

DBMS를 이용한 樹木園 管理(I)

— 植栽된 樹木의 管理를 中心으로 —

金 泰 旭 · 徐 敏 桓

(서울大學校 農科大學 林學科)

A DBMS Program on the Management of University Arboretum(I)

—With Respect to Woody Species Planted in the Arboretum—

Tae Wook Kim · Min Hwan Suh

(Dept. of Forestry, Coll. of Agri. Seoul Nat'l Univ.)

Summary

It is much time-consuming to manage the arboretum, and a computer program may be very useful to deal with the management of trees planted in the arboretum. The Data Base Management System(DBMS) is widely being used in various parts of work systems. The objectives of this study were to develop a simple arboretum management program using data base systems. This management system program contains registered plant list including Korean name, English name, scientific name, family name, height and diameter of plants, planting site.

I. 緒 論

本大學 樹木園에는 現在 約 95,000本(安養樹木園—70,000本, 水原樹木園—25,000本)의 植物이 植栽되어 있다. 이들은 現在 各各 植物 블록 單位로 나누어 植栽되어 있어서 이 自體만으로도 植物을 研究하는 이들에게 좋은 資料로서 제공될 수 있으나 이들을 좀 더 잘 活用하기 위해서는 植物 目錄에 대한 더욱 體系的인 管理가 필요하다. 그러나 樹木園에 植栽되어 있는 植物의 數字만으로도 엄청난 分量이며, 또한 植栽되어 있는 植物들 各各에 대하여 여러가지 情報가 添加된다고 하면 더욱 資料의 量이 커지게 된다. 이와 같이 많은 量의 資料를 일일이 手作業을 거쳐 處理 한다고 하는 것은 거의 불가능한 일이며, 또한 處理가 가능하다고 하더라도 能率을 따져 볼 때 결코 바람직한 것이 못된다. 이에 電算化 作業의 必要性이 대두된다.

樹木園 業務 전체를 電算化 하려면 우선 既存에 植栽되어 있는 樹木 및 새로 導入되는 樹種에 대한 管理 業務가 포함되며, 外國 樹木園과의 種子 交流와 그에 따라 導入되는 外來 樹種에 대한 植栽 및 管理 問題, 또 標本室에 있는 標本의 管理 및 새로운 標本の 製作

問題 등을 모두 포괄할 수 있도록 구성되어야 한다. 그러나 이러한 作業은 資料의 量으로 볼 때 엄청난 일이며 費用 問題 등을 고려할 때 短時間 內에 달성하기란 매우 어렵다. 이에 본 研究에서는 우선 樹木園內에 植栽되어 있는 植物들을 體系的으로 管理하기 위하여 既存에 植栽되어 있는 植物과 새로 導入되는 植物들의 目錄을 電算化 시키는 作業으로 限定지으려 하며, 앞으로 이러한 資料를 토대로 하여 계속하여 樹木園의 樹木管理 業務 전체를 電算化하는 作業을 진행시킬 예정이다.

II. 資料의 範圍 및 프로그램 概要

컴퓨터를 利用한 情報化 作業은 소프트웨어(software), 하드웨어(hard ware) 및 이 兩者를 연결시키는 運營體系(operating system)의 三位一體가 되어야 效率의으로 달성되며, 어느 한 分野의 發達만으로는 어려운 것이다. 小型컴퓨터 특히, 마이크로 프로세서(micro-processor)의 發達로 70年代 後半부터 컴퓨터의 接近性を 飛躍의으로 向上시켰으며, 이러한 趨勢에 偏乘한 소프트웨어 工學의 發達は 大型 컴퓨터(main-frame)에서만 가능하였던 DBMS를 小型컴퓨터에서

利用할 수 있게 되었다.

DBMS는 크게 關係 DBMS(relational DBMS)와 階層 DBMS(hierarchical DBMS)로 나눌수 있으며(李錫浩, 1985) DBMS를 構築함으로써 一般 言語 보다 많은 長點을 가진다. 즉 데이터의 獨立性을 가질 수 있는데, 이는 重複을 最少化하며, 一貫性, 標準化와 維持補修性에서 利點을 가진다.

1. SYSTEM 개요

本 樹木管理시스템은 10M바이트 以上の 하드디스크 (hard disk) 16비트를 장착한 IBM 혹은 互換機種으로 136column의 9핀 혹은 24핀 프린터를 필요로 한다.

