A 6872' C CHK_INDY_RLINK: LD A,(WORK_RLINK_BYTE) ; Remove Case #2
622F' FE 00 C CP 00H ;
6231' 20 2C C JR NZ,NONNULL_LR_LINK ; Right link of removed index
6233' ED 48 6867' C LD BC,(PARENT_INDY_REC) ; is null.
6237' ED 43 6884' C LD (ADJUST_INDY_REC),BC ;
623B' ED 48 6869' C LD BC,(PARENT_INDY_BYTE) ;
623F' ED 43 6886' C LD (ADJUST_INDY_BYTE),BC ;
6243' 3A 6868' C LD A,(PARENT_LINK_FLAG) ;

```



ศูนย์วิทยบริการ
สงครณาลัย

```

6246' 32 6888' C LD (ADJUST_LINK_FLAG),A ;
6249' ED 4B 686C' C LD BC,(WORK_LLINK_REC) ;
624D' ED 43 6889' C LD (ADJUST_LINK_REC),BC ;
6251' ED 4B 686E' C LD BC,(WORK_LLINK_BYTE) ;
6255' ED 43 6888' C LD (ADJUST_LINK_BYTE),BC ;
6259' CD 66D4' C CALL ADJUST_LR_LINK ;
625C' C3 63B9' C JP ADJUST_HEADER ;
C
625F' ED 4B 6870' C NONNULL_LR_LINK: LD BC,(WORK_RLINK_REC) ; Remove Case #3
6263' ED 43 6A57' C LD (WORK_INDX_REC),BC ;
6267' ED 4B 6872' C LD BC,(WORK_RLINK_BYTE) ; Left & right link of removed
626B' ED 43 6A59' C LD (WORK_INDX_BYTE),BC ; index are not null and left
626F' CD 6796' C CALL READ_INDX_NODE ; link of the index pointed
6272' 21 686C' C LD HL,WORK_INDX_LINK ; by right link of removed index
6275' 11 6874' C LD DE,SAVE_INDX_LINK ; is null link.
6278' 01 0008 C LD BC,0008H ;
627B' ED 80 C LDIR ;
627D' 21 6870' C LD HL,WORK_RLINK_REC ;
6280' 11 687C' C LD DE,LAST_INDX_REC ;
6283' 01 0004 C LD BC,0004H ;
6286' ED 80 C LDIR ;
6288' CD 67E1' C CALL CREATE_INDX_LINK ;
628B' 3A 686E' C LD A,(WORK_LLINK_BYTE) ;
628E' FE 00 C CP 00H ;
6290' 20 54 C JR NZ,NONNULL_L_LINK ;
6292' ED 4B 6878' C LD BC,(SAVE_RLINK_REC) ;
6296' ED 43 6884' C LD (ADJUST_INDX_REC),BC ;
629A' ED 4B 687A' C LD BC,(SAVE_RLINK_BYTE) ;
629E' ED 43 6886' C LD (ADJUST_INDX_BYTE),BC ;
62A2' 3E 01 C LD A,01H ;
62A4' 32 6888' C LD (ADJUST_LINK_FLAG),A ;
62A7' ED 4B 6874' C LD BC,(SAVE_LLINK_REC) ;
62AB' ED 43 6889' C LD (ADJUST_LINK_REC),BC ;
62AF' ED 4B 6876' C LD BC,(SAVE_LLINK_BYTE) ;
62B3' ED 43 688B' C LD (ADJUST_LINK_BYTE),BC ;
62B7' CD 66D4' C CALL ADJUST_LR_LINK ; Adjust link of children node
C ; of removed index.
62BA' ED 4B 6867' C LD BC,(PARENT_INDX_REC) ;
62BE' ED 43 6884' C LD (ADJUST_INDX_REC),BC ;
62C2' ED 4B 6869' C LD BC,(PARENT_INDX_BYTE) ;
62C6' ED 43 6886' C LD (ADJUST_INDX_BYTE),BC ;
62CA' 3A 686B' C LD A,(PARENT_LINK_FLAG) ;
62CD' 32 6888' C LD (ADJUST_LINK_FLAG),A ;
62D0' ED 4B 6878' C LD BC,(SAVE_RLINK_REC) ;
62D4' ED 43 6889' C LD (ADJUST_LINK_REC),BC ;
62D8' ED 4B 687A' C LD BC,(SAVE_RLINK_BYTE) ;
62DC' ED 43 688B' C LD (ADJUST_LINK_BYTE),BC ;
62E0' CD 66D4' C CALL ADJUST_LR_LINK ; Adjust link of parent node
62E3' C3 63B9' C JP ADJUST_HEADER ; of removed index.
C
62E6' ED 4B 686C' C NONNULL_L_LINK: LD BC,(WORK_LLINK_REC) ; Remove Case #4
62EA' ED 43 6A57' C LD (WORK_INDX_REC),BC ;
62EE' ED 4B 686E' C LD BC,(WORK_LLINK_BYTE) ; Left & right link of removed
62F2' ED 43 6A59' C LD (WORK_INDX_BYTE),BC ; index are not null and left
62F6' CD 6796' C CALL READ_INDX_NODE ; link of the index pointed
62F9' 21 6A5F' C LD HL,WORK_INDX_NODE ; by right link of removed index
    
```

```

62FC' ED 4B 6A45' C LD RC,(SAVE_KEY_LEN) ; is not null link
6300' 09 C ADD HL,BC ;
6301' 01 0004 C LD BC,0004H ;
6304' 09 C ADD HL,BC ;
6305' 7E C LD A,(HL) ;
6306' FE 00 C CP 00H ;
6308' 28 10 C JR 2,FIND_NULL_LLINK ;
630A' 21 6B6C' C LD HL,WORK_LLINK_REC ;
630D' 11 6B7C' C LD DE,LAST_INDX_REC ;
6310' 01 0004 C LD BC,0004H ;
6313' ED 80 C LDIR ;
6315' CD 67E1' C CALL CREATE_INDX_LINK ;
6318' 18 CC C JR NONNULL_L_LINK ;
631A' 21 6B6C' C FIND_NULL_LLINK: LD HL,WORK_LLINK_REC ;
631D' 11 6B80' C LD DE,SUCC_INDX_REC ;
6320' 01 0004 C LD BC,0004H ;
6323' ED 80 C LDIR ;
6325' CD 67E1' C CALL CREATE_INDX_LINK ;
6328' ED 4B 6B7C' C LD BC,(LAST_INDX_REC) ;
632C' ED 43 6B84' C LD (ADJUST_INDX_REC),BC ;
6330' ED 4B 6B7E' C LD BC,(LAST_INDX_BYTE) ;
6334' ED 43 6B86' C LD (ADJUST_INDX_BYTE),BC ;
6338' 3E 01 C LD A,01H ;
633A' 32 6B88' C LD (ADJUST_LINK_FLAG),A ;
633D' ED 4B 6B70' C LD BC,(WORK_RLINK_REC) ;
6341' ED 43 6B89' C LD (ADJUST_LINK_REC),BC ;
6345' ED 4B 6B72' C LD BC,(WORK_RLINK_BYTE) ;
6349' ED 43 6B88' C LD (ADJUST_LINK_BYTE),BC ;
634D' CD 66D4' C CALL ADJUST_LR_LINK ;
6350' ED 4B 6B80' C LD BC,(SUCC_INDX_REC) ;
6354' ED 43 6B84' C LD (ADJUST_INDX_REC),BC ;
6358' ED 4B 6B82' C LD BC,(SUCC_INDX_BYTE) ;
635C' ED 43 6B86' C LD (ADJUST_INDX_BYTE),BC ;
6360' 3E 01 C LD A,01H ;
6362' 32 6B88' C LD (ADJUST_LINK_FLAG),A ;
6365' ED 4B 6B74' C LD BC,(SAVE_LLINK_REC) ;
6369' ED 43 6B89' C LD (ADJUST_LINK_REC),BC ;
636D' ED 4B 6B76' C LD BC,(SAVE_LLINK_BYTE) ;
6371' ED 43 6B88' C LD (ADJUST_LINK_BYTE),BC ;
6375' CD 66D4' C CALL ADJUST_LR_LINK ;
6378' 3E 02 C LD A,02H ;
637A' 32 6B88' C LD (ADJUST_LINK_FLAG),A ;
637D' ED 4B 6B78' C LD BC,(SAVE_RLINK_REC) ;
6381' ED 43 6B89' C LD (ADJUST_LINK_REC),BC ;
6385' ED 4B 6B7A' C LD BC,(SAVE_RLINK_BYTE) ;
6389' ED 43 6B88' C LD (ADJUST_LINK_BYTE),BC ;
638D' CD 66D4' C CALL ADJUST_LR_LINK ;
6390' ED 4B 6B67' C LD BC,(PARENT_INDX_REC) ;
6394' ED 43 6B84' C LD (ADJUST_INDX_REC),BC ;
6398' ED 4B 6B69' C LD BC,(PARENT_INDX_BYTE) ;
639C' ED 43 6B86' C LD (ADJUST_INDX_BYTE),BC ;
63A0' 3A 6B6B' C LD A,(PARENT_LINK_FLAG) ;
63A3' 32 6B88' C LD (ADJUST_LINK_FLAG),A ;
63A6' ED 4B 6B80' C LD BC,(SUCC_INDX_REC) ;
63AA' ED 43 6B89' C LD (ADJUST_LINK_REC),BC ;
63AE' ED 4B 6B82' C LD BC,(SUCC_INDX_BYTE) ;

```



```

63B2' ED 43 688B' C LD (ADJUST_LINK_BYTE),BC ;
63B6' CD 66D4' C CALL ADJUST_LR_LINK ;
C
C ; ----- Store File Information to Table -----
C ; & Update Header Record of Index File
C
63B9' ED 4B 68F3' C ADJUST_HEADER: LD BC,(ANAL_FILE_NO) ; Read header record from
63BD' 03 C INC BC ; index file to prepare for
63BE' ED 43 6908' C LD (WORK_FILE_NO),BC ; updating.
63C2' 01 0001 C LD BC,0001H ;
63C5' ED 43 6910' C LD (WORK_REC_NO),BC ;
63C9' CD 6634' C CALL READ_RANDOM_REC ;
C
63CC' ED 4B 6A51' C LD BC,(SAVE_NO_REC) ; Update file information after
63D0' 0B C DEC BC ; removing old record.
63D1' ED 43 6A51' C LD (SAVE_NO_REC),BC ; Decrement no. of record.
63D5' CD 681D' C CALL STORE_FILE_INFO ; Store file info back to table.
C
63D8' 2A 6A53' C LD HL,(SAVE_INDX_ADDR) ; Move file information to index
63DB' 01 0011 C LD BC,0011H ; file buffer to prepare for
63DE' 09 C ADD HL,BC ; updating.
63DF' 11 6A47' C LD DE,SAVE_DATA_REC ;
63E2' EB C EX DE,HL ;
63E3' 01 000C C LD BC,000CH ;
63E6' ED 80 C LDIR ;
63E8' CD 6612' C CALL WRITE_RANDOM_REC ; Rewrite header rec of index file.
C
63EB' C9 C RET
C
C INCLUDE CLOSEX.MAC
C
C PAGE

```

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

