CHAPTER IV

SOFTWARE DEVELOPMENT

The pharmacokinetics of gentamicin are complex, requiring at least a two-compartment model for a mathematically accurate description of serum concentrations (Siber et al., 1975; Schentag et al., 1977b); however, the one-compartment model adequately describes gentamicin pharmacokinetics for clinical purposes (Siber et al., 1975; Sawchuk and Zaske, 1976; Chow et al., 1978). Although dosage calculations and serum level predictions based on a one-compartment model are less complicated, these still require the use of logarithms and exponents, precluding rapid calculation without an electronic calculator or a computer.

Use of automated methods to generate therapeutic guidelines for dosing aminoglycosides is well accepted in the literature (Jelliffe, 1970; Mawer et al., 1974 quoted in Mawer, 1976; Foster and Bourne, 1977; Bennett and Scott, 1980; Ng, 1980; Kaka and Buchanan, 1983; Schentag and Adelman, 1983; Robinson et al., 1984). These methods are more accurate and rapid than manual calculations. To date, several programs designed for programmable hand-held calculators, large mainframe computers, or microcomputers are available, but not in Thailand.

The programmable hand-held calculators have limitations; for example, they have relatively little memory and storage capacity for constants and subprogram calculations. This limits their ability to

handle more complex sequences of calculations. They cannot provide for storage of patient files or for updating the file as the patient is treated. Finally, their programming is more complicated because it is more cryptic than simple computer languages like Pascal. Microcomputers have more than enough storage for pharmacokinetic calculations since large data bases are usually not necessary. In comparison with large computers or mainframes, microcomputers are easier to operate, much more common, and less expensive. At present, microcomputers are available in several hospitals of Thailand and as costs of computer hardware are coming down dramatically, they are becoming sufficiently affordable. Hence, the software for predicting gentamicin dosage regimens and serum levels based on the one-compartment pharmacokinetic model is developed on a microcomputer.

Materials and Methods

All programming employed Turbo Pascal version 5.5 (copyright by Borland International, Inc.) on an IBM Personal Computer (IBM-PC) compatible with a 80286 microporcessor, 640 K memory, a monochrome monitor, one 5.25-inch floppy disk drive and one 30 M hard disk drive. A dot matrix printer, Epson LQ-550 or compatible, was employed to print consultation sheets for entry into patient charts. The primary goal was to make the software user-friendly, requiring minimal operator training. In order to provide convenience to the user, the entire program is loaded into memory from the program disk; thus, all subsequent file management can be handled from seperate file disks. Therefore, disk manipulation was held to an absolute minimum.

Problem Analysis

1. Problem Definition

Write a microcomputer program to evaluate an existing gentamicin dosage regimen and to determine a new dosage regimen and steady-state peak and trough serum levels using an individualized pharmacokinetic approach. This approach is established for drug administration by multiple constant-rate intravenous infusions and applicable where the elimination kinetics are first order and can be represented by a one-compartment open model. Calculations are based on litererature-averaged estimates of population data or measured serum levels of individual patients. Moreover, recommendations for monitoring of therapy such as blood sampling times are provided.

The written program also provides storage and retrieval of patient data, with immediate access to facilitate updating and dosing regimen changes and all changes in renal functions as they occur. Additionally, the program generates hardcopies or printouts.

The program focuses on two of the most often encountered clinical situations; initiation of gentamicin therapy and consultation after therapy has been initiaited. Theoretical concepts have been described below and processes in detail are shown in Appendix A (Gentamicin Dosing Guidelines).

1.1 Initiation of Therapy.

In this situation, the program assists the user in selecting empiric dosing that is based on two fundamental pharmacokinetic parameters: clearance and distribution volume. Gentamicin clearance is derived directly from the patient's measured

or estimated creatinine clearance (Cl_{cr}). Estimates of Cl_{cr} are derived from age, sex, weight, height, stable serum creatinine level, and specific conditions or disease states of the patient. Then the patient's first-order elimination rate constant (k_{el}) can be estimated on the basis of population relationships that relate Cl_{cr} to k_{el}. The estimate of the patient's volume of distribution (V_d) is determined by adjusting for body weight and specific-conditions. After calculating the dosing interval (T) to achieve the desirable steady-state peak (Cp'_{max}) and trough (Cp'_{min}) levels, a loading dose (LD) and the maintenance dose (MD) can be estimated and employed until results of serum concentration determinations are available.

In addition to establishing an initial dosing regimen, the program predicts the corresponding steady-state peak (Cp_{maxx}) and trough (Cp_{minx}) levels by using the equations of Sawchuk and Zaske. It also provides recommended blood sampling times.

Timing of Serum Gentamicin Levels Measurements.

After administration of the LD, a series of three serum gentamicin concentrations (SGC) are obtained to characterize gentamicin half-life, k_{el}, and V_d in each patient. The time interval over which these samples are drawn should encompass approximately one half-life. Although the correlation between S_{cr} and half-life has been reported to be moderate at best, it can be used to provide a reasonable estimate of the period over which the blood samples should be drawn.

1.2 Consultation after Therapy Has Been Initiated.

When the patient already has received several doses of gentamicin, the program estimates the steady-state peak (Cp_{max*}) and trough (Cp_{min*}) serum levels by using the equations of Sawchuk and Zaske. Since the (actual) dosage regimen (MD and T) are known, the Cp_{max*} and Cp_{min*} can be calculated once the patient's V_d and k_{el} are estimated. V_d and k_{el} are determined based on measured SGCs or Cl_{cr} if SGC data are not available.