2. 資料의 範圍

앞에서도 언급한 바와 같이 樹木園의 樹木 管理 業務는 매우 방대한 資資를 對象으로 하게 된다. 이러한 樹木 管理 業務를 模式圖로 그려 보면 그림 1과 같다. 이 그림에서 보는 바와 같이 樹木園의 樹木 管理는 크게 두가지로 나누어 진다. 첫째는 樹木을 植栽하고 管理하는 일이며, 두번째는 이들 樹木을 對象으로 하여 研究를 遂行하는 일이다. 또 樹木에 대한 研究는 既存에 植栽되어 있는 樹木을 對象으로 할 수도 있고, 外國 樹木園과의 種子交流에 의하여 導入되는 外來 樹種

을 통하여 適應 試驗 등을 할 수도 있으며, 또 한가지는 標本室에 所藏되어 있는 標本을 통하여 研究를 遂行할 수도 있다. 그러나 이러한 研究의 始作은 거의 大部分이 일단 樹木을 들여와서 植栽하고 管理하는 일에서 始作 되므로 樹木의 植栽 및 管理 業務와 별개의 것으로 생각할 수 없으며 各各의 段階는 有機的으로 연계되어 있다고 볼 수 있다.

本 研究에서 對象으로 삼고 있는 資料의 範圍는 樹木園에 이미 심겨져 管理되고 있는 植物들과 앞으로 導入되어 追加되는 植物들, 그리고 枯死 및 移植으로 인하여 줄어들게 될 植物들의 基本的인 目錄 및 植栽地, 植栽本數, 開化 및 結實時期 등으로 하였다. 더욱 細部的인 事項, 즉 樹木의 形態的, 生理的, 生態的 特性, 外來 導入樹種에 관한 部分 및 標本室에 관한 사항은 추후에 다룰 예정이다.

3. 프로그램 概要

本 研究에서는 Personal computer를 利用하여 樹木園內의 樹木管理 業務를 效率的으로 遂行할 수 있도록 프로그램을 제시하고 있다. 이 프로그램은 Database를 利用하여 작성한 것으로서 樹木園에 植栽된 樹種의 國名, 學名, 英名, 科名, 植栽年度, 樹高, 胸高直徑, 樹冠幅, 開化時期, 結實時期 등을 記錄할 수 있도록 만들어졌다. 全體的인 프로그램은 主프로그램(main program) 밑에 여러개의 副프로그램(sub-program)이 달려 있는 형태인데, 그림 2에서 보는 바와 같이 副프로그램은 主로 資料의 入·出力 및 修正, 檢索 作業을 위해 만들어져 있다.

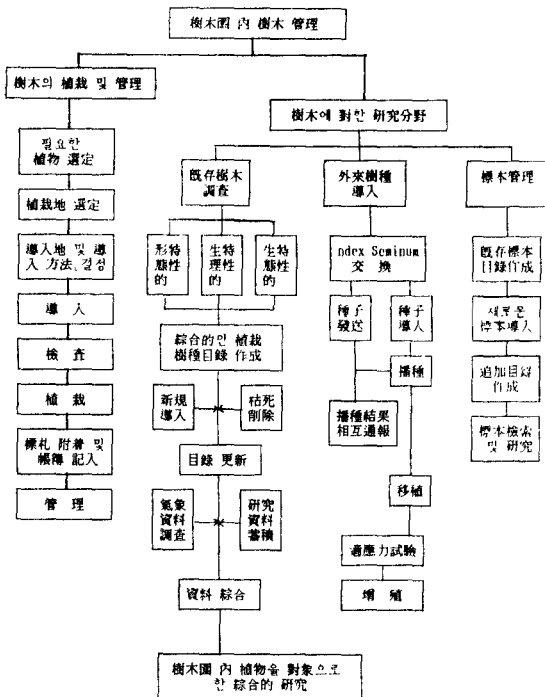


Fig. 1. Diagram on the mangement of arboretum around the mangement of plants.

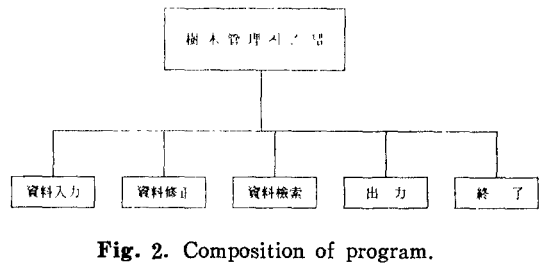


Fig. 2. Composition of program.

Ⅲ. 利用 可能性

컴퓨터의 補給과 더불어 DBMS를 利用하여 業務를 處理하는 分野가 점점 늘어가고 있는데 이러한 分野들이 거의 공통적으로 지니고 있는 特徵 중의 하나가 處理할 資料의 量이 많은 分野라는 것이다. 樹木園內의 植物을 管理하는 作業도 이와 마찬가지로 많은 量의 資料를 다루는 分野이며, 電算化 作業을 통하여 전체

적인 植物管理 體系를 統制해 나감으로써 作業時間도 줄일 수 있고, 人件費 등의 費用도 節減하는 效果를 가져올 수 있을 것이다.