```

C
C ; *****
C ; *      PROGRAM ID   :  CLOSEX.MAC      *
C ; *      AUTHOR      :  Prasert Fungwanit *
C ; *      DATE WRITTEN :  27 August  1986  *
C ; *      DATE REVISION :  6 May    1987  *
C ; *****
C ; I      FUNCTION :-                      I
C ; I      - Close index file which consist of I
C ; I          Data file                    I
C ; I          Index file                    I
C ; I      - Rewrite header record to index file I
C ; *****
C
63EC'  CD 63FD' C CLOSEX:      CALL CLOSEX_INIT
63EF'  CD 640E' C              CALL CLOSEX_FTYPE_ANAL
63F2'  38 03   C              JR  C,CLOSEX_RET      ; It isn't the index file
63FA'  CD 6424' C              CALL CLOSEX_EXEC
C ; ----- old last close routine -----
63F7'  C1     C CLOSEX_RET:   POP  BC
63F8'  E1     C              POP  HL
63F9'  7E     C              LD   A,(HL)      ; next program text
63FA'  C3 5538 C              JP   5538H      ; Back to close routine
C
C ;
C ;      This subroutine initialize working variable of CLOSEX command
C ;      by compute address of index file type
C ;      and save file no. of index file from address OCD4H
C ;
63FD'  ED 4B OCD4 C CLOSEX_INIT:  LD   BC,(OCD4H) ; file no. of file.dat
6401'  03       C              INC  BC      ; file no. of file.ndx
6402'  ED 43 6908' C              LD   (WORK_FILE_NO),BC ; save file no. of file.ndx
6406'  CD 6831'  C              CALL CAL_FILE_ADDR ; compute file address of file.ndx
6409'  21 000A  C              LD   HL,000AH
640C'  09       C              ADD  HL,BC   ; hl = ftype address of file.ndx
640D'  C9       C              RET
C
C ;
C ;      This subroutine check file type of next file = "ndx" ?
C ;      and send back the message code
C ;      - carry flag off for file type = "ndx"
C ;      - carry flag on  for file type not = "ndx"
C ;
640E'  7E     C CLOSEX_FTYPE_ANAL: LD  A,(HL)
640F'  FE 4E   C              CP   4EH      ; "n"
6411'  20 0F   C              JR  NZ,CLOSEX_END_ANAL ; File type not = ndx
6413'  23     C              INC  HL
6414'  7E     C              LD   A,(HL)
6415'  FE 44   C              CP   44H      ; "d"
6417'  20 09   C              JR  NZ,CLOSEX_END_ANAL ; File type not = ndx
6419'  23     C              INC  HL
641A'  7E     C              LD   A,(HL)

```

```

641B' FE 58      C          CP 58H          ; "x"
641D' C2 6422'   C          JP NZ,CLOSEX_END_ANAL ; File type not = ndx
6420' B7         C          OR A          ; Reset carry flag
6421' C9         C          RET

C
6422' 37         C CLOSEX_END_ANAL: SCF          ; Set carry flag
6423' C9         C          RET

C
C ;
C ;
C ;          This subroutine executes CLOSEX command by
C ;          - Write header record
C ;          - Close index file
C ;
C ;
C ; ----- write header record -----
C ;
6424' 01 0001   C CLOSEX_EXEC: LD BC,0001H
6427' ED 43 6910' C          LD (WORK_REC_NO),BC
642B' CD 6634'   C          CALL READ_RANDOM_REC
642E' ED 48 6908' C          LD BC,(WORK_FILE_NO)
6432' 0B         C          DEC BC
6433' CD 683A'   C          CALL CAL_TABL_ADDR ; compute header record address
6436' 01 0006   C          LD BC,0006H
6439' 09         C          ADD HL,BC
643A' E5         C          PUSH HL ; save header record address
643B' ED 48 6908' C          LD BC,(WORK_FILE_NO)
643F' CD 6831'   C          CALL CAL_FILE_ADDR ; compute file address of file.ndx
6442' 21 00C3   C          LD HL,00C3H
6445' 09         C          ADD HL,BC ; compute record buffer of file.ndx
6446' D1         C          POP DE ; restore header record address
6447' EB         C          EX DE,HL
6448' 01 000C   C          LD BC,000CH
644B' ED B0         C          LDIR ; move header record to rec. buff.
644D' CD 6612'   C          CALL WRITE_RANDOM_REC ; write header record

C ; ----- close index file -----
C ;
6450' CD 6607'   C          CALL CLOSE_RANDOM_FILE ; close file.ndx
6453' C9         C          RET

C
C          INCLUDE SHARE.MAC
C
6454' CD 189A   C COMM_EXIT: CALL 189AH
6457' 2A 68E4'   C          LD HL,(NEXT_COMM_ADDR)
645A' C9         C          RET

C
C          PAGE

```

```

C
C ;
C ;           This module consists of several subroutine, each subroutine is
C ;           design to analyze individual field of the command
C ;
C
C ; ----- Analyze File Mode -----
645B' C CHK_FILE_MODE:
645B' CD 685F' C GET_FILE_MODE:      CALL GET_TOKEN
645E' E5           C           PUSH HL
645F' 7B           C CHECK_FILE_MODE: LD A,E
6460' FE FF       C           CP OFFH
6462' 20 05       C           JR NZ,FILE_MODE_EXIST
6464' 1E 02       C           LD E,02H
6466' C3 6845'    C           JP ERROR_ROUTINE ; Syntax error
6469' 21 688D'    C FILE_MODE_EXIST: LD HL,TOKEN
646C' 7E           C           LD A,(HL)
646D' FE 22       C           CP 22H ; Check file name whether string
646F' 28 05       C           JR Z,QUOTE_HAVE
6471' 1E 0D       C           LD E,0DH
6473' C3 6845'    C           JP ERROR_ROUTINE ; Type mismatch
6476' 23           C QUOTE_HAVE: INC HL
6477' 7E           C           LD A,(HL)
6478' FE 58       C           CP 58H
647A' 28 0C       C           JR Z,MODE_X
647C' E1           C           POP HL ; Last prog text addr
647D' C1           C           POP BC ; Return addr
647E' C1           C           POP BC ; Return addr
647F' 01 0004     C           LD BC,0004H
6482' A7           C           AND A
6483' ED 42       C           SBC HL,BC
6485' C3 5971     C           JP 5971H ; Jump to org OPEN
6488' 32 68F2'    C MODE_X: LD (ANAL_FILE_MODE),A
648B' E1           C           POP HL
648C' C9           C           RET
C
C ; ----- Analyze File No. -----
648D' C CHK_FILE_NO:
648D' CD 685F' C GET_FILE_NO:      CALL GET_TOKEN
6490' E5           C           PUSH HL
6491' 7B           C CHECK_FILE_NO: LD A,E
6492' FE FF       C           CP OFFH
6494' 20 05       C           JR NZ,FILE_NO_EXIST
6496' 1E 02       C           LD E,02H
6498' C3 6845'    C           JP ERROR_ROUTINE ; Syntax Error
649B' 21 688D'    C FILE_NO_EXIST: LD HL,TOKEN
649E' 7E           C           LD A,(HL)
649F' FE 23       C           CP 23H ; Check '#'
64A1' 20 01       C           JR NZ,NO_SHARP_CHAR
64A3' 23           C           INC HL
64A4' CD 68BF'    C NO_SHARP_CHAR: CALL CONV_NUM
64A7' 7B           C           LD A,E
64A8' FE FF       C           CP OFFH ; If convert complete, continue
64AA' 20 05       C           JR NZ,NUMERIC_FILE_NO
64AC' 1E 0D       C           LD E,0DH
    
```

```

64AE' C3 6845' C JP ERROR_ROUTINE ; Type Mismatch
64B1' 3A 0893 C NUMERIC_FILE_NO: LD A,(0893H) ; Load A with max. file no.
64B4' B9 C CP C
64B5' 30 05 C JR NC,VALID_FILE_NO
64B7' 1E 34 C LD E,34H
64B9' C3 6845' C JP ERROR_ROUTINE ; Bad file number
64BC' ED 43 68F3' C VALID_FILE_NO: LD (ANAL_FILE_NO),BC
64C0' E1 C POP HL
64C1' C9 C RET
C
C ; ----- Analyze File Specification -----
64C2' C CHK_FILE_SPEC:
64C2' CD 685F' C GET_FILE_SPEC: CALL GET_TOKEN
64C5' E5 C PUSH HL
64C6' 7B C CHECK_FILE_SPEC: LD A,E
64C7' FE FF C CP OFFH
64C9' 20 05 C JR NZ,FILE_SPEC_EXIST
64CB' 1E 02 C LD E,02H
64CD' C3 6845' C JP ERROR_ROUTINE ; Syntax error
64D0' 21 688D' C FILE_SPEC_EXIST: LD HL,TOKEN
64D3' 7E C LD A,(HL)
64D4' FE 22 C CP 22H ; Check file name whether string
64D6' 28 05 C JR Z,QUOTE_EXIST
64D8' 1E 0D C LD E,0DH
64DA' C3 6845' C JP ERROR_ROUTINE ; Type mismatch
64DD' C QUOTE_EXIST:
64DD' 23 C INC HL
64DE' 23 C INC HL
64DF' 7E C LD A,(HL)
64E0' FE 3A C CP 3AH ; Check file drive
64E2' 28 0D C JR Z,GET_FILE_DRIVE ; There is drive spec.
64E4' 3A 0004 C LD A,(0004H)
64E7' E6 0F C AND 0FH
64E9' 3C C INC A
64EA' 32 68E6' C LD (ANAL_CUR_DRIVE),A ; Set default drive
64ED' 28 C DEC HL ; There is not drive spec.
64EE' 28 C DEC HL
64EF' 18 08 C JR INIT_GET_FILE_NAME
C
64F1' 28 C GET_FILE_DRIVE: DEC HL
64F2' 7E C LD A,(HL)
64F3' E6 0F C AND 0FH
64F5' 32 68E6' C LD (ANAL_CUR_DRIVE),A
64F8' 23 C INC HL ; Colon byte
C
64F9' C INIT_GET_FILE_NAME:
64F9' 0E 0D C LD C,0DH ; No. char of file name
64FB' 11 68E7' C LD DE,ANAL_FILE_NAME
64FE' 23 C GET_FILE_NAME: INC HL
64FF' 7E C LD A,(HL)
6500' FE 2E C CP 2EH ; Check "."
6502' 28 15 C JR Z,VALID_FILE_NAME
6504' FE 22 C CP 22H
6506' 28 11 C JR Z,VALID_FILE_NAME
6508' FE 20 C CP 20H
650A' 28 0D C JR Z,VALID_FILE_NAME
    