If the calculated peak is excessively high or low (i.e., Cp_{max*} is > 10 mg/L or < 4 mg/L), or if the trough (Cp_{min*}) is > 2 mg/L, or if the patient's serum creatinine is unstable, a new dose and dosing interval should be calculated for the patient to achieve the desired steady-state peak (Cp'_{max}) and trough (Cp'_{min}) levels.

If the calculated peak and trough are within the accepted range (i.e., Cp_{max*} is between 4 and 10 mg/L, and Cp_{min*} is < 2 mg/L), and the patient's renal function is stable, three serial blood samples should be drawn when the patient is at steady-atate. To estimate the time required to reach steady-state, multiply the estimated half-life by five.

1.3 Monitoring of Therapy.

For all patients, follow-up serum gentamicin concentration determinations are recommended to ensure that therapeutic, yet nontoxic, concentrations are achieved, and to assess the accuracy of the pharmacokinetic model.

For those patients with stable renal function, two SGC determinations should be carried out when the patient has reached steady - state on the new dosing schedule (five half-lives) after the initial evaluation while for patients with fluctuating renal function two SGC determinations should be obtained more frequent together with the S_{cr} ordered. The timing of these samples depends on the drug's half-life as reflected by the patient's dosing interval.   The first sample should be drawn at 0.5, 1, or 2 hours after end of infusion, to ensure that the distribution phase is completed. The second sample should be drawn at a time equal to half of the dosing interval. These values should be used to calculate the Cp_{max} and Cp_{min}.   Then the evaluation process of the calculated values begins.   If the calculated peak and trough values are $\pm$ 10 and 20 percent of the desired peak and trough concentrations, respectively, and the patient is improving clinically, with stable renal function, the established dosing regimen may be continued. SGCs should then be reassessed in three to five days, provided there are no drastic changes in the patient's clinical status.   On the other hand, if the expected SGCs are not attained (within a range), the measured SGCs should be used to reestimate the patient's k_{el} and V_{d}.   A new dosing regimen based on these parameters and the desired peak and trough SGCs then can be calculated and recommended for the patient on the same day the SGCs are analyzed. When the patient has reached steady-state, this dosing regimen should be reevaluated and the above cycle then can be repeated. This process (recommendation, assessment) will need to be repeated until a predictable kinetic pattern is established.

2. Input Specifications

The program should accept as input from the terminal (on-line keyboard) the following patient information:

Demographic Data.

The user must provide patient demographic data including: name, hospital no., ward/bed, and doctor name.

Physical Data.

The user must provide patient demographic data including: age in years, height in centimeters, weight in kilograms, and sex in character 'F' for female or 'M' for male.

Clinical Data:

The user must provide patient specific conditions such as hydration status (normal hydration, or dehydrated, or edematous or overhydrated), renal function (normal renal function, or chronic renal failure, or end-stage renal failure), type of infection and causative microorganism, disease states (diabetic, liver disease, shock), and concurrent drug therapy (vancomycin, amphotericin B).

Laboratory Data:

The user must provide laboratory data including: serum creatinine concentration (mg/dl), urine creatinine concentration (mg/dl), 24-hour urine volume (ml), and measured serum gentamicin concentration (mcg/ml). Also, the times at which serum is drawn must be recorded.

Treatment Data:

The user must provide patient treatment data including: administered loading dose (mg), maintenance dose (mg), dosing interval (hour), i.v. infusion time (hr), and also their start (and stop) time; and desired steady-state peak and trough levels (mcg/ml).

3. Output Specifications:

The program should provide as output to the screen or printer the following information:

Patient Input Data:

The program must provide all patient input data above.

Calculated Values:

The program must provide calculated values including: lean body weight and dosing weight (kg) for adults or ideal body mass and adjusted body mass (kg) for children, creatinine clearance (ml/min), first-order elimination rate constant (per hour), apparent volume of distribution (L), recommended dosage regimen (loading dose and maintenance dose in mg, dosing interval in hour) and the corresponding predicted steady-state peak and trough levels (mcg/ml).

Consult Recommendations:

The program must provide the following recommendations: blood sampling times (for initial estimation of k_e, and Vd, and for monitoring of therapy), desired steady-state peak and trough levels (according to infectious process or causative microorganism), and the evaluation of the current administered or chosen dosage regimen.

4. Special Handling:

If the input data is illegal, the program will show the error message along with how it can be corrected, such as these examples:

*** Sex must be F or M. ***

*** The answer must be Y or N ***

*** The input is out of range. ***

In some cases the user might not want to try again, the program will provide alternative choices such as:

Esc = Exit, PgDn = Next Screen

Program Design

In order to communicate with the user, several screens have to be created. Thus, the program was designed to composed of several routines or modules according to the screen showed. The main program consists of three modules as follows (Figure 1):

1. Init_rtn
2. Process_rtn
3. End_rtn

The Process_rtn is composed of the following separate but integrally linked modules according to identification number of screen (Figure 2):

0. Logo_Screen module
1. Main_Menu_Screen module
2. Review_Chart_Screen module
3. Lab_Data_Screen module
4. Assess_Dosage_Regimen_Screen module
5. Estimate_Dosage_Regimen_Screen module
6. Autocalculation_Screen module
7. Laboratory_Information_Screen module
8. Dosage_Regimen_Information_Screen module
9. Delete_Patient_Record module