앞에서도 언급한 바와 같이 本 研究에서 다루고 있는 것은 樹木園, 그 중에서도 특히 大學에 소속되어 있는 樹木園이 다루어야 할 내용 중 상당히 적은 部分에 불과하다. 그러나 樹木園 內의 樹木管理 業務를 處理하는데 여기에서 제시한 프로그램을 잘 利用함으로써 여러가지 役割을 遂行할 수 있다고 생각된다. 우선 樹木 전체의 目錄을 入力한 상태에서 每年 새로 導入되거나 枯死 등에 의하여 削除되는 植物을 添加 또는 削除함으로써 每年 간단한 作業에 의하여 樹種 目錄을 更新시킬 수 있으며, 樹木 하나하나에 대한 카드 資料를 出力함으로써 그 樹木의 歷史를 기록할 수 있다. 또한 每年 再調査에 의하여 바뀌어지는 部分에 대한 資料를 모으면 이들은 時間이 지나에 따라 그 자체로서 귀중한 研究 資料가 될 수 있을 것이다

參 考 文 獻

1. Hetzel, W. and D.R. Adams. 1985. Computer information systems development: Principles and case study. South-Wastern Publishing Co., Cincinnati. p.314.
2. Toada, H., Y. Kobayashi and T. Ohkawara. 1981. A card retrieval system for forest herbarium. Page 498 in 17th IUFRO World Congress, Div. 6, Kyoto, Japan.
3. 李敦求, 辛俊煥, 李慶學. 1983. 컴퓨터 프로그램을 이용한 林業苗圃의 效率의 管理. 韓國林學會誌 第62號: pp.53-59.
4. 李錫浩. 1985. 화인 처리론. 正益社, 서울. p.416.
5. 李昌福. 1980. 大韓植物圖鑑. 鄉文社, 서울. p.990.

APPENDIX

```

****PULL DOWN Main.PRG ****
SET TALK OFF
SET STATUS OFF
SET ESCAPE ON
SET SCOREBOARD OFF
SET BELL OFF
SET DELIMITER OFF
SET DATE ANSI
SET MESSAGE TO 24
SET PROCEDURE TO BOXES
SET DELETED ON
PUBLIC TITLE, MEM1
*****
STORE '서울대학교 수목원 ' to TITLE
*****

CLEAR

TRUE = .T.
DO WHILE TRUE

    CLEAR
    @ 0,0 SAY " "
    @ 1,0 SAY " "
    @ 2,0 SAY " "
    @ 23,0 SAY " "

    @ 1, 4 PROMPT ' 1.입 력 'MESSAGE ;
        +'樹木DB와 樹木자료를 입력합니다.
    @ 1,20 PROMPT ' 2.수 정 'MESSAGE ;
        +'내용을 수정 합니다.
    @ 1,35 PROMPT ' 3.검 색 'MESSAGE ;
        +' 자료를 검색합니다.
    @ 1,50 PROMPT ' 4.출 력 'MESSAGE ;
        +'수목목록이나 보고서 프린터 또는 화면으로 출력
    @ 1,65 PROMPT ' 5.종 료 'MESSAGE ;
        +'작업을 끝내고 도스로 돌아갑니다.

    MENU TO CHOICE
    SAVE SCREEN TO MENU1

    DO CASE
    *****
    CASE CHOICE = 1
    *****
        DO BOX1 WITH 0 && DRAW MENUBOX

        @ 3,3 PROMPT '1.메 뉴 ' MESSAGE ;
            +'메인메뉴로 복귀
        @ 5,3 PROMPT '2.Code DB입력' MESSAGE ;
            +'수목데이터베이스의 자료를 입력합니다.
        @ 7,3 PROMPT '3.수목등록 ' MESSAGE ;
            +'새로운 수목을 등록합니다.

        slct = 1
        MENU TO slct
        DO CASE
            CASE slct = 1
            CASE slct = 3
                DO INPUT2 && TREE DATA
            CASE slct = 2
                RESTORE SCREEN
                DO INPUT1
    
```

```

ENDCASE
CHOICE = 1
*****
CASE CHOICE = 2
*****
DO BOX1 WITH 20
@ 3,23 PROMPT ' 1.매 뉴 ' MESSAGE ;
+ '매인메뉴로 복귀
@ 5,23 PROMPT ' 2.수록 D B' MESSAGE ;
+ '수록DATA BASE의 내용을 수정합니다.
@ 7,23 PROMPT ' 3.수록수정 ' MESSAGE ;
+ '수록의 내용은 수정합니다.'
cho = 1
MENU TO cho
DO CASE
CASE cho = 1
CASE cho = 2
DO EDITA
CASE cho = 3
DO EDITB
ENDCASE
CHOICE = 2
*****
CASE CHOICE = 3
*****
DO BOX1 WITH 34
@ 3,37 PROMPT ' 1. C O D E ' MESSAGE ;
+ 'CODE를 이용하여 수록D를 조회
@ 5,37 PROMPT ' 2. 수록명 ' MESSAGE ;
+ '수록명을 이용하여 수록DB를 조회
@ 7,37 PROMPT ' 3. 분류번호' MESSAGE ;
+ '분류번호로 수록식재상황을 스크린에 출력
@ 9,37 PROMPT ' 4. 수록명 ' MESSAGE ;
+ '수록명으로 수록식재상황을 스크린에 출력

ch = 1
MENU TO CH
DO CASE
CASE CH = 1
DO SEARCHA
CASE CH = 2
DO SEARCHB
CASE CH = 3
DO SEARCHC
CASE CH = 4
DO SEARCHD
ENDCASE
CHOICE = 3
*****
CASE CHOICE = 4
*****
DO BOX1 WITH 50
@ 3,53 PROMPT ' 1. 분류번호' MESSAGE ;
+ '분류번호순으로 출력
@ 5,53 PROMPT ' 2. 과별출력' MESSAGE ;
+ '과별로 출력합니다.
@ 7,53 PROMPT ' 3. 가나다순' MESSAGE ;
+ '수록의 가나다순으로 출력합니다.
*****
FOR EXPANSION AT FUTURE
* @ 7,53 PROMPT ' 4. ' MESSAGE ;
* +
* @ 8,53 PROMPT ' 5. ' MESSAGE ;
* +
* @ 9,53 PROMPT ' 6.전 체 ' MESSAGE ;
*
*