```



ศูนย์วิทยบริการ
 วิทยาลัย

```

650C' 12 C LD (DE),A
650D' 13 C INC DE
650E' 0C C INC C
650F' 79 C LD A,C
6510' FE 09 C CP 09H
6512' 38 EA C JR C,GET_FILE_NAME
6514' 1E 40 C LD E,40H
6516' C3 6845' C JP ERROR_ROUTINE ; Bad file name
6519' 0E 00 C VALID_FILE_NAME: LD C,00H
651B' 11 68EF' C LD DE,ANAL_FILE_TYPE
651E' 23 C GET_FILE_TYPE: INC HL
651F' 7E C LD A,(HL)
6520' FE 22 C CP 22H
6522' 28 0F C JR Z,VALID_FILE_TYPE
6524' FE 20 C CP 20H
6526' 28 0B C JR Z,VALID_FILE_TYPE
6528' 12 C LD (DE),A
6529' 13 C INC DE
652A' 0C C INC C
652B' 79 C LD A,C
652C' FE 03 C CP 03H
652E' 28 03 C JR Z,VALID_FILE_TYPE
6530' C3 651E' C JP GET_FILE_TYPE
6533' E1 C VALID_FILE_TYPE: POP HL
6534' C9 C RET
C
C ; ----- Analyze Record Length -----
6535' C CHK_REC_LEN:
6535' CD 685F' C GET_REC_LEN: CALL GET_TOKEN
6538' E5 C PUSH HL
6539' 7B C CHECK_REC_LEN: LD A,E
653A' FE FF C CP OFFH
653C' 20 05 C JR NZ,REC_LEN_EXIST
653E' 1E 02 C LD E,02H
6540' C3 6845' C JP ERROR_ROUTINE ; Syntax error
6543' 21 688D' C REC_LEN_EXIST: LD HL,TOKEN
6546' CD 68BF' C CALL CONV_NUM
6549' 7B C LD A,E
654A' FE FF C CP OFFH
654C' 20 05 C JR NZ,NUMERIC_REC_LEN
654E' 1E 0D C LD E,0DH
6550' C3 6845' C JP ERROR_ROUTINE ; Type mismatch
6553' 2A 0CBA C NUMERIC_REC_LEN: LD HL,(OCBAH)
6556' A7 C AND A
6557' ED 42 C SBC HL,BC
6559' 30 05 C JR NC,VALID_REC_LEN
655B' 1E 05 C LD E,05H
655D' C3 6845' C JP ERROR_ROUTINE ; Illegal function call
6560' ED 43 68F5' C VALID_REC_LEN: LD (ANAL_REC_LEN),BC
6564' E1 C POP HL
6565' C9 C RET
C
C ; ----- Analyze Key Position -----
6566' C CHK_KEY_POSI:
6566' CD 685F' C GET_KEY_POSI: CALL GET_TOKEN
6569' E5 C PUSH HL

```

```

656A' 7B      C CHECK_KEY_POSI: LD A,E
656B' FE FF   C                CP OFFH
656D' 20 05   C                JR NZ,KEY_POSI_EXIST
656F' 1E 02   C                LD E,02H
6571' C3 6845' C                JP ERROR_ROUTINE ; Syntax error
6574' 21 688D' C KEY_POSI_EXIST: LD HL,TOKEN
6577' CD 68BF' C                CALL CONV_NUM
657A' 7B      C                LD A,E
657B' FE FF   C                CP OFFH
657D' 20 05   C                JR NZ,NUMERIC_KEY_POSI
657F' 1E 0D   C                LD E,0DH
6581' C3 6845' C                JP ERROR_ROUTINE ; Type mismatch
6584' 2A 68F5' C NUMERIC_KEY_POSI: LD HL,(ANAL_REC_LEN)
6587' A7      C                AND A
6588' ED 42   C                SBC HL,BC
658A' 30 05   C                JR NC,VALID_KEY_POSI
658C' 1E 05   C                LD E,05H
658E' C3 6845' C                JP ERROR_ROUTINE ; Illegal function call
6591' ED 43 68F7' C VALID_KEY_POSI: LD (ANAL_KEY_POSI),BC
6595' E1      C                POP HL
6596' C9      C                RET

```

C ; ----- Analyze Key Length -----

```

6597' C CHK_KEY_LEN:
6597' CD 685F' C GET_KEY_LEN: CALL GET_TOKEN
659A' E5      C                PUSH HL
659B' 7B      C CHECK_KEY_LEN: LD A,E
659C' FE FF   C                CP OFFH
659E' 20 05   C                JR NZ,KEY_LEN_EXIST
65A0' 1E 02   C                LD E,02H
65A2' C3 6845' C                JP ERROR_ROUTINE ; Syntax error
65A5' 21 688D' C KEY_LEN_EXIST: LD HL,TOKEN
65A8' CD 68BF' C                CALL CONV_NUM
65AB' 7B      C                LD A,E
65AC' FE FF   C                CP OFFH
65AE' 20 05   C                JR NZ,NUMERIC_KEY_LEN
65B0' 1E 0D   C                LD E,0DH
65B2' C3 6845' C                JP ERROR_ROUTINE ; Type mismatch
65B5' 2A 68F7' C NUMERIC_KEY_LEN: LD HL,(ANAL_KEY_POSI)
65B8' 09      C                ADD HL,BC
65B9' 2B      C                DEC HL
65BA' ED 58 68F5' C                LD DE,(ANAL_REC_LEN)
65BE' EB      C                EX DE,HL
65BF' A7      C                AND A
65C0' ED 42   C                SBC HL,BC
65C2' 30 05   C                JR NC,VALID_KEY_LEN
65C4' 1E 05   C                LD E,05H
65C6' C3 6845' C                JP ERROR_ROUTINE ; illegal function call
65C9' 0B      C VALID_KEY_LEN: DEC BC
65CA' ED 43 68F9' C                LD (ANAL_KEY_LEN),BC
65CE' E1      C                POP HL
65CF' C9      C                RET

```

```

C
C ;
C ;      This subroutine opens random file of current file no.
C ;      Input Paramete :
C ;      WORK_FILE_SPEC      File specification
C ;      WORK_FILE_NO        Current file no.
C ;      WORK_REC_LEN        Record length
C ;
C
65D0' 21 68FB' C OPEN_RANDOM_FILE: LD HL,WORK_FILE_SPEC ; Set up File Spec at
65D3' 11 08CD C                      LD DE,08CDH ; location 08CDH
65D6' 01 000C C                      LD BC,000CH ;
65D9' ED 80 C                      LDIR ;
C
65DB' ED 4B 6908' C                      LD BC,(WORK_FILE_NO) ; Calculate file buffer address
65DF' CD 6831' C                      CALL CAL_FILE_ADDR ; of current file no.
C
65E2' 21 00A9 C                      LD HL,00A9H ; Save record length at
65E5' 09 C                      ADD HL,8C ; one location that is offset
65E6' ED 5B 690A' C                      LD DE,(WORK_REC_LEN) ; from begining current file
65EA' 73 C                      LD (HL),E ; buffer = A9H
65EB' 23 C                      INC HL ;
65EC' 72 C                      LD (HL),D ;
C
65ED' AF C                      XOR A ; Initialize location follow
65EE' 1E 07 C                      LD E,07H ; the location that save record
65F0' 23 C CLR_FILE_BUF: INC HL ; length to 00H, total is
65F1' 77 C                      LD (HL),A ; 7 locations
65F2' 1B C                      DEC E ;
65F3' 20 FB C                      JR NZ,CLR_FILE_BUF ;
C
65F5' 11 6606' C                      LD DE,JMP_59CA + 3 ; Save return address from
65F8' D5 C                      PUSH DE ; jumping to 59CAH
65F9' 2A 68E4' C                      LD HL,(NEXT_COMM_ADDR) ;
65FC' 16 03 C                      LD D,03H ; Set file mode to random file
65FE' 3A 6908' C                      LD A,(WORK_FILE_NO) ; Set file no.
6601' 5F C                      LD E,A ;
6602' C5 C                      PUSH BC ; Send file buffer address.
6603' C3 59CA C JMP_59CA: JP 59CAH ; Jump to open random file.
6606' C9 C                      RET
C
C ;
C ;      This subroutine close random file of current file no.
C ;      Input Parameter :
C ;      WORK_FILE_NO        Current file no.
C ;
C
6607' ED 4B 6908' C CLOSE_RANDOM_FILE: LD BC,(WORK_FILE_NO) ; Calculate file buffer address
660B' CD 6831' C                      CALL CAL_FILE_ADDR ; of current file no.
660E' CD 5742 C                      CALL 5742H ; Call original close routine
6611' C9 C                      RET
C
C
C PAGE

```


999

```

C
C ;
C ;
C ;
C ;
C ;
C ;
C ;
C ;
C ;
C ;
C ;
6612' 3E AF C WRITE_RANDOM_REC: LD A,0AFH ; Set "PUT" flag
6614' 32 SE3A C LD (SE3AH),A ;
6617' ED 4B 6908' C LD BC,(WORK_FILE_NO)
6618' CD 6831' C CALL CAL_FILE_ADDR
661E' 21 00AE C LD HL,00AEH
6621' 09 C ADD HL,8C
6622' 11 6633' C LD DE,JMP_5C08_PUT + 3
6625' D5 C PUSH DE
6626' C5 C PUSH BC
6627' ED 5B 68E4' C LD DE,(NEXT_COMM_ADDR)
662B' D5 C PUSH DE
662C' ED 5B 6910' C LD DE,(WORK_REC_NO) ;
6630' C3 5C08 C JMP_5C08_PUT: JP 5C08H
6633' C9 C RET

```

```

C ;
C ;
C ;
C ;
C ;
C ;
C ;
C ;
C ;
C ;
C ;
6634' 3E 00 C READ_RANDOM_REC: LD A,00H ; Set "GET" flag
6636' 32 SE3A C LD (SE3AH),A ;
6639' ED 4B 6908' C LD BC,(WORK_FILE_NO)
663D' CD 6831' C CALL CAL_FILE_ADDR
6640' 21 00AE C LD HL,00AEH
6643' 09 C ADD HL,8C
6644' 11 6655' C LD DE,JMP_5C08_GET + 3
6647' D5 C PUSH DE
6648' C5 C PUSH BC
6649' ED 5B 68E4' C LD DE,(NEXT_COMM_ADDR)
664D' D5 C PUSH DE
664E' ED 5B 6910' C LD DE,(WORK_REC_NO) ;
6652' C3 5C08 C JMP_5C08_GET: JP 5C08H
6655' C9 C RET

```