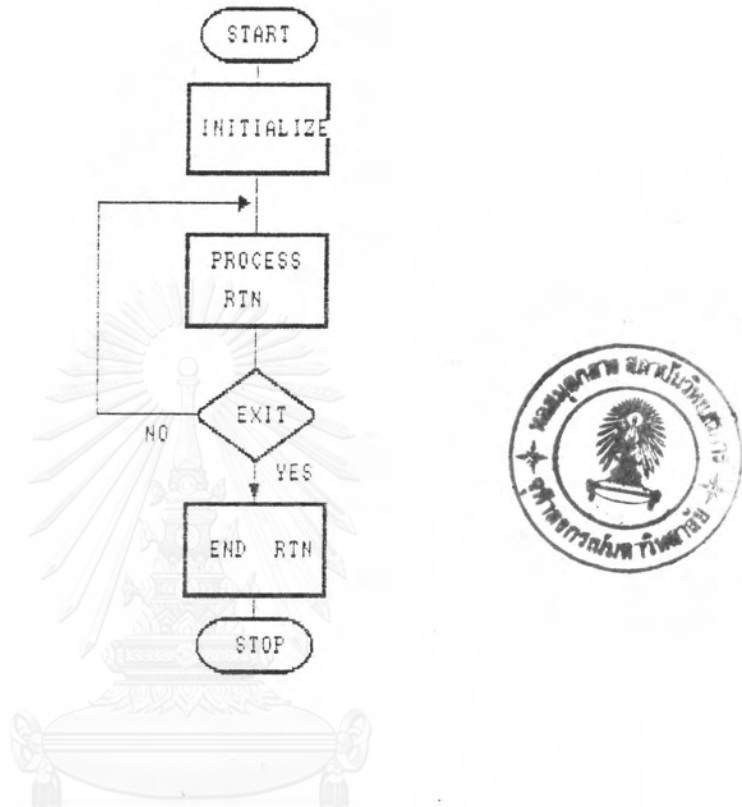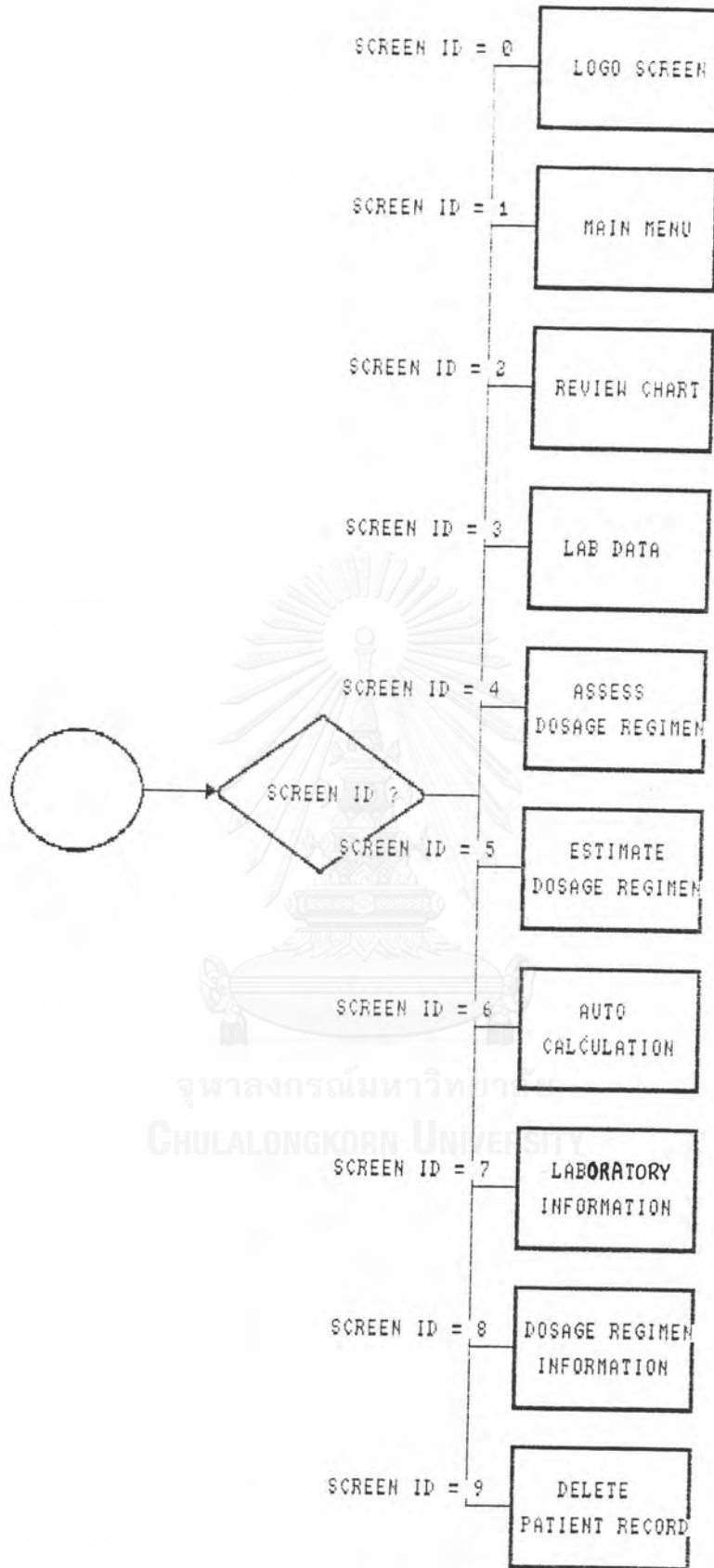Figure 1.   Flowchart for Gentamicin_Dosage_Regimens program