```

```

slet = 1
MENU TO slet
DO CASE
  CASE slet = 1
    GO PRIN1
  CASE slet = 2
    DO PRIN2
  CASE slet = 3
    DO PRIN3
ENDCASE
CHOICE = 1

*****
CASE CHOICE = 5
*****
DO BOX1 WITH G2      && DRAW MENUBOX PARATERS COLUMN G2

@ 3,65 PROMPT ' 1.대   뉴 ' MESSAGE ;
+'메인메뉴로 복귀
@ 5,65 PROMPT ' 2.P A C K ' MESSAGE ;
+'삭제 표시된 내이티플 데이터 화일에서 완전히 제거
@ 7,65 PROMPT ' 3. RECALL ' MESSAGE ;
+'삭제 표시된 내이타를 부호화한다.
@ 9,65 PROMPT ' 4.   쿨   ' MESSAGE ;
+'작업을 끝내고 도스프로 돌아갑니다.

CHO = 1
MENU TO CHO
DO CASE
  CASE CHO =
    CHOICE = 1
  CASE CHO = 2
    USE TREE INDEX CODETR,NOTR
    PACK
    CLOSE DATABASE
    USE TREEDB INDEX CODEDE
    PACK
    CLOSE DATABASE
    CHOICE = 5
  CASE CHO = 3
    USE TREE INDEX CODETR,NOTR
    RECALL ALL
    CLOSE DATABASE
    USE TREEDB INDEX CODEDB
    RECALL ALL
    CHOICE = 5
  CASE CHO = 4
    TRUE = .F.
    SET TALK ON
    SET STATUS ON
    SET SCOREBOARD ON
    SET ESCAPE ON
    SET BELL ON
    SET DELIMITER ON
    SET DELETED OFF
ENDCASE
*****
ENDCASE
ENDDO && DO WHILE TRUE
**** input1.prg ****
** input tree'database ****
SET ESCAPE ON
USE TREEDB INDEX CODEDB
GOTO BOTTOM
ICODE=CODE
PUBLIC TITLE