```

C
C ;
C ;      This subroutine search index file for specified key.
C ;      Input Parameter :
C ;          SAVE_FILE_INFO
C ;          Specified key which is in file buffer of data file
C ;      Output Parameter :
C ;          (E)          --) 00H Specified key is found
C ;                      --) 01H Key not found at left
C ;                      --) 02H Key not found at right
C ;
C ;
6656' ED 4B 6A4D' C SEARCH_KEY: LD BC,(SAVE_ROOT_REC)
665A' ED 43 6A57' C LD (WORK_INDX_REC),BC
665E' ED 4B 6A4F' C LD BC,(SAVE_ROOT_BYTE)
6662' ED 43 6A59' C LD (WORK_INDX_BYTE),BC
6666' CD 6796' C READ_NEXT_INDX: CALL READ_INDX_NODE
6669' 21 6A5F' C LD HL,WORK_INDX_NODE
666C' ED 5B 6A43' C LD DE,(SAVE_KEY_ADDR)
6670' 3A 6A45' C LD A,(SAVE_KEY_LEN)
6673' 47 C LD B,A
6674' 1A C COMP_KEY_BYTE: LD A,(DE)
6675' BE C CP (HL)
6676' 38 09 C JR C,CHK_LEFT_SUBTREE
6678' 20 1C C JR NZ,CHK_RIGHT_SUBTREE
667A' 23 C INC HL
667B' 13 C INC DE
667C' 10 F6 C DJNZ COMP_KEY_BYTE
667E' 1E 00 C LD E,00H
6680' C9 C RET
6681' 21 6A5F' C CHK_LEFT_SUBTREE: LD HL,WORK_INDX_NODE
6684' ED 4B 6A45' C LD BC,(SAVE_KEY_LEN)
6688' 09 C ADD HL,BC
6689' 01 0004 C LD BC,0004H
668C' 09 C ADD HL,BC
668D' 3E 01 C LD A,01H
668F' 32 6B6B' C LD (PARENT_LINK_FLAG),A
6692' 1E 01 C LD E,01H
6694' 18 13 C JR READ_SUBTREE_KEY
6696' 21 6A5F' C CHK_RIGHT_SUBTREE: LD HL,WORK_INDX_NODE
6699' ED 4B 6A45' C LD BC,(SAVE_KEY_LEN)
669D' 09 C ADD HL,BC
669E' 01 0007 C LD BC,0007H
66A1' 09 C ADD HL,BC
66A2' 3E 02 C LD A,02H
66A4' 32 6B6B' C LD (PARENT_LINK_FLAG),A
66A7' 1E 02 C LD E,02H
66A9' ED 4B 6A57' C READ_SUBTREE_KEY: LD BC,(WORK_INDX_REC)
66AD' ED 43 6B67' C LD (PARENT_INDX_REC),BC
66B1' ED 4B 6A59' C LD BC,(WORK_INDX_BYTE)
66B5' ED 43 6B69' C LD (PARENT_INDX_BYTE),BC
66B9' 7E C LD A,(HL)
66BA' FE 00 C CP 00H
66BC' 20 01 C JR NZ,NON_LEAF_NODE
66BE' C9 C RET

```

66BF'	28	C NON_LEAF_NODE:	DEC HL	
66C0'	28	C	DEC HL	
66C1'	11 6A57'	C	LD DE,WORK_INDY_REC	
66C4'	01 0002	C	LD BC,0002H	
66C7'	ED 80	C	LDIR	
66C9'	7E	C	LD A,(HL)	
66CA'	4F	C	LD C,A	
66CB'	06 00	C	LD B,00H	
66CD'	ED 43 6A59'	C	LD (WORK_INDY_BYTE),BC	
66D1'	C3 6666'	C	JP READ_NEXT_INDY	
		C		
		C		
		C		
			PAGE	

990



ศูนย์วิทยพัชร์พยากร
จุฬาลงกรณมหาวิทยาลัย

```

C
C ;
C ;      This subroutine adjust left link or right link of adjust index
C ;      node of index file.
C ;      Input Parameter :
C ;      (ADJUST_INDX_REC) First rec no. of adjust index
C ;      (ADJUST_INDX_BYTE) First byte no. of adjust index
C ;      (ADJUST_LINK_FLAG) Adjust left or right link flag
C ;      (ADJUST_LINK_REC) New rec no. of adjust link
C ;      (ADJUST_LINK_BYTE) New byte no. of adjust link
C ;      Output Parameter :
C ;
C ;
66D4' ED 4B 6884' C ADJUST_LR_LINK: LD BC,(ADJUST_INDX_REC)
66D8' ED 43 6A57' C LD (WORK_INDX_REC),BC
66DC' ED 4B 6886' C LD BC,(ADJUST_INDX_BYTE)
66E0' ED 43 6A59' C LD (WORK_INDX_BYTE),BC
66E4' CD 6796' C CALL READ_INDX_NODE
66E7' 21 6A5F' C CHK_LR_LINK: LD HL,WORK_INDX_NODE
66EA' ED 4B 6A45' C LD BC,(SAVE_KEY_LEN)
66EE' 09 C ADD HL,BC
66EF' 3A 6888' C LD A,(ADJUST_LINK_FLAG)
66F2' FE 02 C CP 02H
66F4' 28 06 C JR Z,RLINK_CHILD
66F6' 01 0002 C LLINK_CHILD: LD BC,0002H
66F9' 09 C ADD HL,BC
66FA' 18 04 C JR SETUP_LINK
66FC' 01 0005 C RLINK_CHILD: LD BC,0005H
66FF' 09 C ADD HL,BC
6700' 11 6889' C SETUP_LINK: LD DE,ADJUST_LINK_REC
6703' EB C EX DE,HL
6704' 01 0003 C LD BC,0003H
6707' ED 80 C LDIR ; Set up link complete.
C
6709' 01 0001 C LD BC,0001H
670C' ED 43 6A58' C LD (WORK_MOVE_POST),BC
6710' 3A 6A45' C LD A,(SAVE_KEY_LEN)
6713' C6 08 C ADD A,08H
6715' 32 6A5D' C LD (WORK_MOVE_LEN),A
6718' 3E 00 C LD A,00H
671A' 32 6A5E' C LD (WORK_INDX_FLAG),A
671D' CD 6721' C CALL WRITE_INDX_NODE
6720' C9 C RET
C
C PAGE

```

```

C
C ;
C ;      This subroutine write index to index file, the index may be new
C ;      new index or existing index.
C ;      Input Parameter :
C ;      (WORK_INDXX_REC)   First rec no of index
C ;      (WORK_INDXX_BYTE) Beginning byte no in first rec of index
C ;      (WORK_MOVE_POSI)  Starting move position in working index
C ;      (WORK_MOVE_LEN)   Length og moving
C ;      (WORK_INDXX_NODE) Key + Data Record No + Link
C ;      (WORK_INDXX_FLAG) Indicate new or old index
C ;      Output Parameter :
C ;      (WORK_INDXX_REC)   Last rec no of index
C ;      (WORK_INDXX_BYTE) Ending byte no of index plus one
C ;
C ;
6721' ED 4B 68F3' C WRITE_INDXX_MODE: LD BC,(ANAL_FILE_NO) ; Read first record of index
6725' 03          C                   INC BC           ; from index file to prepare
6726' ED 43 6908' C                   LD (WORK_FILE_NO),BC ; for writing index.
672A' ED 4B 6A57' C                   LD BC,(WORK_INDXX_REC) ;
672E' ED 43 6910' C                   LD (WORK_REC_NO),BC ;
6732' CD 6634'   C                   CALL READ_RANDOM_REC ;
C
6735' 2A 6A53'   C                   LD HL,(SAVE_INDXX_ADDR) ; Set up pointer & index length
6738' ED 4B 6A59' C                   LD BC,(WORK_INDXX_BYTE) ; for moving index as :-
673C' 09          C                   ADD HL,BC           ;
673D' 2B          C                   DEC HL           ; HL point to working index
673E' E5          C                   PUSH HL          ; DE point to index file buffer
673F' C5          C                   PUSH BC           ; addr of index
6740' 21 6A5E'   C                   LD HL,WORK_INDXX_NODE - 1
6743' ED 4B 6A5B' C                   LD BC,(WORK_MOVE_POSI) ; C byte no in first rec of
6747' 09          C                   ADD HL,BC           ; index
6748' C1          C                   POP BC           ; B length of moving
6749' D1          C                   POP DE           ;
674A' 3A 6A5D'   C                   LD A,(WORK_MOVE_LEN) ;
674D' 47          C                   LD B,A           ;
C
674E' 7E          C LOAD_INDXX_BYTE: LD A,(HL)        ; Move index byte from working
674F' 12          C                   LD (DE),A         ; index to index file buffer.
6750' 79          C                   LD A,C           ;
6751' FE 80       C                   CP 80H          ; If move over index record
6753' 28 09       C                   JR Z,WRITE_PART_INDXX ; boundary, write partial index
6755' 23          C                   INC HL          ; to index file before go on
6756' 13          C                   INC DE          ; moving index.
6757' 0C          C                   INC C           ;
6758' 10 F4       C CHK_LAST_LOAD: DJNZ LOAD_INDXX_BYTE ; If B = 0, no more index.
675A' 79          C                   LD A,C           ;
675B' 32 6A59'   C                   LD (WORK_INDXX_BYTE),A ; Save last byte no of indx plus 1.
675E' E5          C WRITE_PART_INDXX: PUSH HL          ;
675F' C5          C                   PUSH BC          ;
6760' ED 4B 6A57' C                   LD BC,(WORK_INDXX_REC) ; Write partial index to index
6764' ED 43 6910' C                   LD (WORK_REC_NO),BC ; file.
6768' CD 6612'   C                   CALL WRITE_RANDOM_REC ;
676B' C1          C                   POP BC           ;
676C' 78          C                   LD A,B           ; If B = 0, no more index to
    
```

```

676D' FE 00      C          CP  00H          ; write to index file.    99b
676F' 28 23      C          JR  2,WRITE_NODE_EXIT ;
6771' C5          C          PUSH BC          ; If B (<) 0, there are any more
6772' ED 48 6A57' C          LD  BC,(WORK_INDX_REC) ; index to write, so increment
6776' 03          C          INC  BC          ; index file record no for next
6777' ED 43 6A57' C          LD  (WORK_INDX_REC),BC ; writing.
677B' 3A 6A5E'   C          LD  A,(WORK_INDX_FLAG)
677E' FE FF      C          CP  0FFH
6780' 28 07      C          JR  2,NEW_INDX_NODE
6782' ED 43 6910' C          LD  (WORK_REC_NO),BC
6786' CD 6634'   C          CALL READ_RANDOM_REC
6789' C1          C NEW_INDX_NODE: POP  BC
678A' E1          C          POP  HL
678B' 23          C          INC  HL
678C' ED 58 6A53' C          LD  DE,(SAVE_INDX_ADDR)
6790' 0E 01      C          LD  C,01H
6792' 18 C4      C          JR  CHK_LAST_LOAD
6794' E1          C WRITE_NODE_EXIT: POP  HL
6795' C9          C          RET
          C
          C          PAGE

```

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

```

C
C ;
C ;      This subroutine read index from index file and store in
C ;      working index.
C ;      Input Parameter :
C ;      (WORK_INDX_REC)   First rec no of index
C ;      (WORK_INDX_BYTE) Beginning byte no in first rec of index
C ;      Output Parameter :
C ;      (WORK_INDX_REC)   The same as input
C ;      (WORK_INDX_BYTE) The same as input
C ;      (WORK_INDX_MODE)  Key + Data Record No + Link
C ;
C
6796' ED 4B 68F3' C READ_INDX_MODE: LD BC,(ANAL_FILE_NO) ; Read first record of index
679A' 03          C                INC BC           ; from index file to prepare
679B' ED 43 6908' C                LD (WORK_FILE_NO),BC ; for writing index.
679F' ED 4B 6A57' C                LD BC,(WORK_INDX_REC) ;
67A3' ED 43 6910' C                LD (WORK_REC_NO),BC ;
67A7' CD 6634'   C                CALL READ_RANDOM_REC ;
C
67AA' 2A 6A53'   C                LD HL,(SAVE_INDX_ADDR) ; Set up pointer & index length
67AD' ED 4B 6A59' C                LD BC,(WORK_INDX_BYTE) ; for moving index as :-
67B1' 09         C                ADD HL,BC           ;
67B2' 28         C                DEC HL             ; DE point to working index
67B3' 11 6A5F'   C                LD DE,WORK_INDX_MODE ; HL point to index file buffer
67B6' 3A 6A45'   C                LD A,(SAVE_KEY_LEN) ; addr of index
67B9' C6 08      C                ADD A,08H          ; C byte no in first rec of indx
67BB' 47         C                LD B,A             ; B length of index
C
67BC' 7E         C STORE_INDX_BYTE: LD A,(HL)         ; Move index byte from index
67BD' 12         C                LD (DE),A         ; file buffer to working index.
67BE' 79         C                LD A,C           ;
67BF' FE 80      C                CP 80H           ; If move over index record
67C1' 28 06      C                JR Z,READ_PART_INDX ; boundary, read partial index
67C3' 23         C                INC HL           ; from index file before go on
67C4' 13         C                INC DE           ; moving index.
67C5' 0C         C                INC C           ;
67C6' 10 F4      C CHK_LAST_STORE: DJNZ STORE_INDX_BYTE ; If B = 0, no more index.
67C8' C9         C                RET
C
67C9' D5         C READ_PART_INDX: PUSH DE           ;
67CA' C5         C                PUSH BC          ;
67CB' ED 4B 6910' C                LD BC,(WORK_REC_NO) ;
67CF' 03         C                INC BC           ;
67D0' ED 43 6910' C                LD (WORK_REC_NO),BC ;
67D4' CD 6634'   C                CALL READ_RANDOM_REC ;
67D7' C1         C                POP BC           ;
67D8' D1         C                POP DE           ;
67D9' 2A 6A53'   C                LD HL,(SAVE_INDX_ADDR)
67DC' 13         C                INC DE           ;
67DD' 0E 01      C                LD C,01H         ;
67DF' 18 E5      C                JR CHK_LAST_STORE
C
C                PAGE
    