Figure 2. Flowchart for Process_rtn module

An abstract data type to be used in the program is a record of patient data called "PtRec^", a pointer variable, of a record type declared as "PtRectT". This data type when virtually declared is shown in Figure 3. Because the record is quite large, the data file was divided into 2 files; one stores demographic physical, and clinical data fields (as in REVIEW CHART screen), and the other stores laboratory and treatment data fields (as in LABORATORY DATA screen) including the hospital number, "hn", field as a key.

Figure 4 represents an operation flowchart of the program Gentamicin_Dosage_Regimens. In addition to the operation flowchart, the module specifications are described below.

1. __Init_rtn__

This module sets attribute for display monitor being used; initializes or sets initial values for all globol variables, in particular, all fields of "PtRec^"; opens file; and obtains the address of an unused memory location of the appropriate type and assigns it to "PtRec^".

2. __Process_rtn__

2.1 Logo_Screen module

This module writes a LOGO screen (Appendix B) and waits for the user to press a key. Then, by invocation of the Set_Date_Time module, it shows the SET UP DATE & TIME screen which allows the user to input values for correct date and time.

```
{*- Declare Type of Data Used In Program TGEN.PAS & Environment -*}

      Str30  = String[30];
      Str10  = String[10];
      WithTimeRecT  = RECORD
                          value : Real;
                          t     : DateTime;
                      END;
      WithTimePeriodT = RECORD
                          value: Real;
                          starttime, finishtime: DateTime;
                      END;

      DrugLevelT = ARRAY [1..MAX_DRUG_LEVEL] OF WithTimeRecT;
      PtRecT = RECORD
          {*- input data -*}
          {*- Review Chart -*}
                  hn              : Str10;     {*- Hospital Number -*}
                  name            : Str30;     {*- Name & Surname -*}
                  bed             : String[25]; {*- Ward/Bed -*}
                  doctor          : Str30;     {*- Doctor name -*}
                  weight          : real;
                  height          : real;
                  age             : real;
                  sex             : char;
                  prv_ag          : char;
                  scr_over_2      : char;
                  diff_scr_over   : char;
                  diabetic        : char;
                  ab_liver        : char;
                  shock           : char;
                  vanco_ampho     : char;
                  hydration       : char;
                  renal_func      : char;
                  objective1      : string[60];
                  objective2      : string[60];

          {*- Lab & Treatment Data -*}
                  genta_monitor_seq : byte;
                  previousDose    : WithTimePeriodT;
                  testdose        : WithTimePeriodT;
                  predoseC        : WithTimeRecT;
                  Cp              : DrugLevelT;

                  renal_monitor_seq : byte;
                  scr             : longint; {*- Serum creatinine -*}
                  scr_hr          : byte;
                  scr_min         : byte;
                  scr_dd          : byte;
                  scr_mm          : byte;
                  scr_yy          : byte;
```

Figure 3. Data declaration in Gentamicin_Dosage_Regimens program

```
        ucr                  : longint;
        v24                  : longint;    {*- 24 hour urine volume -*}
        v24_fhr              : byte;
        v24_fmin             : byte;
        v24_fdd              : byte;
        v24_fmm              : byte;
        v24_fyy              : byte;
        v24_thr              : byte;
        v24_tmin             : byte;
        v24_tdd              : byte;
        v24_tmm              : byte;
        v24_tyy              : byte;

        treatment_seq        : byte;
        ld                   : longint;
        ld_hr                : byte;
        ld_min               : byte;
        ld_dd                : byte;
        ld_mm                : byte;
        ld_yy                : byte;
        md                   : longint;
        md_hr                : byte;
        md_min               : byte;
        md_dd                : byte;
        md_mm                : byte;
        md_yy                : byte;
        inft_hr              : integer;
        di_hr                : byte;

{*- output data -*}
{*- Common  Output -*}
        new_pat              : boolean;
        new_pat_save         : boolean;
        num                  : byte;
        LBW                  : Real;   {*- Lean Body Weight -*}
        DW                   : Real;   {*- Dosing Weight -*}
        IBM                  : Real;   {*- Ideal Body Mass -*}
        ABM                  : Real;   {*- Adjusted Body Mass -*}
        Clcr                 : Real;   {*- Creatinine clearance -*}
        kel                  : Real;   {*- Elimination rate constant -*}
        halflife             : Real;   {*- Half-life -*}
        Vd                   : Real;   {*- Volume of distribution -*}
        TBC                  : Real;   {*- Total Body Clearance -*}
        a_prim               : Real;   {*- intercept or Cmax -*}
        rsqr                 : Real;   {*- R-squared -*}
{*- Evaluate  -*}
        eva_pcmax            : Real;   {*- eva steady-state peak -*}
        eva_pcmin            : Real;   {*- eva steady-state trough -*}
        eva_Comment_id       : byte;
```