```

```
RESTORE SCREEN FROM MENU1
TRUE = .T.
DO WHILE TRUE
```

```
** APPEND BLANK
```

```
** TRUE = .T.
DO WHILE TRUE
```

```
@ 3,0 CLEAR TO 22,79
@ 3,21 SAY "
@ 4,21 SAY " DATA BASE 입력 "
@ 5,21 SAY "
@ 4,24 SAY TITLE
```

```
*****
```

```
DO BOX3
* READ
DO BOX3SAY
DO WHILE TRUE
  @ 7,12 GET ICODE
  READ
  IF LEN(TRIM(ICODE))=0
    @ 21,0 SAY " CODE를 다시입력하십시오 !!! "
    LOOP
  ENDIF
  FIND &ICODE
  IF .NOT. EOF()
    @ 21,1 SAY " 이미 등록된 CODE입니다! 다시입력하십시오 !! "
    DO BOX3SAY
    LOOP
  ELSE
    APPEN BLANK
  EXIT
ENDIF
ENDDO. && CHECK DATE EXITED
DO BOX3GET
READ
```

```
* MNAME = NAME
```

```
*****
```

```
DO WHILE .T.
  @22, 5 PROMPT ' 수정 ' MESSAGE ;
  + '입력된 자료를 수정합니다.
  @22,20 PROMPT ' 등록 ' MESSAGE ;
  + '현재 화면의 내용을 등록합니다.
  @22,35 PROMPT ' 등록 및 Main MENU' MESSAGE ;
  + '화면의 내용을 등록한 후 전화면으로 갑니다.
  @22,57 PROMPT ' 취소 ' MESSAGE ;
  + '현재 화면의 내용을 취소하고 전화면으로 갑니다.
  MENU TO mcho
DO CASE
  CASE mcho = 1
    CLEAR GETS
    EXIT
  DELETE
  CASE mcho = 2
    replace code with icode
    EXIT
  CASE mcho = 3
    replace code with icode
    EXIT
  CASE mcho = 4
    CLEAR GETS
    DELETE
    EXIT
```

```

        ENDCASE
    ENDDO    && .T.
    IF mcho = 1
        mcho = 2
        LOOP
    ENDIF
    IF mcho = 3 .OR. mcho = 4
        EXIT
    ENDIF
    ENDDO && WHILE TRUE
    IF mcho = 3 .OR. mcho = 4
        EXIT
    ENDIF
    ENDDO    && TRUE
    CLOSE DATA
    RETURN
    ***** input2.prg *****
    ***** this program input tree data *****
    ***** *****
    SELECT A
    USE TREE INDEX codetr, notr
    GOTO BOTTOM
    INAME=SPACE(28)
    INO=SPACE(5)
    ICODE=SPACE(5)
    SELECT B
    USE TREEDB INDEX KNAMEDB
    SET ESCAPE ON
    SET RELATION TO CODE INTO A
    PUBLIC TITLE
    RESTORE SCREEN FROM MENU1
    TRUE =.T.

DO WHILE TRUE
    @3,0 CLEAR TO 22,79
    @ 3,21 SAY "
    @ 4,21 SAY "  수목자료입력 "
    @ 5,21 SAY "
    @ 4,24 SAY TITLE

*****
    DO BOX2
    SELECT A
    DO BOX2SAY
    SELECT B
    DO WHILE .T. && CHECK DATA, IF EXITED
    INAME=SPACE(20)
    SELECT B
    @ 7,44 GET INAME
    READ
    INAME=RTRIM(INAME)
    IF LEN(INAME)=0
    @ 24, 0 SAY " 자료입력이 없되었습니다. 다시 입력하십시오 ! "
    READ
    LOOP
    ENDIF
    FIND &INAME
    IF EOF()
    @ 24,1 SAY " 수목DATA BASE에 등록되지 않았음!! 수목DB에서 입력후;
    다시 입력하십시오!! "
    READ
    RETURN
    ENDIF
    STORE CODE TO ICODE
    SELECT A
    DO BOX2SAY
    @ 7,12 GET INO
    READ
    INO=RTRIM(INO)
    IF LEN(INO)=0

```



```

@ 24, 0 say " 분류번호를 다시 입력하십시오 !!! "
READ
INO=SPACE(5)
LOOP
ENDIF
SET ORDER TO ?
FIND &INO
IF .NOT.EOF()
  @ 24, 1 SAY " 이미 등록이 된 수목입니다 !! 분류번호를 확인하십시오! "
  READ
  SET ORDER TO 1
  LOOP
ELSE
  SET ORDER TO 1
  EXIT
ENDIF
ENDDO      && CHECK DATA, IF
DO BOX2SAY
APPEND BLANK
DO BOX2GET
READ
*****

DO WHILE .T.
  @22, 5 PROMPT ' 수 정 ' MESSAGE ;
  + ' 입력한 것이 틀렸습니다.
  @22,20 PROMPT ' 등 록 ' MESSAGE ;
  + ' 화일에 등록을 합니다
  @22,35 PROMPT ' 취 소 ' MESSAGE ;
  + ' 현재 화면의 내용을 등록 합니다.
  @22,50 PROMPT ' 등록 및 Main MENU ' MESSAGE ;
  + ' 현재 내용을 등록후 이전MENU로 갑니다.
MENU TO mcho
DO CASE
  CASE mcho = 1
    CLEAR GETS
    DELETE
    EXIT
  CASE mcho = 2
    REPLACE CODE WITH ICODE, NO WITH INO,KNAME WITH INAME
    EXIT
  CASE mcho = 3
    CLEAR GETS
    DELETE
    EXIT
  CASE mcho = 4
    REPLACE CODE WITH ICODE, NO WITH INO,KNAME WITH INAME
    EXIT
ENDCASE
ENDDO
IF mcho = 3 .OR. mcho = 4
  CLOSE DATABASE
  EXIT
ENDIF
ENDDO
***** EDITA.PRG *****
USE TREEEDR INDEX KNAMEDB,CODED9
PUBLIC TITLE
INAME=SPACE(28)
  ? 3,0 CLEAR TO 22,79
  @ 3,21 SAY " "
  @ 4,21 SAY " DATA BASE 입력 "
  ? 5,21 SAY " "
  @ 4,24 SAY TITLE

DO WHILE .T.
  @13,20 SAY " "
  @14,20 SAY " 수정할 수종명? "
  @15,20 SAY " "
  @14,38 GET INAME
  READ