```

```

C
C ;
C ;      This subroutine create link from left and right link in working
C ;      index and save in WORK_INDXX_LINK
C ;      Input Parameter :
C ;      (WORK_INDXX_NODE)      Working index node
C ;      Output Parameter :
C ;      (WORK_INDXX_LINK)     Left & right link of working index
C ;
C
67E1' 21 6A5F' C CREATE_INDXX_LINK: LD HL,WORK_INDXX_NODE
67E4' ED 4B 6A45' C      LD BC,(SAVE_KEY_LEN)
67E8' 09          C      ADD HL,BC
67E9' 01 0002    C      LD BC,0002H
67EC' 09          C      ADD HL,BC
67ED' 11 686C'   C      LD DE,WORK_LLXX_LINK_REC
67F0' 01 0003    C      LD BC,0003H
67F3' ED 80      C      LDIR
67F5' E5          C      PUSH HL
67F6' 21 686F'   C      LD HL,WORK_LLXX_LINK_BYTE + 1
67F9' 36 00      C      LD (HL),00H
67FB' E1          C      POP HL
67FC' 11 6870'   C      LD DE,WORK_RLXX_LINK_REC
67FF' 01 0003    C      LD BC,0003H
6802' ED 80      C      LDIR
6804' 21 6873'   C      LD HL,WORK_RLXX_LINK_BYTE + 1
6807' 36 00      C      LD (HL),00H
6809' C9          C      RET
C
C      PAGE
C

```

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย


```

C
C ;
C ;
C ;      This subroutine load file information of current file no from
C ;      table into save area.
C ;      Input Parameter :
C ;      (ANAL_FILE_NO)          Current file no
C ;
C ;
C
680A' ED 4B 68F3' C LOAD_FILE_INFO: LD BC,(ANAL_FILE_NO) ; Calculate address of table
680E' CD 683A'   C                CALL CAL_TABL_ADDR ; element no n, n = current
6811' 22 6A55'   C                LD (SAVE_TABL_ADDR),HL ; file no.
6814' 11 6A41'   C                LD DE,SAVE_FILE_INFO ;
6817' 01 0014   C                LD BC,0014H ;
681A' ED 80     C                LDIR ;
681C' C9       C                RET ;
C
C ;
C ;      This subroutine store file information of current file no from
C ;      save area into table.
C ;      Input Parameter :
C ;      (ANAL_FILE_NO)          Current file no
C ;
C ;
C
681D' ED 4B 68F3' C STORE_FILE_INFO: LD BC,(ANAL_FILE_NO) ; Calculate address of table
6821' CD 683A'   C                CALL CAL_TABL_ADDR ; element no n, n = current
6824' 22 6A55'   C                LD (SAVE_TABL_ADDR),HL ; file no.
6827' 11 6A41'   C                LD DE,SAVE_FILE_INFO ;
682A' EB         C                EX DE,HL ;
682B' 01 0014   C                LD BC,0014H ;
682E' ED 80     C                LDIR ;
6830' C9       C                RET ;
C
C                PAGE
    
```

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย



```

C
C ;
C ;      This subroutine calculate beginning location of file buffer
C ;      address of current file no.
C ;      Input Parameter :
C ;      (BC)                Current file no
C ;
C
6831' 21 0873 C CAL_FILE_ADDR: LD HL,0873H      ; Determine file buffer addr
6834' 09      C      ADD HL,BC          ; that will be kept in (BC)
6835' 09      C      ADD HL,BC          ;
6836' 4E      C      LD C,(HL)          ;
6837' 23      C      INC HL            ;
6838' 46      C      LD B,(HL)         ;
6839' 09      C      RET
C
C      PAGE
C

```

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

```

C
C ;
C ;      This subroutine calculate beginning location of table element
C ;      of current file no.
C ;      Input Parameter :
C ;      (BC)                Current file no
C ;
C
683A' 21 6901' C CAL_TABL_ADDR: LD HL,TABL_FILE_INFO - 20 ; Load HL with table address.
683D' 11 0014 C          LD DE,0014H          ; Load DE with element length.
6840' 41 C          LD B,C              ; Now file no is in B.
6841' 19 C SKIP_NEXT_ELMT: ADD HL,DE      ; Skip to next element
6842' 10 FD C          DJNZ SKIP_NEXT_ELMT ; until reach desired element.
6844' C9 C          RET
C
C          PAGE
C

```

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

```

C
C ;
C ;      This subroutine determine error code & output error message,
C ;      then direct BASIC Interpreter to abort running program.
C ;      Input Parameter :
C ;      (E)                Error code of BASIC
C ;
C
6845' E1 C ERROR_ROUTINE: POP HL           ; Pop prog text address
6846' E1 C                POP HL           ; Pop return address
6847' E1 C                POP HL           ;
6848' D5 C                PUSH DE
C ; ----- Clear file buffer -----
6849' 3A 0893 C CLEAR_FILE_BUF: LD A,(0893H)
684C' 47 C                LD B,A
684D' 21 0873 C                LD HL,0873H
6850' AF C                XOR A
6851' 5E C CLEAR_OPEN_FLAG: LD E,(HL)      ; clear open file flag
6852' 23 C                INC HL
6853' 56 C                LD D,(HL)
6854' 23 C                INC HL
6855' 12 C                LD (DE),A
6856' 10 F9 C                DJNZ CLEAR_OPEN_FLAG
6858' CD 554F C                CALL 554FH          ; clear file buffer
C ; -----
6858' D1 C                POP DE
685C' C3 0D94 C                JP 0D94H
C
C
C                PAGE
    
```

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

```

C
C ;
C ; This subroutine scans text and get next token that is delimited
C ; by End program line marker (00H) or Comma (2CH) or Colon (3AH).
C ; Input Parameter : (HL) First character of text to be scanned
C ; Output Parameter : (TOKEN) Whose length is in (BC)
C ; : (TOKEN_LEN) that kept token length (2 bytes)
C ; : (HL) Last character of token in text
C ; : (E) 00H --> Token exist
C ; : FFH --> Null token
C ;
C
685F' E5 C GET_TOKEN: PUSH HL ; Save first text char position.
6860' 21 68A3' C LD HL,BLANK_CHAR ; -----
6863' 01 0014 C LD BC,0014H ; Initialize TOKEN to twenty
6866' 11 688D' C LD DE,TOKEN ; blank.
6869' ED B0 C LDIR ; -----
686B' 11 688D' C LD DE,TOKEN ; Load DE with address of token.
686E' E1 C POP HL ; Restore first text char posi.
686F' 01 0000 C LD BC,0000H ; Reset token length to zero (C)
C ; and found quote no. (B)
C
6872' 7E C SCAN_BLANK: LD A,(HL) ;
6873' FE 20 C CP 20H ; This loop scans text until
6875' 20 03 C JR NZ,MOV_TOKEN ; find non-blank character.
6877' 23 C INC HL ;
6878' 18 F8 C JR SCAN_BLANK ;
C
687A' FE 00 C MOV_TOKEN: CP 00H ; -----
687C' 28 28 C JR Z,LAST_TOKEN ; Compare until find
687E' FE 2C C CP 2CH ; End program line marker, 00H
6880' 28 27 C JR Z,LAST_TOKEN ; Comma, 2CH
6882' FE 3A C CP 3AH ; Colon, 3AH
6884' 28 0B C JR Z,FOUND_COLON ; Token length = 20, (C) = 14H
6886' FE 20 C CP 20H ; If find blank, skip blank to
6888' 28 1B C JR Z,SKIP_BLANK ; next character
688A' FE 22 C CP 22H ;
688C' 20 0F C JR NZ,VALID_CHAR ;
688E' 04 C INC B ; And if find quote, count no. of
688F' 18 0C C JR VALID_CHAR ; quote
6891' F5 C FOUND_COLON: PUSH AF ; Save value of token
6892' 78 C LD A,B ; If find colon, check no. of quote
6893' FE 01 C CP 1 ; if equal 1 token ok
6895' 20 03 C JR NZ,COLON_MARKER ; otherwise end program line marker
6897' F1 C POP AF ; Restore value of token
6898' 18 03 C JR VALID_CHAR ;
689A' F1 C COLON_MARKER: POP AF ; Restore value of token
689B' 18 0C C JR LAST_TOKEN ; -----
C
689D' 12 C VALID_CHAR: LD (DE),A ; Move valid char. to TOKEN.
689E' 13 C INC DE ; Skip to next token char.
689F' 03 C INC BC ; Add 1 to token length.
68A0' 79 C LD A,C ;
68A1' FE 14 C CP 14H ; Token full ?
68A3' 28 04 C JR Z,LAST_TOKEN ;

```

```

68A5' 23      C SKIP_BLANK:      INC HL          ; Load A with next text char. 9BE
68A6' 7E      C                      LD A,(HL)       ;
68A7' 18 D1   C                      JR MOV_TOKEN   ; -----
C
68A9' FE 00   C LAST_TOKEN:      CP 00H         ; If latest compared char. is
68AB' 28 01   C                      JR Z,SAVE_TOKEN_LEN ; "00H", allow (HL) is the same,
68AD' 23      C                      INC HL        ; otherwise add 1 to (HL)
68AE' ED 43 68A1' C SAVE_TOKEN_LEN: LD (TOKEN_LEN),BC ; Save token length
68B2' 1E 00   C                      LD E,00H      ; Assume token exist
68B4' 79      C                      LD A,C        ;
68B5' FE 00   C                      CP 00H         ; If (BC) = 0000H
68B7' 20 05   C                      JR NZ,TOKEN_EXIT ; set (E) = FFH --> Null token
68B9' 78      C                      LD A,B        ;
68BA' 20 02   C                      JR NZ,TOKEN_EXIT ;
68BC' 1E FF   C NULL_TOKEN:      LD E,0FFH     ;
68BE' C9      C TOKEN_EXIT:      RET           ;
C
C PAGE

```



ศูนย์วิทยทรัพยากร
 จุฬาลงกรณ์มหาวิทยาลัย