Figure 3.   Data declaration in Gentamicin_Dosage_Regimens program

(Continued...)

```
{*- Prediction Input -*}
      prd_MD                 : Real;   {*- Maintenance Dose -*}
      prd_q                  : Byte;   {*- Dosing Interval -*}
      prd_InfT               : Real;   {*- Infusion Time -*}
{*- Prediction Output -*}
      prd_rsqr               : Real;   {*- R-squared -*}
      prd_a_prim             : Real;   {*- A-prime or intercept -*}
      prd_pcmax              : Real;   {*- Predicted steady-state peak -*}
      prd_pcmin              : Real;   {*- Predicted steady-state trough -*}
{*- Estimation Input -*}
      est_dCmax              : Real;   {*- Desired steady-state peak -*}
      est_dCmin              : Real;   {*- Desired steady-state trough -*}
      est_dInfT              : Real;   {*- Desired infusion time -*}
{*- Estimation Output -*}
      est_calq               : Real;   {*- Calculated Dosing Interval -*}
      est_cal_md             : Real;   {*- Calculated Maintenance Dose -*}
      est_cal_ld             : Real;   {*- Calculated Loading Dose -*}
      est_rMD                : Real;   {*- Recommended Maintenance Dose -*}
      est_rLD                : Real;   {*- Recommended Loading Dose -*}
      est_rq                 : Byte;   {*- Recommended Dosing Interval -*}
      est_pcmax              : Real;   {*- Predicted steady-state peak -*}
      est_pcmin              : Real;   {*- Predicted steady-state trough -*}
{*- Recommend  -*}
      recm_rMD               : Real;   {*- Recommended Maintenance Dose -*}
      recm_rMD_amt           : byte;   {*- Recommended Maintenance Dose amt -*}
      recm_rMD_vol           : Real;   {*- Recommended Maintenance Dose vol -*}
      recm_rLD               : Real;   {*- Recommended Loading Dose -*}
      recm_rLD_amt           : byte;   {*- Recommended Loading Dose amt -*}
      recm_rLD_vol           : Real;   {*- Recommended Loading Dose vol -*}
      recm_q                 : Byte;   {*- Recommended Dosing Interval -*}
      recm_rinf              : Real;   {*- Recommended Infusion time  -*}
      recm_pcmax             : Real;   {*- Predicted steady-state peak -*}
      recm_pcmin             : Real;   {*- Predicted steady-state trough -*}
      Comment_id             : byte;
{*- Blood Sampling
      genta_msg1             : string;
      genta_msg2             : string;
      genta_msg3             : string;
      genta_msg3_2           : string;
      genta_msg4             : string; -*}
    END;

PtRecT_ptr =  ^PtRecT;
```

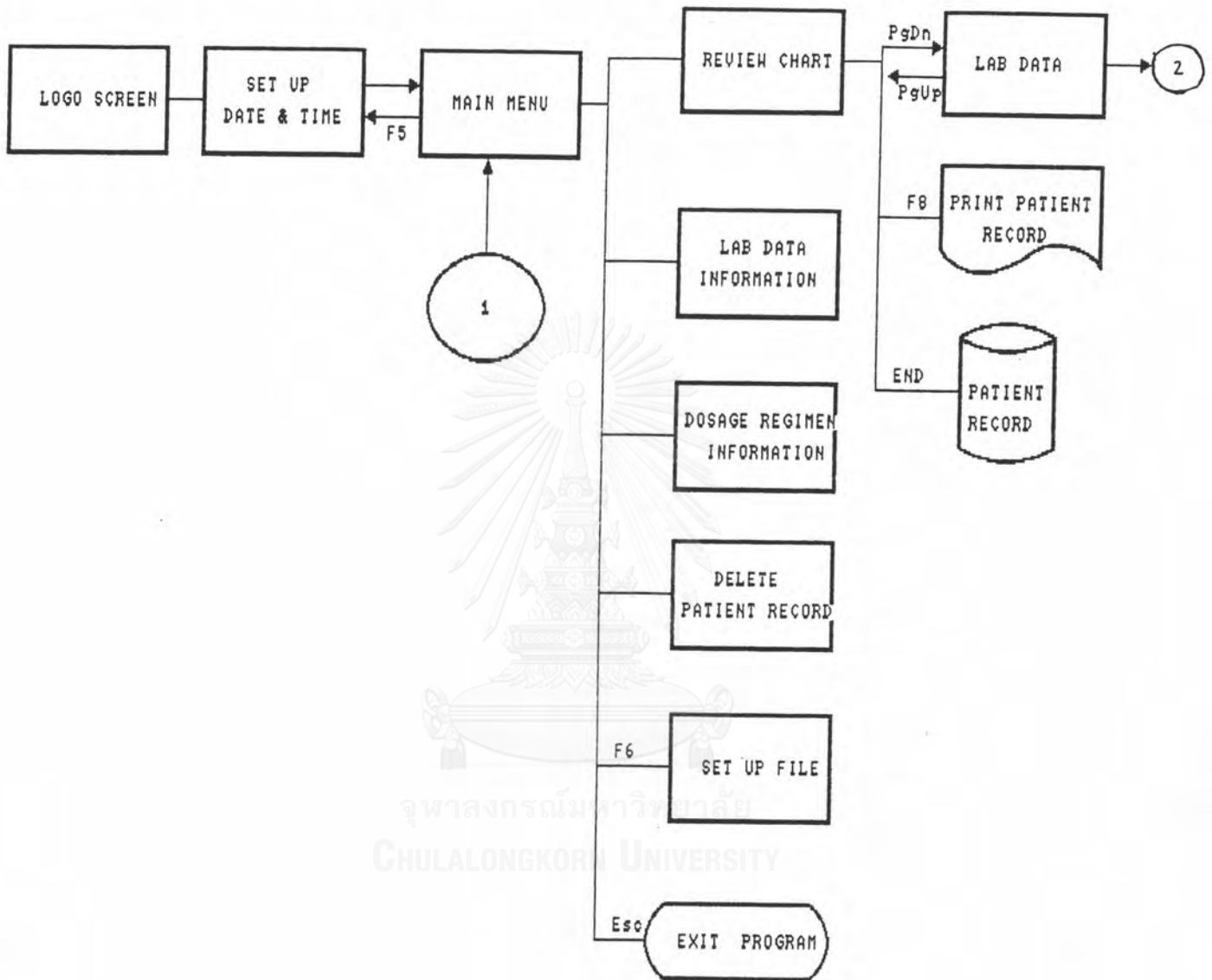Figure 3.   Data declaration in Gentamicin_Dosage_Regimens program

(Continued...)

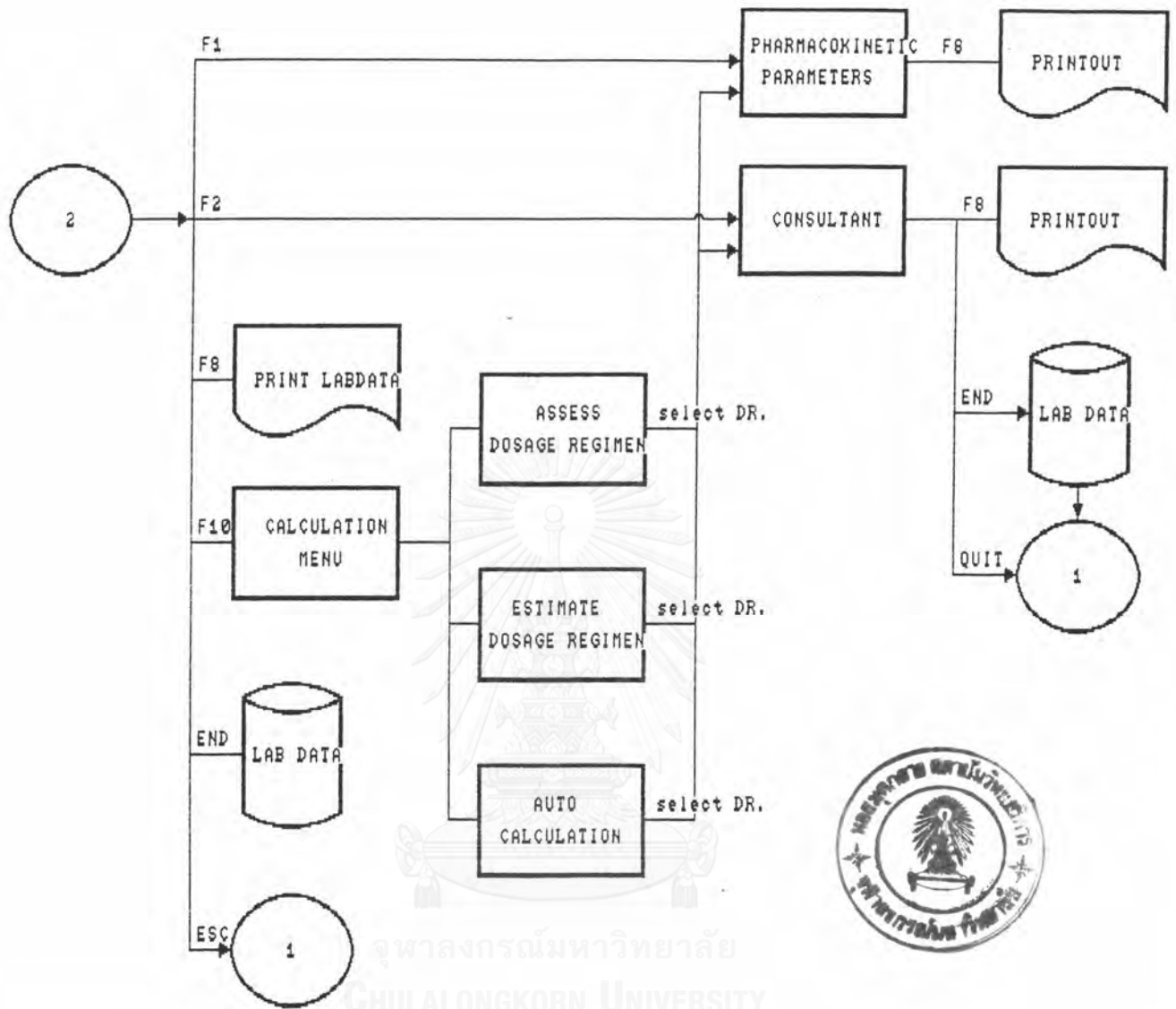Figure 4. Flowchart representing the operation of Gentamicin_Dosage
_Regimens program

Figure 4. Flowchart representing the operation of Gentamicin_Dosage
_Regimens program (Continued...)

2.2 Set_Date_Time module

This module opens the window to write a SET UP DATE & TIME screen, shows the current date and time set in the operating system, and then requests the user to input new values for date and time or press "Enter" key to accept data as shown. Each input is validated for legality and plausibility as it is entered. The module sets the current date and time in the operating system according to the valid input and then exit to the calling program.

2.3 Main_Menu_Screen module

This module provides the user a menu to select what to do next which could be done when user either enters a menu item or uses the arrow keys to move to the highlighted item and presses "Enter". As a result, the module calls another module according to the selected menu item:

"1" calls Review_Chart_Screen module,

"2" calls Laboratory_Information_Screen module,

"3" calls Dosage_Regimen_Information_Screen module,

"4" calls Delete_Patient_Record module.

Apart from the menu items, the module also performs the task according to the function key pressed:

"F5" invokes Set_Date_Time module,

"F6" invokes Set_Up_File module, and

"Esc" invokes Exit_Program module.

2.4 Set_Up_File module

This module requests user to input value for a file name. After validation and acceptance of the legal value, the module will determine whether this file is already existed in the disk or not. If the file is found, the module will open the file to be used; otherwise, it will create a new file with the input name assigned to.

2.5 Exit_Program module

This module questions the user to enter Y or N to indicate whether he or she wants to exit the program or not. If the user enters Y the module will invoke End_rtn module; otherwise, it will exit to the calling program.

2.6 Review_Chart_Screen module

This module writes the REVIEW CHART screen and asks the user to input value for, "hn", a unique hospital number of the patient. It then validates that this value is syntactically valid. If input value is not valid, the module asks the user to reenter it again. The module will continue asking for input and validating it until the user finally enters the value correctly. Then the module accepts the legal value for "hn" and determine whether it is, as the key or index, already existed in the currently opened file or not.

If "hn" is found, the module will read the file and give this patient record to be shown. On the other hand, if "hn" is not found the module continues asking the user to enter another input value for all demographic data, physical data, and clinical

data as indicated in the problem specification document. All input are validated for legality and also for plausibility in the same manner as "hn". The module will write or echo print the input value as it is accepted. Afterwards, the module returns to the calling program legal values for these data fields.

Alternatively, for calling from another module, i.e., the Lab_Data_Screen module, the module is given a "hn" and then recalls all accepted data to show to screen.

From within this screen, the module performs the task according to the function key pressed:

"End" saves the data currently accepted in the screen,

"F8" prints the data currently accepted in the screen,

"PgDn" invokes Lab_Data_Screen module, and

"Esc" invokes Main_Menu_Screen module, except when in the hospital number field all input is clear and the input process begins again.

2.7 Lab_Data_Screen module

This module writes the LABORATORY DATA screen and requests the user to input values for the remaining input data on patient chart, laboratory and treatment data, as stated in the problem specification document. Similar to the Review_Chart_Screen module, all input fields are validated that they are legal and fall within the plausible ranges once each of them is entered. Then the module returns to the calling program legal values for these data.

From within this screen, the module performs the task according to the function key pressed:

"End" saves the data currently accepted in the screen,

"F8" prints the data currently accepted in the screen,

"PgUp" invokes Review_Chart_Screen module,

"Esc" invokes Main_Menu_Screen module,

"F1" invokes PharmacokineticParametersScreen module,

"F2" invokes AssessDosageRegimen module by passing the current dosage regimen as input and then gets the results to invoke the Consultant_Screen module, and

"F10" invokes CalculationMenu module.

2.8 PharmacokineticParametersScreen module

This module is given as input all pharmacokinetic parameter estimates by calling CalKineticParameters module. Then, it writes to the screen all input as output. From within this screen, user can press "F8" to obtain printout.

2.9 CalKineticParameters module

This module is given, as input, legal values for patient data. It then computes and returns all pharmacokinetic parameters as written in the problem specification document to the calling program.

2.10 AssessDosageRegimen module

This module is given both legal values for a given dosage regimen and pharmacokinetic parameter estimates as input. The

dosage regimen input consists of maintenance dose, dosing interval, and infusion time while the pharmacokinetic parameters used are k_e, and V_d estimates. This module computes and returns as output to the calling program the predicted steady-state peak and trough levels yielded from the dosage regimen input along with the comment about these values.

2.11 Consultant_Screen module

This module is given as input a dosage regimen consists of maintenance dose and dosing interval, as well as infusion time, the corresponding predicted steady-state peak and trough levels, and comment. Then, it writes to the screen all input as output and also provides the dose as the number of milliliters to be drawn from the common dosage form. The second part of the screen displays the recommended blood sampling times for both creatinine and gentamicin levels which are given by calling the Consult module.

From within this screen, the module performs the task according to the function key pressed:

"End" saves the data currently accepted in the screen,

"F8" prints the data currently accepted in the screen,

"Esc" invokes Main_Menu_Screen module.

2.12 Consult module

This module is given patient input data fields for age, sex, S_{cr} value, renal status and risk factors to concern about appropriate blood sampling times for both creatinine and gentamicin levels. It returns these messages as output.

2.13 CalculationMenu module

This module provides the user a menu to select which way to determine a desired dosage regimen. This could be done as user either enters a menu item or uses the arrow keys to move to the highlighted item and presses "Enter". As a result, the module calls another module according to the selected menu item:

"1" calls Assess_Dosage_Regimen_Screen module,

"2" calls Estimate_Dosage_Regimen_Screen module, and

"3" calls Autocalculation_Screen module.

User can press "Esc" to exit this module.

2.14 Assess_Dosage_Regimen_Screen module

This module writes the ASSESS A DOSAGE REGIMEN screen and requests the user to input values for a propose dosage regimen consists of maintenance dose, dosing interval, and infusion time. All input must be validated before acceptance. The module then passes these values into the calculating module, **AssessDosageRegimen module**, and the result will be returned as output to the module. The result consists of predicted steady-state peak and trough levels yielded from the dosage regimen input along with the comment about these values is shown on the screen and the module allows for entry of another input. Process will be repeated up to 30 times or for 30 dosage regimens. The module asks the user to select a dosage regimen by pressing a number. This selected dosage regimen is passed to the Consultant module immediately by invocation of that module.

2.15 Estimate_Dosage_Regimen_Screen module

This module writes ESTIMATE A DOSAGE REGIMEN screen and requests the user to input values for the desired steady-state peak and trough levels and desired infusion time. Input validation must be done before acceptance of the value. Afterwards, by calling the EstimateDosageRegimen module, passing these values as input, the results are returned as output to be shown on the screen. These values include a practical dosage regimen consists of maintenance dose and dosing interval, the corresponding predicted steady-state peak and trough levels and comment. The module then allows for entry of another input. Process will be repeated upto 30 times or for 30 dosage regimens. The module asks the user to select a dosage regimen by pressing a number. This selected dosage regimen is passed to the Consultant module immediately by invocation of that module.

2.16 EstimateDosageRegimen module

This module is given both legal values for the desired steady-state peak and trough levels as well as the desired infusion time and pharmacokinetic parameter estimates, k_{e_1} and V_d as input. It calculates an appropriate dosage regimen consisted of maintenance dose and dosing interval and rounds up these values to practical values. The recommended dosage regimen is then assessed by calling the AssessDosageRegimen module. Afterwards, the values for a practical dosage regimen including the corresponding predicted steady-state peak and trough levels and comment are returned to the calling program.

2.17 Autocalculation_Screen module

This module writes AUTO-CALCULATION screen and asks the user to input values for the desired range of steady-state peak and trough levels, i.e., minimum peak, maximum peak, minimum trough, and maximum trough; as well as 4 values for the desired infusion time and 6 values for the desired dosing interval. Input validation must be done before acceptance of the value. Afterwards, by calling the Autocalculation module, passing these values as input, the results are returned as output to be shown on the screen. These values include 30 (or less) practical dosage regimens; each consists of maintenance dose and dosing interval, the corresponding predicted steady-state peak and trough levels and comment. The module then asks the user to select a dosage regimen by pressing a number. This selected dosage regimen is passed to the Consultant module immediately by invocation of that module.

2.18 Autocalculation module

This module is given, as input, legal values for the desired range of steady-state peak and trough levels, i.e., minimum peak, maximum peak, minimum trough, and maximum trough; as well as the desired infusion time varies in 4 values and the desired dosing interval varies in 6 values. Also, the module is given the pharmacokinetic parameter estimates, k_{e_1} and V_d as input. Both input are used to generate up to 30 dosage regimens which meet the requirements or the ranges specified. The first step is to estimate a minimum dose by calling the EstimateDosageRegimen module and passing the values for mininmum peak, minimum trough, minimum

infusion time, and minimum dosing interval. This dose is then used to begin the calculation process which will be repeated by switching to next value of infusion time and dosing interval as well as increasing dose until 30 dosage regimens are gained or the criteria are met. Between repeat calculation, each dosage regimen is assessed by calling the AssessDosageRegimen module. Finally, the values for 30 dosage regimens (or less) including the corresponding predicted steady-state peak and trough levels together with the comment are returned to the calling program.

2.19 Lab_Data_Information_Screen module

This module writes the LABORATORY INFORMATION screen and asks the user to input value for "hn". Then, the input is validated for legality and plausibility. When the module accepts the legal value for "hn", it will determine whether the value is already existed in the currently opened file or not. If "hn" is found, the module will read the file and give the patient records to be shown. These data include all loboratory data entry at different times: renal monitors consist of S_{cr} and the date on which it was measured, U_{cr}, V_{24}; and gentamicin level monitors consist of previous dose, predose level, test dose and the corresponding postdose levels.

2.20 Dosage_Regimen_Information_Screen module

This module writes the DOSAGE REGIMEN INFORMATION screen and asks the user to input value for "hn". After validation and acceptance of the legal value for "hn", it will determine whether the value is already existed in the currently opened file or not. If

"hn" is found, the module will read the file and give the patient records to be shown. These data include all treatment data entry at different times. Each record consists of the maintenance dose, dosing interval, infusion time, and the date on which this dosage regimen was started. In addition to the treatment data, the module displays the corresponding predicted steady-state peak and trough levels together with the pharmacokinetic parameters, $k_{e l}$ and V_d.

2.21 Delete_Patient_Record module

This module requests user to input value for "hn". After validation and acceptance of the legal value for "hn", it will determine whether the value is already existed in the currently opened file or not. If "hn" is found, the module will delete this record; otherwise, it will provide an appropriate message.

3. End_rtn

This module returns a memory location to the pool of locations available for other use, closes file, clears screen, and writes "Good-bye" message before return to the disk operating system.