```

```

IF LEN(TRIM(INAME)) = 0
  EXiT
ENDIF
*****
mcho2 = 0
FIND &INAME
IF EOF()
  @ 18,10 SAY "찾는 수증명이 없습니다. !! "
  LOOP
ENDIF
*****
DO BOX3
DO BOX3GET
READ
*****

DO WHILE .T.
  @22, 5 PROMPT '수 정 ' MESSAGE ;
+ ' 찾으신 수증을 다시 수정합니다.
  @22,20 PROMPT ' 취 소 ' MESSAGE ;
+ ' 화면의 내용의 수정을 취소합니다.
  @22,35 PROMPT ' 등 록 ' MESSAGE ;
+ ' 수정된 내용을 등록합니다.
  MENU TO mcho
  DO CASE
    CASE mcho = 1
      exit
    CASE mcho = 2
      CLEAR GETS
      EXiT
    CASE mcho = 3
      EXiT
  ENDCASE
ENDDO

IF mcho = 1
  mcho = 1
  LOOP
ENDIF
IF mcho = 3
  mcho = 1
  EXiT
ENDIF
ENDDO && WHILE KK
***** EDITB.PRG *****
USE TREE INDEX KNAME?P.
PUBLIC TITLE
INAME=SPACE(28)
DO WHILE .T.
  @13,20 SAY "
  @14,20 SAY " 수정할 수증명? "
  @15,20 SAY "
  @14,38 GET INAME
  READ

  IF LEN(TRIM(INAME)) = 0
    EXiT
  ENDIF
  mcho2 = 0
  *****
  FIND &INAME
  IF EOF()
    @ 18,10 SAY "찾는 수증명이 없습니다. !! "
    INAME=SPACE(28)
  LOOP
  ENDIF
  *****
  DO BOX2
  DO BOX2GET
  READ
  *****

```

```

DO WHILE .T.
  @22, 5 PROMPT '삭제' MESSAGE ;
+ '찾으신 수증을 삭제합니다.
  @22,20 PROMPT '취소' MESSAGE ;
+ '화면의 내용의 수정을 취소합니다.
  @22,35 PROMPT '등록' MESSAGE ;
+ '수정된 내용을 등록합니다.
  MENU TO mcho
  DO CASE
    CASE mcho = 1
      DELETE
    CASE mcho = 2
      CLEAR GETS
      EXIT
    CASE mcho = 3
      EXIT
  ENDCASE
ENDDO

IF mcho = 1
  mcho = 1
  LOOP
ENDIF
IF mcho = 3
  mcho = 1
  EXIT
ENDIF
ENDDO

*** SEARCHA.PRG *****
USE TREEDB INDEX CODEDB
PUBLIC TITLE
ICODE=SPACE(5)
INAME=SPACE(28)
RESTORE SCREEN FROM MENU!
  @ 3,0 CLEAR TO 22,79
  @ 3,21 SAY " "
  @ 4,21 SAY " DATA BASE 검색 "
  @ 5,21 SAY " "
  @ 4,24 SAY TITLE

DO WHILE .T.
  @13,20 SAY " "
  @14,20 SAY " 검색할 수증의 CODE? "
  @15,20 SAY " "
  @14,45 GET ICODE
  READ

  IF LEN(TRIM(ICODE)) = 0
    EXIT
  ENDIF
  *****
  mcho2 = 0
  FIND &ICODE
  IF EOF()
    @ 18,10 SAY "찾는 C O D E는 없습니다. !! "
    LOOP
  ENDIF
  ICODE=CODE
  INAME=KNAME
  *****
  DO BOX3
  DO BOX3SAY
  READ
  *****
  ICODE=SPACE(5)
ENDDO

```

```

***** SEARCHA.PRG *****
USE TREEDB INDEX KNAMEDB
PUBLIC TITLE
ICODE=SPACE(5)
INAME=SPACE(28)
RESTORE SCREEN FROM MENU1
@ 3,0 CLEAR TO 22,79
@ 3,21 SAY " "
@ 4,21 SAY " DATA BASE 검색 "
@ 5,21 SAY " "
@ 4,24 SAY TITLE

DO WHILE .T.
@13,20 SAY " "
@14,20 SAY " 수목명은 ? "
@15,20 SAY " "
@14,35 GET INAME
READ

IF LEN(TRIM(INAME)) = 0
EXIT
ENDIF
*****
mcho2 = 0
FIND &INAME
IF EOF()
@ 18,10 SAY "찾는 수목명이 없습니다. !! "
LOOP
ENDIF
ICODE=CODE
INAME=KNAME
*****
DO BOX3
DO BOX3SAY
READ
*****
INAME=SPACE(28)
ENDDO

***** SEARCHA.PRG *****
SELECT A
USE TREE INDEX NOTR
PUBLIC TITLE
SELECT B
USE TREEDB INDEX CODEDB
SELECT A
SET RELATION TO CODE INTO B
ICODE=SPACE(5)
INO=SPACE(5)
INAME=SPACE(28)
RESTORE SCREEN FROM MENU1
@ 3,0 CLEAR TO 22,79
@ 3,21 SAY " "
@ 4,21 SAY " 식재수목 검색 "
@ 5,21 SAY " "
@ 4,24 SAY TITLE

DO WHILE .T.
INO=SPACE(5)
@13,20 SAY " "
@14,20 SAY " 검색할 수목의 분류번호는 ? "
@15,20 SAY " "
@14,55 GET INO
READ

IF LEN(TRIM(INO)) = 0
EXIT
ENDIF
*****
SEEK INO
* SUSP
@ 19,19 SAY "INO "+INO+ " NO "+NO
IF EOF()
@ 18,10 SAY "찾는 분류번호는 없습니다. !! "
LOOP
ENDIF
ICODE=CODE
INAME=KNAME
*****
DO BOX2
DO BOX2SAY
READ
*****
ENDDO

```

```

**** SEARCHD.PRG *****
SELECT A
USE TREE INDEX KNAME
SELECT B
USE TREEDB INDEX CODEDB
SELECT A
SET RELATION TO CODE INTO B
PUBLIC TITLE
ICODE=SPACE(5)
INO=SPACE(5)
INAME=SPACE(28)
RESTORE SCREEN FROM MENU1
@ 3,0 CLEAR TO 22,79
@ 3,21 SAY " "
@ 4,21 SAY " 식재수목 검색 "
@ 5,21 SAY " "
@ 4,24 SAY TITLE

DO WHILE .T.
INAME=SPACE(28)
@13,20 SAY " "
@14,20 SAY " 수목명은? "
@15,20 SAY " "
@14,35 GET INAME
READ

IF LEN(TRIM(INAME)) = 0
EXIT
ENDIF
*****
SEEK INAME
IF EOF()
@ 5,1 CLEAR TO 18,79
@ 18,10 SAY "찾는 수목은 없습니다. !! "
LOOP
ENDIF
ICODE=CODE
INAME=KNAME
*****
DO BOX2
DO BOX2 SAY
READ
*****
ENDDO

**** PRIN1.PRG ****
SET SAFE OFF
SELECT A
USE TREE
INDEX ON NO TO TEMP
SELECT B
USE TREEDB INDEX CODEDB
SELECT A
SET RELATION TO CODE INTO B
@ 15,18 SAY " "
@ 16,18 SAY " 분류번호순 출력 "
@ 17,18 SAY " "
@ 20,18 SAY " 프린터를 준비하시고 아무키나 치십시오 ! "
WAIT " "
SET DEVICE TO PRIN
DO WHILE .NOT. EOF()
@ 2,0 say ;
" "
@ 3,0 SAY;
" 분류 국 명 학 명 과 명 수 "
@ 4,0 SAY;
" "
DO WHILE .NOT. EOF() .AND. PROW() < 53
@ prow() + 1,01 say NO
@ prow(),7 TRIM(B->KNAME)
@ prow(),27 say;
TRIM(B->GNAME)+ " "+TRIM(B->SNAME)+ " "+TRIM(B->NOMINATOR)
@ prow(),62 say TRIM(B->KFNAME)+STR(TRNUM,5)
SKIP
ENDDO
@ prow()+1,0
" "
ENDDO
SET DEVICE TO SCREEN
*ERASE TEMP.IDX
SET SAFE ON
RETURN

```

```

**** PRIN2.PRG ****
SET SAFE OFF
SELECT A
USE TREEEDB
INDEX ON KFNAME TO TEMP
SELECT B
USE TREE INDEX CODETR
SELECT A
SET RELATION TO CODE INTO B
@ 13,18 SAY "=====
@ 14,18 SAY "          과 별 출 력          "
@ 15,18 SAY "=====
@ 18,18 SAY "          프린터를 준비하시고 아무키나 치십시오 !"
WAIT " "
SET DEVICE TO PRIN
DO WHILE .NOT. EOF()
"=====
@ 3,0 SAY;
"분류   국   명   학   명   과 명 수 량"
@ 4,0 SAY;
"=====

DO WHILE .NOT. EOF() .AND. PROW()<>53
@ prow() + 1,01 say B->NO
@ prow(),7 TRIM(KNAME)
@ prow(),27 say;
TRIM(GNAME)+" "+TRIM(SNAME)+" "+TRIM(NOMINATOR)
@ prow(),62 say TRIM(KFNAME)+STR(b->TRNUM,5)
SKIP
ENDDO
@ PROW()+1,0 SAY;
"=====

ENDDO
SET DEVICE TO SCREEN
*ERASE TEMP.IDX
SET SAFE ON
RETURN

**** PRIN3.PRG ****
SET SAFE OFF
SELECT A
USE TREEEDB
INDEX ON KNAME TO TEMP
SELECT B
USE TREE INDEX CODETR
SELECT A
SET RELATION TO CODE INTO B
@ 13,18 SAY "=====
@ 14,18 SAY "          과 별 출 력          "
@ 15,18 SAY "=====
@ 18,18 SAY "          프린터를 준비하시고 아무키나 치십시오 !"
WAIT " "
SET DEVICE TO PRIN
DO WHILE .NOT. EOF()
"=====
@ 2,0 say ;
"=====
@ 3,0 SAY;
"분류   국   명   학   명   과 명 수 "
@ 4,0 SAY;
"=====

DO WHILE .NOT. EOF() .AND. PROW()<>53
@ prow() + 1,01 say B->NO
@ prow(),7 TRIM(KNAME)
@ prow(),27 say;
TRIM(GNAME)+" "+TRIM(SNAME)+" "+TRIM(NOMINATOR)
@ prow(),62 say TRIM(KFNAME)+STR(B->TRNUM,5)
SKIP
ENDDO
@ PROW()+1,0 SAY;
"=====

ENDDO
SET DEVICE TO SCREEN
*ERASE TEMP.IDX
SET SAFE ON
RETURN

```

** BOXES.PRG ***

 * PROCEDURE BOX1 *

PROCEDURE BOX1

PARAMETERS COL

@ 3,0 CLEAR TO 22,79
 @ 2,COL SAY " " "
 @ 3,COL SAY " " "
 @ 4,COL SAY " " "
 @ 5,COL SAY " " "
 @ 6,COL SAY " " "
 @ 7,COL SAY " " "
 @ 8,COL SAY " " "
 @ 9,COL SAY " " "
 @10,COL SAY " " "

 * PROCEDURE BOX2 *

PROCEDURE BOX2

@ 6,0 SAY " " "
 @ 7,0 SAY " 분류번호 13345 CODE 12345 국 명 45 " "
 @ 8,0 SAY " " " "
 @ 9,0 SAY " 속 명 13 과(K)명 45 " "
 @10,0 SAY " " " "
 @11,0 SAY " " " "
 @12,0 SAY " " " "
 @13,0 SAY " 시계년도 13345 식 제 지 45 334567890123456789012345678 " "
 @14,0 SAY " " " "
 @15,0 SAY " 도입지 13 식재분수 " " "
 @16,0 SAY " " " "
 @17,0 SAY " 수 고 13 근원경 34 총 고 54 수간폭 72 " "
 @18,0 SAY " " " "

PROCEDURE BOX2GET

* @ 7,12 GET NO
 * @ 7,28 GET CODE
 * @ 7,44 GET KNAME
 @13,12 GET YEAR
 @13,44 GET PLACE
 @15,12 GET SOURCE
 @15,63 GET TRNUM
 @17,13 GET HEIGHT
 @17,34 GET DRH
 @17,54 GET DBH
 @17,72 GET CRWIDTH

PROCEDURE BOX2SAY

@ 7,12 SAY INO
 @ 7,28 SAY ICODE
 * @ 7,44 SAY INAME
 @ 9,12 SAY B->GNAME
 @ 9,44 SAY B->KNAME
 @11,12 SAY B->SNAME
 @13,12 SAY YEAR
 @13,44 SAY PLACE
 @15,12 SAY SOURCE
 @15,63 SAY TRNUM
 @17,13 SAY HEIGHT
 @17,34 SAY DRH
 @17,54 SAY DBH
 @17,72 SAY CRWIDTH

 * PROCEDURE BOX3

PROCEDURE BOX3

@ 6,0 SAY " " "
 @ 7,0 SAY " C O D E 13345 국 명 3334567890123456789012345678 " "
 @ 8,0 SAY " " " "
 @ 9,0 SAY " 과(E)명 13 과(K)명 45 " "
 @10,0 SAY " " " "
 @11,0 SAY " 속(E)명 13 " "
 @12,0 SAY " " " "
 @13,0 SAY " 중(E)명 " " "
 @14,0 SAY " " " "
 @15,0 SAY " 명 명 자 13 " "
 @16,0 SAY " " " "
 @17,0 SAY " 영(E)명 13 " "
 @18,0 SAY " " " "

```
*****  
* PROCEDURE BOX3SAY  
*****  
PROCEDURE BOX3SAY  
  @ 7,12 SAY CODE  
  @ 7,45 SAY KNAME  
  @ 9,12 SAY FNAME  
  @ 9,44 SAY KFNAME  
  @11,12 SAY GNAME  
  @13,12 SAY SNAME  
  @15,12 SAY NOMINATOR  
  @17,12 SAY ENAME  
*****  
PROCEDURE BOX3GET  
  @ 7,45 GET KNAME  
  @ 9,12 GET FNAME  
  @ 9,44 GET KFNAME  
  @11,12 GET GNAME  
  @13,12 GET SNAME  
  @15,12 GET NOMINATOR  
  @17,12 GET ENAME
```