```

C
C ;
C ;      This subroutine converts numeric value that kept in program text
C ;      area into 2 bytes binary. (0000H - 7FFFH)
C ;      Input Parameter : (HL)   First byte of numeric value
C ;      Output Parameter : (BC)   Converted 2 bytes binary
C ;      : (E)      00H --> Convert complete
C ;      :          FFH --> Convert error
C ;
C
68BF' 01 0000 C CONV_NUM: LD BC,0000H ; Reset BC to zero binary
68C2' 1E 00 C LD E,00H ; Assume completely conversion
68C4' 7E C LD A,(HL) ; Load A with numeric type flag
68C5' FE 1C C CP 1CH ; If flag = "1CH"
68C7' 28 0B C JR Z,NUM_TYPE_1 ; --> 256 - 32767
68C9' FE 0F C CP 0FH ; If flag = "0FH"
68CB' 28 0C C JR Z,NUM_TYPE_2 ; --> 10 - 255
68CD' FE 1B C CP 1BH ; If no flag
68CF' 38 0D C JR C,NUM_TYPE_3 ; --> 0 - 9
68D1' 1E FF C LD E,0FFH ; If process until this step
68D3' C9 C RET ; --> Error numeric
C
68D4' 23 C NUM_TYPE_1: INC HL ;
68D5' 4E C LD C,(HL) ; Move 2 next bytes to (BC)
68D6' 23 C INC HL ; first byte is low order
68D7' 46 C LD B,(HL) ; second byte is high order
68D8' C9 C RET ;
C
68D9' 23 C NUM_TYPE_2: INC HL ;
68DA' 4E C LD C,(HL) ; Move only next byte to (C)
68DB' 06 00 C LD B,00H ; & set (B) to zero
68DD' C9 C RET ;
C
68DE' D6 11 C NUM_TYPE_3: SUB 11H ; Subtract 11H from A will
68E0' 4F C LD C,A ; receive the binary value
68E1' 06 00 C LD B,00H ; then store in (C) & set (B)
68E3' C9 C RET ; to zero
C
C PAGE

```

ศูนย์วิทยุทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

```

C
C ; ----- Data Area -----
C
68E4' C NEXT_COMM_ADDR: DEFS 2
C
68E6' C ANAL_FILE_INFO:
68E6' 01 C ANAL_CUR_DRIVE: DEFS 1
68E7' C ANAL_FILE_NAME: DEFS 8
68EF' C ANAL_FILE_TYPE: DEFS 3
68F2' C ANAL_FILE_MODE: DEFS 1
68F3' C ANAL_FILE_NO: DEFS 2
68F5' C ANAL_REC_LEN: DEFS 2
68F7' C ANAL_KEY_POS1: DEFS 2
68F9' C ANAL_KEY_LEN: DEFS 2
C
68FB' C WORK_FILE_INFO:
68FB' C WORK_FILE_SPEC:
68FB' C WORK_CUR_DRIVE: DEFS 1
68FC' C WORK_FILE_NAME: DEFS 8
6904' C WORK_FILE_TYPE: DEFS 3
6907' C WORK_FILE_MODE: DEFS 1
6908' C WORK_FILE_NO: DEFS 2
690A' C WORK_REC_LEN: DEFS 2
690C' C WORK_KEY_POS1: DEFS 2
690E' C WORK_KEY_LEN: DEFS 2
6910' C WORK_REC_NO: DEFS 2
C
6912' 4E 44 58 C STRING_NDX: DEFH "NDX"
C
6915' C TABL_FILE_INFO: DEFS 300 ; This table consists of 15 elements, each
; element consists of follwing fields
C ; - TABL_REC_LEN 2 bytes
C ; - TABL_KEY_ADDR 2 bytes
C ; - TABL_KEY_LEN 2 bytes
C ; - TABL_DATA_REC 2 bytes
C ; - TABL_INDX_REC 2 bytes
C ; - TABL_INDX_BYTE 2 bytes
C ; - TABL_ROOT_REC 2 bytes
C ; - TABL_ROOT_BYTE 2 bytes
C ; - TABL_NO_REC 2 bytes
C ; - TABL_INDX_ADDR 2 bytes
C
6A41' C SAVE_FILE_INFO:
6A41' C SAVE_REC_LEN: DEFS 2
6A43' C SAVE_KEY_ADDR: DEFS 2
6A45' C SAVE_KEY_LEN: DEFS 2
6A47' C SAVE_DATA_REC: DEFS 2
6A49' C SAVE_INDX_REC: DEFS 2
6A4B' C SAVE_INDX_BYTE: DEFS 2
6A4D' C SAVE_ROOT_REC: DEFS 2
6A4F' C SAVE_ROOT_BYTE: DEFS 2
6A51' C SAVE_NO_REC: DEFS 2
6A53' C SAVE_INDX_ADDR: DEFS 2
C
6A55' C SAVE_TABL_ADDR: DEFS 2

```


Macros:

950

Symbols:

63B9'	ADJUST_HEADER	6049'	ADJUST_INDEX	6886'	ADJUST_INDY_BYTE
6884'	ADJUST_INDY_REC	6888'	ADJUST_LINK_BYTE	6888'	ADJUST_LINK_FLAG
6889'	ADJUST_LINK_REC	66D4'	ADJUST_LR_LINK	68E6'	ANAL_CUR_DRIVE
68E6'	ANAL_FILE_INFO	68F2'	ANAL_FILE_MODE	68E7'	ANAL_FILE_NAME
68F3'	ANAL_FILE_NO	68EF'	ANAL_FILE_TYPE	68F9'	ANAL_KEY_LEN
68F7'	ANAL_KEY_POSI	68F5'	ANAL_REC_LEN	68A3'	BLANK_CHAR
6831'	CAL_FILE_ADDR	683A'	CAL_TABL_ADDR	645F'	CHECK_FILE_MODE
6491'	CHECK_FILE_NO	64C6'	CHECK_FILE_SPEC	659B'	CHECK_KEY_LEN
656A'	CHECK_KEY_POSI	6539'	CHECK_REC_LEN	645B'	CHK_FILE_MODE
648D'	CHK_FILE_NO	64C2'	CHK_FILE_SPEC	61F9'	CHK_INDY_LLINK
622C'	CHK_INDY_RLINK	6597'	CHK_KEY_LEN	6566'	CHK_KEY_POSI
6758'	CHK_LAST_LOAD	67C6'	CHK_LAST_STORE	6681'	CHK_LEFT_SUBTREE
66E7'	CHK_LR_LINK	6535'	CHK_REC_LEN	6696'	CHK_RIGHT_SUBTRE
6849'	CLEAR_FILE_BUF	6851'	CLEAR_OPEN_FLAG	63EC'	CLOSEX
6422'	CLOSEX_END_ANAL	6424'	CLOSEX_EXEC	640E'	CLOSEX_FTYPE_ANA
63FD'	CLOSEX_INIT	63F7'	CLOSEX_RET	6607'	CLOSE_RANDOM_FIL
65F0'	CLR_FILE_BUF	689A'	COLON_MARKER	6454'	COMM_EXIT
6674'	COMP_KEY_BYTE	688F'	CONV_NUM	5E66'	CREATE
5E9E'	CREATE_EXEC	67E1'	CREATE_INDY_LINK	5E72'	CREATE_INIT
5E88'	CREATE_SYNT_ANAL	6043'	DUP_KEY	6845'	ERROR_ROUTINE
6469'	FILE_MODE_EXIST	649B'	FILE_NO_EXIST	64D0'	FILE_SPEC_EXIST
61E3'	FILL_ALL_ZERO	5F35'	FILL_HIGHEST_KEY	60C5'	FILL_NULL_LINK
61D3'	FILL_ZERO_BYTE	5F3C'	FILL_ZERO_LINK	631A'	FIND_NULL_LLINK
6891'	FOUND_COLON	64F1'	GET_FILE_DRIVE	645B'	GET_FILE_MODE
64FE'	GET_FILE_NAME	648D'	GET_FILE_NO	64C2'	GET_FILE_SPEC
651E'	GET_FILE_TYPE	6597'	GET_KEY_LEN	6566'	GET_KEY_POSI
6535'	GET_REC_LEN	685F'	GET_TOKEN	0000'	INDEX
64F9'	INIT_GET_FILE_NA	6028'	INSERT	6038'	INSERT_EXEC
6031'	INSERT_SYNT_ANAL	6603'	JMP_59CA	6652'	JMP_5C08_GET
6630'	JMP_5C08_PUT	65A5'	KEY_LEN_EXIST	6574'	KEY_POSI_EXIST
687E'	LAST_INDY_BYTE	687C'	LAST_INDY_REC	68A9'	LAST_TOKEN
66F6'	LLINK_CHILD	680A'	LOAD_FILE_INFO	674E'	LOAD_INDY_BYTE
6488'	MODE_X	687A'	MOV_TOKEN	6789'	NEW_INDY_MODE
68E4'	NEXT_COMM_ADDR	625F'	NONNULL_LR_LINK	62E6'	NONNULL_L_LINK
668F'	NON_LEAF_NODE	64A4'	NO_SHARP_CHAR	688C'	NULL_TOKEN
64B1'	NUMERIC_FILE_NO	6585'	NUMERIC_KEY_LEN	6584'	NUMERIC_KEY_POSI
6553'	NUMERIC_REC_LEN	68D4'	NUM_TYPE_1	68D9'	NUM_TYPE_2
68DE'	NUM_TYPE_3	5F7E'	OPENX	5FB3'	OPEN_EXEC
5F8A'	OPEN_INIT	65D0'	OPEN_RANDOM_FILE	5FA3'	OPEN_SYNT_ANAL
6869'	PARENT_INDY_BYTE	6867'	PARENT_INDY_REC	686B'	PARENT_LINK_FLAG
64DD'	QUOTE_EXIST	6476'	QUOTE_HAVE	6796'	READ_INDY_MODE
6666'	READ_NEXT_INDY	67C9'	READ_PART_INDY	6634'	READ_RANDOM_REC
66A9'	READ_SUBTREE_KEY	6543'	REC_LEN_EXIST	619B'	REMOVE
61AB'	REMOVE_EXEC	618C'	REMOVE_KEY_FOUND	61A4'	REMOVE_SYNT_ANAL
66FC'	RLINK_CHILD	6A47'	SAVE_DATA_REC	6A41'	SAVE_FILE_INFO
6A53'	SAVE_INDY_ADDR	6A48'	SAVE_INDY_BYTE	6B74'	SAVE_INDY_LINK
6A49'	SAVE_INDY_REC	6A43'	SAVE_KEY_ADDR	6A45'	SAVE_KEY_LEN
6876'	SAVE_LLINK_BYTE	6874'	SAVE_LLINK_REC	6A51'	SAVE_NO_REC
6A41'	SAVE_REC_LEN	687A'	SAVE_RLINK_BYTE	6B78'	SAVE_RLINK_REC
6A4F'	SAVE_ROOT_BYTE	6A4D'	SAVE_ROOT_REC	6A55'	SAVE_TABL_ADDR
68AE'	SAVE_TOKEN_LEN	6872'	SCAN_BLANK	6119'	SEARCH
6129'	SEARCH_EXEC	6656'	SEARCH_KEY	613A'	SEARCH_KEY_FOUND

6122'	SEARCH_SYNT_ANAL	6700'	SETUP_LINK	68A5'	SKIP_BLANK
6841'	SKIP_NEXT_ELMT	681D'	STORE_FILE_INFO	67BC'	STORE_INDX_BYTE
6912'	STRING_NDX	6882'	SUCC_INDX_BYTE	6880'	SUCC_INDX_REC
6915'	TABL_FILE_INFO	688D'	TOKEN	688E'	TOKEN_EXJT
68A1'	TOKEN_LEN	615A'	UPDATE	616A'	UPDATE_EXEC
6178'	UPDATE_KEY_FOUND	6163'	UPDATE_SYNT_ANAL	689D'	VALID_CHAR
6519'	VALID_FILE_NAME	648C'	VALID_FILE_NO	6533'	VALID_FILE_TYPE
65C9'	VALID_KEY_LEN	6591'	VALID_KEY_POSJ	6560'	VALID_REC_LEN
68FB'	WORK_CUR_DRIVE	68FB'	WORK_FILE_INFO	6907'	WORK_FILE_MODE
68FC'	WORK_FILE_NAME	6908'	WORK_FILE_NO	68FB'	WORK_FILE_SPEC
6904'	WORK_FILE_TYPE	6A59'	WORK_INDX_BYTE	6A5E'	WORK_INDX_FLAG
686C'	WORK_INDX_LINK	6ASF'	WORK_INDX_NODE	6A57'	WORK_INDX_REC
690E'	WORK_KEY_LEN	690C'	WORK_KEY_POSJ	686E'	WORK_LLINK_BYTE
686C'	WORK_LLINK_REC	6A5D'	WORK_MOVE_LEN	6A5B'	WORK_MOVE_POSJ
690A'	WORK_REC_LEN	6910'	WORK_REC_NO	6872'	WORK_RLINK_BYTE
6B70'	WORK_RLINK_REC	6721'	WRITE_INDX_NODE	6794'	WRITE_NODE_EXJT
675E'	WRITE_PART_INDX	6612'	WRITE_RANDOM_REC		

๑๒๘

No Fatal error(s)

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย



ภาคผนวก ฉ .

ตัวอย่างโปรแกรมการใช้คำสั่งการจับเก็บแฟ้มข้อมูลแบบดัชนี

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

```

1000 REM #####
1005 REM #
1010 REM #   PROGRAM ID       : DEMOINDX.BAS
1015 REM #   FUNCTION       : Demo Index File Organization Command
1020 REM #   AUTHOR        : Prasert Funwanit
1025 REM #   WRITTEN DATE  : 1 May 1987
1030 REM #   REVISION DATE : 1 May 1987
1035 REM #
1040 REM #####
1045 GOTO 1060
1050 SAVE "DEMOINDX.BAS": HOME: END
1055 '
1060 GOSUB 2000 ' File initialization
1065 GOSUB 2500 ' Output demo screen
1070 DEMO.PROGRAM = -1
1075 WHILE DEMO.PROGRAM
1080   GOSUB 3000 ' Select operation code
1085   WHILE SAME.OPERATION.CODE
1090     IF OPERATION.CODE$ (>) "C" THEN 1095 ELSE 1175
1095     GOSUB 3500 ' Read & check student ID.
1100     IF LEFT$(SC.STUDENT.ID$,1) = "/" THEN SAME.OPERATION.CODE = 0:
1105       GOTO 1085
1110       GOSUB 4000 ' Output thesis details
1115       ON OPERATION.CODE GOTO 1115,1115,1135,1150
1120       ' ----- Add or Update Thesis Record -----
1125       GOSUB 4500 ' Read thesis details
1130       GOSUB 5000 ' Add or delete thesis record
1135       GOTO 1165
1140       ' ----- Delete Thesis Record -----
1145       GOSUB 5500 ' Delete thesis details
1150       GOTO 1165
1155       ' ----- Inquire Thesis Record -----
1160       GOSUB 6000 ' Inquire thesis details
1165       GOTO 1165
1170       GOSUB 6500 ' Clear thesis details on screen
1175       GOTO 1185
1180       'ELSE
1185       GOSUB 7000 ' Create thesis file structure
1190       'ENDIF
1195 WEND ' SAME.OPERATION.CODE
1200 WEND ' DEMO.PROGRAM
1205 GOSUB 7500 ' Close index file
1205 END

```

```

2000 REM #####
2005 REM #                               File Initialization
2010 REM #####
2015 '
2020 OPEN "X", #1, "THESIS.DAT", 256
2025 FIELD #1, 7 AS STUDENT.ID$, 40 AS STUDENT.NAME$, 60 AS THESIS.NAME$,
      40 AS ADVISOR.NAME$, 40 AS COADVISOR.NAME$,
      40 AS DEPARTMENT.NAME$, 4 AS EDUCATION.YEAR$, 25 AS FILLER$

2030 '
2035 RETURN
2500 REM #####
2505 REM #                               Output Demo Screen
2510 REM #####
2515 '
2520 HOME
2525 VTAB 21: HTAB 1: PRINT CHR$(22)
2530 VTAB 2: HTAB 37: PRINT "DEMO PROGRAM"
2535 VTAB 4: HTAB 26: PRINT "INDEX FILE ORGANIZATION COMMAND"
2540 VTAB 21: HTAB 1: PRINT CHR$(23)
2545 VTAB 6: HTAB 1: PRINT STRING$(80,"#")
2550 VTAB 21: HTAB 1: PRINT CHR$(22)
2555 VTAB 8: HTAB 3: PRINT "STUDENT ID."
2560 VTAB 8: HTAB 16: PRINT "[ ..... ]"
2565 VTAB 10: HTAB 3: PRINT "STUDENT NAME"
2570 VTAB 10: HTAB 16: PRINT "[ "; STRING$(40,"."); " ]"
2575 VTAB 12: HTAB 3: PRINT "THESIS TITLE"
2580 VTAB 12: HTAB 16: PRINT "[ "; STRING$(60,"."); " ]"
2585 VTAB 14: HTAB 3: PRINT "ADVISOR NAME"
2590 VTAB 14: HTAB 16: PRINT "[ "; STRING$(40,"."); " ]"
2595 VTAB 16: HTAB 16: PRINT "[ "; STRING$(40,"."); " ]"
2600 VTAB 18: HTAB 3: PRINT "DEPARTMENT"
2605 VTAB 18: HTAB 16: PRINT "[ "; STRING$(40,"."); " ]"
2610 VTAB 20: HTAB 3: PRINT "YEAR"
2615 VTAB 20: HTAB 16: PRINT "[ .... ]"
2620 '
2625 RETURN
3000 REM #####
3005 REM #                               Select Operation Code
3010 REM #####
3015 '
3020 VTAB 21: HTAB 1: PRINT CHR$(22)
3025 VTAB 8: HTAB 18: PRINT "....."
3030 VTAB 20: HTAB 55: PRINT "OPERATION CODE"
3035 VTAB 20: HTAB 76: PRINT "[ . ]"
3040 VTAB 21: HTAB 1: PRINT CHR$(23)
3045 VTAB 22: HTAB 10: PRINT "I = Insert"
3050 VTAB 22: HTAB 36: PRINT "U = Update"
3055 VTAB 22: HTAB 61: PRINT "R = Remove"
3060 VTAB 23: HTAB 10: PRINT "S = Search"
3065 VTAB 23: HTAB 36: PRINT "C = Create"

```

```

3070 VTAB 23: HTAB 61: PRINT "X = Exit"
3075 OPERATION.CODE$ = ".": SELECT.OPERATION.CODE = -1
3080 WHILE SELECT.OPERATION.CODE
3085   VT = 20: SP = 78: EP = 78: PICT$ = "X": ST$ = OPERATION.CODE$:
      GOSUB 5000: OPERATION.CODE$ = ST$
3090   OPERATION.CODE = INSTR("IURSCX",OPERATION.CODE$)
3095   IF OPERATION.CODE (>) 0 THEN 3100 ELSE 3130
3100     SELECT.OPERATION.CODE = 0
3105     IF OPERATION.CODE$ (>) "X" THEN 3110 ELSE 3115
3110     SAME.OPERATION.CODE = -1: GOTO 3125
3115     'ELSE
3120     SAME.OPERATION.CODE = 0: DEMO.PROGRAM = 0
3125     'ENDIF
3130     'ENDIF
3135 WEND ' SELECT.OPERATION.CODE
3140 VTAB 19: HTAB 55: PRINT SPACE$(26)
3145 VTAB 20: HTAB 55: PRINT SPACE$(26)
3150 VTAB 21: HTAB 55: PRINT SPACE$(26)
3155 VTAB 22: HTAB 10: PRINT SPACE$(65)
3160 VTAB 23: HTAB 10: PRINT SPACE$(65)
3165 '
3170 RETURN
3500 REM #####
3505 REM #           Read & Check Student ID           #
3510 REM #####
3515 '
3520 VTAB 21: HTAB 1: PRINT CHR$(22)
3525 VTAB 22: HTAB 3: PRINT "Press '/' to select new operation code ";
3530 VTAB 21: HTAB 1: PRINT CHR$(23)
3535 SC.STUDENT.ID$ = ".....": READ.STUDENT.ID = -1
3540 WHILE READ.STUDENT.ID
3545   VT = 8: SP = 18: EP = 24: PICT$ = "X999999": ST$ = SC.STUDENT.ID$:
      GOSUB 5000: SC.STUDENT.ID$ = ST$
3550   IF LEFT$(SC.STUDENT.ID$,1) = "/" THEN READ.STUDENT.ID = 0: GOTO 3540
3555   IF OPERATION.CODE$ = "I" THEN 3560 ELSE 3600
3560     LSET STUDENT.ID$ = SC.STUDENT.ID$
3565     SEARCH #1
3570     IF ERR = 0 THEN 3575 ELSE 3580
3575     VTAB 20: HTAB 41: PRINT "Duplicate Student ID ";: GOTO 3590
3580     'ELSE
3585     READ.STUDENT.ID = 0
3590     'ENDIF
3595     GOTO 3640
3600     'ELSE
3605     LSET STUDENT.ID$ = SC.STUDENT.ID$
3610     SEARCH #1
3615     IF ERR (>) 0 THEN 3620 ELSE 3625
3620     VTAB 20: HTAB 41: PRINT "Student ID Not Found ";: GOTO 3635
3625     'ELSE
3630     READ.STUDENT.ID = 0

```



```

3635     'ENDIF
3640     'ENDIF
3645 WEND ' READ.STUDENT.ID
3650 VTAB 20: HTAB 41: PRINT SPACE$(40)
3655 VTAB 21: HTAB 3: PRINT SPACE$(76)
3660 VTAB 22: HTAB 3: PRINT SPACE$(76)
3665 VTAB 23: HTAB 3: PRINT SPACE$(76)
3670 '
3675 RETURN
4000 REM #####
4005 REM #           Output Thesis Details           #
4010 REM #####
4015 '
4020 VTAB 21: HTAB 1: PRINT CHR$(22)
4025 IF OPERATION.CODE$ = "U" OR OPERATION.CODE$ = "R" OR OPERATION.CODE$ = "S"
      THEN 4030 ELSE 4120
4030   SC.STUDENT.NAME$ = STUDENT.NAME$
4035   SC.THESIS.NAME$ = THESIS.NAME$
4040   SC.ADVISOR.NAME$ = ADVISOR.NAME$
4045   SC.COADVISOR.NAME$ = COADVISOR.NAME$
4050   SC.DEPARTMENT.NAME$ = DEPARTMENT.NAME$
4055   SC.EDUCATION.YEAR$ = EDUCATION.YEAR$
4060   VTAB 10: HTAB 18: PRINT SPACE$(40)
4065   VTAB 10: HTAB 18: PRINT STUDENT.NAME$
4070   VTAB 12: HTAB 18: PRINT SPACE$(60)
4075   VTAB 12: HTAB 18: PRINT THESIS.NAME$
4080   VTAB 14: HTAB 18: PRINT SPACE$(40)
4085   VTAB 14: HTAB 18: PRINT ADVISOR.NAME$
4090   VTAB 16: HTAB 18: PRINT SPACE$(40)
4095   VTAB 16: HTAB 18: PRINT COADVISOR.NAME$
4100   VTAB 18: HTAB 18: PRINT SPACE$(40)
4105   VTAB 18: HTAB 18: PRINT DEPARTMENT.NAME$
4110   VTAB 20: HTAB 18: PRINT EDUCATION.YEAR$
4115   GOTO 4155
4120 'ELSE
4125   SC.STUDENT.NAME$ = SPACE$(40)
4130   SC.THESIS.NAME$ = SPACE$(60)
4135   SC.ADVISOR.NAME$ = SPACE$(40)
4140   SC.COADVISOR.NAME$ = SPACE$(40)
4145   SC.DEPARTMENT.NAME$ = SPACE$(40)
4150   SC.EDUCATION.YEAR$ = "...."
4155 'ENDIF
4160 '
4165 RETURN
4500 REM #####
4505 REM #           Read Thesis Details           #
4510 REM #####
4515 '
4520 OLD.STUDENT.NAME$ = SC.STUDENT.NAME$
4525 OLD.THESIS.NAME$ = SC.THESIS.NAME$

```



```

4530 OLD.ADVISOR.NAME$ = SC.ADVISOR.NAME$
4535 OLD.COADVISOR.NAME$ = SC.COADVISOR.NAME$
4540 OLD.DEPARTMENT.NAME$ = SC.DEPARTMENT.NAME$
4545 OLD.EDUCATION.YEAR$ = SC.EDUCATION.YEAR$
4550 VTAB 10: HTAB 18: INPUT "", SC.STUDENT.NAME$
4555 IF SC.STUDENT.NAME$ = "" THEN SC.STUDENT.NAME$ = OLD.STUDENT.NAME$
4560 VTAB 10: HTAB 18: PRINT SPACE$(40)
4565 VTAB 10: HTAB 18: PRINT SC.STUDENT.NAME$
4570 VTAB 12: HTAB 18: INPUT "", SC.THESIS.NAME$
4575 IF SC.THESIS.NAME$ = "" THEN SC.THESIS.NAME$ = OLD.THESIS.NAME$
4580 VTAB 12: HTAB 18: PRINT SPACE$(60)
4585 VTAB 12: HTAB 18: PRINT SC.THESIS.NAME$
4590 VTAB 14: HTAB 18: INPUT "", SC.ADVISOR.NAME$
4595 IF SC.ADVISOR.NAME$ = "" THEN SC.ADVISOR.NAME$ = OLD.ADVISOR.NAME$
4600 VTAB 14: HTAB 18: PRINT SPACE$(40)
4605 VTAB 14: HTAB 18: PRINT SC.ADVISOR.NAME$
4610 VTAB 16: HTAB 18: INPUT "", SC.COADVISOR.NAME$
4615 IF SC.COADVISOR.NAME$ = "" THEN SC.COADVISOR.NAME$ = OLD.COADVISOR.NAME$
4620 VTAB 16: HTAB 18: PRINT SPACE$(40)
4625 VTAB 16: HTAB 18: PRINT SC.COADVISOR.NAME$
4630 VTAB 18: HTAB 18: INPUT "", SC.DEPARTMENT.NAME$
4635 IF SC.DEPARTMENT.NAME$ = "" THEN SC.DEPARTMENT.NAME$ = OLD.DEPARTMENT.NAME$
4640 VTAB 18: HTAB 18: PRINT SPACE$(40)
4645 VTAB 18: HTAB 18: PRINT SC.DEPARTMENT.NAME$
4650 VT = 20: SP = 18: EP = 21: PICT$ = "9999": ST$ = SC.EDUCATION.YEAR$:
      GOSUB 5000: SC.EDUCATION.YEAR$ = ST$
4655 '
4660 RETURN
5000 REM #####
5005 REM # Add or Update Thesis Record #
5010 REM #####
5015 '
5020 LSET STUDENT.ID$ = SC.STUDENT.ID$
5025 LSET STUDENT.NAME$ = SC.STUDENT.NAME$
5030 LSET THESIS.NAME$ = SC.THESIS.NAME$
5035 LSET ADVISOR.NAME$ = SC.ADVISOR.NAME$
5040 LSET COADVISOR.NAME$ = SC.COADVISOR.NAME$
5045 LSET DEPARTMENT.NAME$ = SC.DEPARTMENT.NAME$
5050 LSET EDUCATION.YEAR$ = SC.EDUCATION.YEAR$
5055 '
5060 IF OPERATION.CODE$ = "I" THEN 5065 ELSE 5080
5065 INSERT #1
5070 IF ERR (>) 0 THEN HOME: PRINT "Insert record error ": END
5075 GOTO 5095
5080 'ELSE
5085 UPDATE #1
5090 IF ERR (>) 0 THEN HOME: PRINT "Update record error ": END
5095 'ENDIF
5100 '
5105 RETURN

```

```

5500 REM #####
5505 REM #           Delete Thesis Record           #
5510 REM #####
5515 '
5520 VTAB 21: HTAB 1: PRINT CHR$(22)
5525 VTAB 22: HTAB 3: PRINT "Confirm deletion by pressing 'YES' [ ... ]";
5530 VTAB 21: HTAB 1: PRINT CHR$(23)
5535 CONFIRM.DELETE$ = "..."
5540 VT = 22: SP = 41: EP = 43: PICT$ = "XXX": ST$ = CONFIRM.DELETE$:
      GOSUB 5000: CONFIRM.DELETE$ = ST$
5545 IF CONFIRM.DELETE$ = "YES" THEN 5550 ELSE 5565
5550   LSET STUDENT.ID$ = SC.STUDENT.ID$
5555   REMOVE #1
5560   IF ERR (<) 0 THEN HOME: PRINT "Delete record error ": END
5565 'ENDIF
5570 VTAB 21: HTAB 3: PRINT SPACE$(50)
5575 VTAB 22: HTAB 3: PRINT SPACE$(50)
5580 VTAB 23: HTAB 3: PRINT SPACE$(50)
5585 '
5590 RETURN
6000 REM #####
6005 REM #           Inquire Thesis Record           #
6010 REM #####
6015 '
6020 VTAB 21: HTAB 1: PRINT CHR$(22)
6025 VTAB 22: HTAB 3: PRINT "Press (RETURN) to inquire another student ";
6030 DELAY$ = INPUT$(1)
6035 IF ASC(DELAY$) (<) 13 THEN 6030
6040 VTAB 21: HTAB 1: PRINT CHR$(23)
6045 VTAB 21: HTAB 3: PRINT SPACE$(50)
6050 VTAB 22: HTAB 3: PRINT SPACE$(50)
6055 VTAB 23: HTAB 3: PRINT SPACE$(50)
6060 '
6065 RETURN
6500 REM #####
6505 REM #           Clear Thesis Details on Screen   #
6510 REM #####
6515 '
6520 VTAB 21: HTAB 1: PRINT CHR$(22)
6525 VTAB 8: HTAB 18: PRINT "....."
6530 VTAB 9: HTAB 18: PRINT SPACE$(40)
6535 VTAB 10: HTAB 18: PRINT STRING$(40, ".")
6540 VTAB 11: HTAB 18: PRINT SPACE$(60)
6545 VTAB 12: HTAB 18: PRINT STRING$(60, ".")
6550 VTAB 13: HTAB 18: PRINT SPACE$(60)
6555 VTAB 14: HTAB 18: PRINT STRING$(40, ".")
6560 VTAB 15: HTAB 18: PRINT SPACE$(40)
6565 VTAB 16: HTAB 18: PRINT STRING$(40, ".")
6570 VTAB 17: HTAB 18: PRINT SPACE$(40)
6575 VTAB 18: HTAB 18: PRINT STRING$(40, ".")

```

```

6580 VTAB 19: HTAB 18: PRINT SPACE$(40)
6585 VTAB 20: HTAB 18: PRINT "...."
6590 '
6595 RETURN
7000 REM #####
7005 REM #           Create Thesis File Structure           #
7010 REM #####
7015 '
7020 VTAB 21: HTAB 1: PRINT CHR$(22)
7025 VTAB 22: HTAB 3: PRINT "Confirm creating file structure by press 'YES' [ ... ]";
7030 VTAB 21: HTAB 1: PRINT CHR$(23)
7035 CONFIRM.CREATE$ = "..."
7040 VT = 22: SP = 53: EP = 55: PICT$ = "XXX": ST$ = CONFIRM.CREATE$:
      GOSUB 50000: CONFIRM.CREATE$ = ST$
7045 IF CONFIRM.CREATE$ = "YES" THEN 7050 ELSE 7070
7050  CLOSE #1
7055  CREATE #1, "THESIS.DAT", 256, 1, 7
7060  OPEN "X", #1, "THESIS.DAT", 256
7065  FIELD #1, 7 AS STUDENT.ID$, 40 AS STUDENT.NAME$, 60 AS THESIS.NAME$,
      40 AS ADVISOR.NAME$, 40 AS COADVISOR.NAME$,
      40 AS DEPARTMENT.NAME$, 4 AS EDUCATION.YEAR$, 25 AS FILLER$

7070 'ENDIF
7075 VTAB 21: HTAB 3: PRINT SPACE$(50)
7080 VTAB 22: HTAB 3: PRINT SPACE$(50)
7085 VTAB 23: HTAB 3: PRINT SPACE$(50)
7090 SAME.OPERATION.CODE = 0
7095 '
7100 RETURN
7350 FOR I = 1 TO 96
7500 REM #####
7505 REM #           Close Index File           #
7510 REM #####
7515 '
7520 CLOSE #1
7525 HOME
7530 FOR I = 1 TO 96
7535  PRINT "Good Luck ... ";
7540 NEXT I
7545 RETURN
50000 PT = SP: FREE.BYTE = FRE("")
50005 '
50010 VTAB VT: HTAB SP: PRINT ST$
50015 WHILE PT (<= EP
50020  VTAB VT: HTAB PT
50025  MASK$ = MID$(PICT$,PT-SP+1,1)
50030  IF MASK$ = "9" THEN LOLIM = 48: HILIM = 57: GOTO 50055
50035  IF MASK$ = "X" THEN LOLIM = 32: HILIM = 127: GOTO 50055
50040  MID$(ST$,PT-SP+1) = MASK$
50045  IF CH = 8 THEN PT = PT-1: GOTO 50020
50050  GOTO 50090

```

```
50055 CH$ = INPUT$(1): CH = ASC(CH$)
50060 IF CH = 13 THEN GOTO 50100
50065 IF CH = 8 AND PT > SP THEN PT = PT-1: GOTO 50020
50070 IF CH = 9 AND PT < EP THEN PT = PT+1: GOTO 50020
50075 IF CH < LOLIM OR CH > HILIM THEN BEEP 20,20: GOTO 50055
50080 MID$(ST$,PT-SP+1,1) = CH$
50085 VTAB VT: HTAB PT: PRINT CH$;
50090 PT = PT+1
50095 WEND
50100 '
50105 VTAB VT: HTAB SP: PRINT ST$;
50110 '
50115 VTAB VT: HTAB EP+1
50120 RETURN
```



ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

ประวัติผู้เขียน

นาย ประเสริฐ ฝูงวานิช เกิดวันที่ ๓๐ กันยายน ๒๕๐๒ ที่จังหวัดกรุงเทพมหานคร
สำเร็จการศึกษา วิทยาศาสตร์บัณฑิต (ฟิสิกส์) จากคณะวิทยาศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย
ในปี พ.ศ. ๒๕๒๔



ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย