Work Section 1

1.

```
/****************************** REXX ***********************************/
/* Program                                                             */
/* rx10111                                                             */
/* Description                                                         */
/* REXX program to display a list of names                             */
/* Author       : Michaelangelo DeParma                                */
/* Date         : 1st January 2000                                     */
/*----------------------- Amendment History -------------------------*/
/*********************************************************************/
SAY "Bob Flemming"
SAY "Gary Stinker"
SAY "John Thomas"
```

2.

```
/****************************** REXX ***********************************/
/* Program                                                             */
/* rx10112                                                             */
/* Description                                                         */
/* REXX program to welcome a name to the screen                        */
/* Author       : Michaelangelo DeParma                                */
/* Date         : 1st January 2000                                     */
/*----------------------- Amendment History -------------------------*/
/*********************************************************************/
SAY "Please enter your name :"
PULL your_name
SAY
SAY "Hello "your_name
SAY "Welcome to the course."
```

Additional Program

```
/****************************** REXX ***********************************/
/* Program                                                             */
/* rx10113                                                             */
/* Description                                                         */
/* REXX program to welcome a name to the screen                        */
/* Author       : Michaelangelo DeParma                                */
/* Date         : 1st January 2000                                     */
/*----------------------- Amendment History -------------------------*/
/*********************************************************************/
SAY "Please enter your forename : "
PULL fore_name
SAY
SAY "Please enter your surname : "
PULL sur_name
SAY
SAY "Welcome to the course - "fore_name sur_name
```

Addition Program 2

```
/****************************** REXX ***********************************/
/* Program                                                             */
/* rx10114                                                             */
/* Description                                                         */
/* REXX program to welcome a name to the screen                        */
/* Author       : Michaelangelo DeParma                                */
/* Date         : 1st January 2000                                     */
/*----------------------- Amendment History -------------------------*/
/*********************************************************************/
CLEAR
SAY "Please enter your Forename and Surname :"
PULL fore_name sur_name
SAY
SAY "Welcome to the course - "sur_name fore_name
```

Test21

```
/****************************** REXX ***********************************/
/* Program                                                             */
/* test21                                                              */
/* Description                                                         */
/* REXX program to test statements                                     */
/* Author       : Michaelangelo DeParma                                */
/* Date         : 1st January 2000                                     */
/*----------------------- Amendment History -------------------------*/
/***********************************************************************/
CLEAR
/* REXX EXEC */
a = "REXX EXEC"
c = 94 4F
d = '94 4F'
e = 32
f = '32'
g = "32"x
b = "REXX EXEC"x
REXX EXEC
```

Test22

```
/****************************** REXX ***********************************/
/* Program                                                             */
/* test22                                                              */
/* Description                                                         */
/* REXX program to test statements                                     */
/* Author       : Michaelangelo DeParma                                */
/* Date         : 1st January 2000                                     */
/*----------------------- Amendment History --------------------------*/
/***********************************************************************/
CLEAR
Answer = "yes"
The-Answer = "no"
Msg.Text = "IST510I"
1ST_Class = "never"
the_worlds_largest_variable = "test"
REPLLY? = "Z NET,QUICK"
SAY reply
```

Test23

```
14
20
8.1
4
***
```

```
/***************************** REXX ***********************************/
/* Program     : test23                                                */
/* Description : sanswer to TEST23                                     */
/* Author      : Mike Parma                                            */
/* Date        : 25th November 1999                                    */
/* Version     : 1.0                                                   */
/*---------------------- Amendment History -------------------------*/
/* Date        ¦ Author             ¦ Description                     */
/*-------------------------------------------------------------------*/
/* dd/mm/yyyy  ¦                    ¦                                 */
/*********************************************************************/
"clear"
SAY 2+3*4
SAY (2+3)*4
SAY 9**2/10
SAY 23%5//2**3
```

Test24

```
REXXCourse
REXXCourse
who knowsthe answer
First Second
who knowsANSWER
```

```
/****************************** REXX ***********************************/
/* Program     : test24                                                 */
/* Description : answer to TEST24                                       */
/* Author      : Mike Parma                                             */
/* Date        : 25th November 1999                                     */
/* Version     : 1.0                                                    */
/*---------------------- Amendment History -------------------------*/
/* Date        ¦ Author             ¦ Description                       */
/*-----------------------------------------------------------------------*/
/* dd/mm/yyyy  ¦                    ¦                                   */
/*************************************************************************/
"clear"
SAY "REXX"       ||        "Course"
SAY "REXX"||"Course"
variable = "who knows"
SAY variable"the answer"
SAY "First"          "Second"
SAY variable||""||answer
```

Work Section 2

1.

```
/****************************** REXX ***********************************/
/* Program                                                             */
/* rx10121                                                             */
/* Description                                                         */
/* REXX program to check concatenation.                                */
/* Author       : Michaelangelo DeParma                                */
/* Date         : 1st January 2000                                     */
/*---------------------- Amendment History -------------------------*/
/*********************************************************************/
CLEAR
SAY "Please enter your first name."
PARSE PULL first_name
SAY "Please enter your last name."
PARSE PULL last_name
SAY first_name last_name
SAY last_name"," first_name
SAY last_name||first_name
SAY first_name||last_name
```

2.

```
/****************************** REXX ***********************************/
/* Program                                                             */
/* rx10122                                                             */
/* Description                                                         */
/* REXX program to calculate TAX.                                      */
/* Author       : Michaelangelo DeParma                                */
/* Date         : 1st January 2000                                     */
/*---------------------- Amendment History -------------------------*/
/*********************************************************************/
CLEAR
SAY "Please enter your first name."
PARSE PULL first_name
SAY "Please enter your salary."
PARSE PULL salary
tax = .3 * salary
SAY "Your tax paid is :"tax
```

Work Section 3

1.

```
/****************************** REXX ***********************************/
/* Program                                                             */
/* rx10131                                                             */
/* Description                                                         */
/* REXX program to accept an argument.                                 */
/* Author       : Michaelangelo DeParma                                */
/* Date         : 1st January 2000                                     */
/*---------------------- Amendment History -------------------------*/
/*********************************************************************/
CLEAR
PARSE ARG name
SAY "Hello "name
```

2.

```
/****************************** REXX ***********************************/
/* Program                                                             */
/* rx10132                                                             */
/* Description                                                         */
/* REXX program to accept an argument and split the dataset name.     */
/* Author       : Michaelangelo DeParma                                */
/* Date         : 1st January 2000                                     */
/*---------------------- Amendment History -------------------------*/
/*********************************************************************/
CLEAR
PARSE ARG data_set
PARSE VAR data_set project "." group "." type
SAY "Project : "project
SAY "Group   : "group
SAY "Type    : "type
```

Additional Program

```
/****************************** REXX ***********************************/
/* Program                                                             */
/* rx10133                                                             */
/* Description                                                         */
/* REXX program to welcome a name to the screen                        */
/* Author       : Michaelangelo DeParma                                */
/* Date         : 1st January 2000                                     */
/*----------------------- Amendment History --------------------------*/
/***********************************************************************/
CLEAR
SAY "Please enter your name :"
PULL name
PARSE VAR name your_name .
SAY
SAY "Hello "your_name
SAY "Welcome to the course."
```

Test 41

```
test_value = 1
IF test_value = 1 THEN
   SAY "Yes"
IF test_value = 1 THEN
   SAY "Yes"
IF test_value = 1 THEN
   SAY "Yes"
ELSE
   SAY "No"
IF test_value = 1 THEN
   SAY "Yes"
ELSE
   SAY "No"
IF test_value = 1
   THEN
      SAY "Yes"
   ELSE
      SAY "No"
```

Work Section 4

1.

```
/****************************** REXX ***********************************/
/* Program                                                             */
/* rx10141                                                             */
/* Description                                                         */
/* REXX program to test IF statements                                  */
/* Author       : Michaelangelo DeParma                                */
/* Date         : 1st January 2000                                     */
/*---------------------- Amendment History -------------------------*/
/*********************************************************************/
CLEAR
PARSE ARG age
IF age > 64 THEN
   SAY "65 or over"
IF age < 65 & age > 21 THEN
   SAY "Over 21 and under 65"
IF age >= 16 & age <= 21 THEN
   SAY "Between 16 and 21"
IF age < 16 THEN
SAY "Under 16"
```

2.

```
/****************************** REXX ***********************************/
/* Program                                                             */
/* rx10142                                                             */
/* Description                                                         */
/* REXX program to test IF statements                                  */
/* Author       : Michaelangelo DeParma                                */
/* Date         : 1st January 2000                                     */
/*---------------------- Amendment History -------------------------*/
/*********************************************************************/
CLEAR
PARSE ARG age
SELECT
   WHEN age < 16 THEN SAY "Under 16"
   WHEN age <=21 THEN SAY "Between 16 and 21"
   WHEN age < 65 THEN SAY "Over 21 and under 65"
   OTHERWISE SAY "65 or over"
END
```

Additional Program

```
/****************************** REXX ***********************************/
/* Program                                                             */
/* rx10143                                                             */
/* Description                                                         */
/* REXX program to test IF statements                                  */
/* Author       : Michaelangelo DeParma                                */
/* Date         : 1st January 2000                                     */
/*---------------------- Amendment History --------------------------*/
/*********************************************************************/
CLEAR
SAY "Please enter your salary."
PARSE UPPER PULL salary un_used
SAY "Please enter your TAX band."
PARSE UPPER PULL tax un_used
SELECT
   WHEN tax = 10 THEN DO
      tax_paid = salary * 0.1
   END
   WHEN tax = 20 THEN DO
      tax_paid = salary * 0.2
   END
   WHEN tax = 50 THEN DO
      tax_paid = salary * 0.5
   END
   WHEN tax = 70 THEN DO
      tax_paid = salary * 0.7
   END
   WHEN tax = 80 THEN DO
      tax_paid = salary * 0.8
   END
   OTHERWISE DO
      SAY "Incorrect tax code : "tax
      EXIT
   END
END
SAY "Your tax for :" salary ": is :" tax_paid
```

Work Section 5

1.

```
/****************************** REXX ***********************************/
/* Program                                                             */
/* rx10151                                                             */
/* Description                                                         */
/* REXX program to welcome a name to the screen and test TRACE         */
/* Author       : Michaelangelo DeParma                                */
/* Date         : 2nd February 2000                                    */
/*----------------------- Amendment History -------------------------*/
/*********************************************************************/
CLEAR
TRACE A
SAY "Please enter your name :"
PULL your_name
SAY
SAY "Hello "your_name
SAY "Welcome to the course."
```

2.

```
/****************************** REXX ***********************************/
/* Program                                                             */
/* rx10152                                                             */
/* Description                                                         */
/* REXX program to welcome a name to the screen and test TRACE         */
/* Author       : Michaelangelo DeParma                                */
/* Date         : 2nd February 2000                                    */
/*----------------------- Amendment History -------------------------*/
/*********************************************************************/
CLEAR
TRACE I?
day = "Tuesday"
month.day = "May"
tuesday = "Tuesday"
SAY month.Tuesday
```

Additional Program

```
/****************************** REXX ***********************************/
/* Program                                                             */
/* rx10153                                                             */
/* Description                                                         */
/* REXX program to welcome a name to the screen and test TRACE         */
/* Author       : Michaelangelo DeParma                                */
/* Date         : 2nd February 2000                                    */
/*----------------------- Amendment History -------------------------*/
/*********************************************************************/
CLEAR
TRACE(I)
Day = "Tuesday"
Month.day = "May"
Tuesday = "Tuesday"
SAY month.Tuesday
```

Additional Program 2

```
/****************************** REXX ***********************************/
/* Program     : WK54                                                  */
/* Description : This program will test the SIGNAL ON NOVALUE          */
/* Author      : Neville Croker                                        */
/* Date        : 13th December 2000.                                  */
/* Version     : 1.0                                                   */
/*----------------------- Amendment History -------------------------*/
/* Date        ¦ Author             ¦ Description                      */
/*-------------------------------------------------------------------*/
/* dd/mm/yyyy  ¦                    ¦                                  */
/*********************************************************************/
SIGNAL ON NOVALUE
SAY "Your name is "name
EXIT

NOVALUE:
   SAY "================================="
   SAY "You have used an uninitialised"
   SAY "Variable : "||CONDITION("D")
   SAY "in line : "||SIGL
   SAY "================================="
   RETURN
```

Test 61

```
/****************************** REXX ***********************************/
/* Program                                                             */
/* test61                                                              */
/* Description                                                         */
/* REXX program to test do statements                                  */
/* Author       : Michaelangelo DeParma                                */
/* Date         : 2nd February 2000                                    */
/*----------------------- Amendment History -------------------------*/
/***********************************************************************/
CLEAR
Loop_counter = 0
DO 5
   loop_counter = loop_counter + 1
   even_result = loop_counter // 2
   IF even_result = 0 THEN DO
      SAY "Count is : "loop_counter
   END
END
```

Test 62

```
/****************************** REXX ***********************************/
/* Program                                                             */
/* test62                                                              */
/* Description                                                         */
/* REXX program to test DO statements                                  */
/* Author       : Michaelangelo DeParma                                */
/* Date         : 2nd February 2000                                    */
/*----------------------- Amendment History -------------------------*/
/***********************************************************************/
CLEAR
DO loop_counter = 1 TO 10 BY 3
   SAY loop_counter
END
```

Work Section 6

1.

```
/****************************** REXX ***********************************/
/* Program                                                             */
/* rx10161                                                             */
/* Description                                                         */
/* REXX program to test looping.                                       */
/* Author       : Michaelangelo DeParma                                */
/* Date         : 2nd February 2000                                    */
/*---------------------- Amendment History -------------------------*/
/***********************************************************************/
CLEAR
Total = 0
DO loop_counter = 1 TO 5
   SAY "Please enter a number:"
   PARSE PULL number.loop_counter
   total = total + number.loop_counter
END
DO new_counter = 1 TO 5
   SAY number.new_counter
END
Average = total / 5
SAY "Average = "average
```

1.a.

```
/****************************** REXX ***********************************/
/* Program                                                             */
/* rx10161a                                                            */
/* Description                                                         */
/* REXX program to test looping.                                       */
/* Author       : Michaelangelo DeParma                                */
/* Date         : 2nd February 2000                                    */
/*---------------------- Amendment History --------------------------*/
/***********************************************************************/
CLEAR
Total = 0
DO loop_counter = 1 TO 5
   SAY "Please enter a number:"
   PARSE PULL number.loop_counter
   IF highest < number.loop_counter THEN DO
      highest = number.loop_counter
   END
   total = total + number.loop_counter
END
DO new_counter = 1 TO 5
   SAY number.new_counter
END
Average = total / 5
SAY "Average = "average
SAY "Highest = "highest
```

2.

```
/****************************** REXX ***********************************/
/* Program                                                             */
/* rx10162                                                             */
/* Description                                                         */
/* REXX program to loop round work section 2.1                         */
/* Author       : Michaelangelo DeParma                                */
/* Date         : 2nd February 2000                                    */
/*---------------------- Amendment History -------------------------*/
/***********************************************************************/
CLEAR
DO FOREVER
   SAY "Please enter your first name."
   PARSE PULL first_name
   IF first_name = "" THEN DO
      EXIT
   END
   SAY "Please enter your last name."
   PARSE PULL last_name
   SAY first_name last_name
   SAY last_name"," first_name
   SAY last_name||first_name
   SAY first_name||last_name
END
```

Work Section 7

1.

```
/***************************** REXX ***********************************/
/* Program                                                            */
/* rx10171                                                            */
/* Description                                                        */
/* REXX program to test the use of functions.                         */
/* Author       : Michaelangelo DeParma                               */
/* Date         : 3rd February 2000                                   */
/*---------------------- Amendment History -------------------------*/
/**********************************************************************/
CLEAR
Program_title = CENTRE("Function Program", 79)
Under_line = CENTRE("==================", 79)
SAY program_title
SAY under_line
SAY
SAY "The american formatted date is : "DATE("U")
SAY "This program was executed at : "TIME()
SAY "The european date : "TRANSLATE(DATE("E"), "-", "/")
```

2.

```
/****************************** REXX ***********************************/
/* Program                                                             */
/* rx10172                                                             */
/* Description                                                         */
/* REXX program to test the use of functions.                          */
/* Author       : Michaelangelo DeParma                                */
/* Date         : 3rd February 2000                                    */
/*----------------------- Amendment History --------------------------*/
/*********************************************************************/
CLEAR
DO FOREVER
   SAY "Please enter your name :"
   PARSE UPPER PULL name
   PARSE VAR name full_name .
   IF ABBREV("MIKE DEPARMA", full_name, 4) = 0 THEN DO
      ITERATE
   END
   ELSE DO
      LEAVE
   END
END
SAY "Please enter four numbers"
PARSE PULL one_num
PARSE PULL two_num
PARSE PULL three_num
PARSE PULL four_num
SAY "The highest number is : "MAX(one_num, two_num, three_num,
four_num)
SAY "The lowest number is : "MIN(one_num, two_num, three_num, four_num)
```

Additional Program.

```
/****************************** REXX ***********************************/
/* Program                                                             */
/* rx10172a                                                            */
/* Description                                                         */
/* REXX program to test the use of functions.                          */
/* Author       : Michaelangelo DeParma                                */
/* Date         : 3rd February 2000                                    */
/*---------------------- Amendment History -------------------------*/
/*********************************************************************/
CLEAR
Game_number = RANDOM(1, 100)
go_counter = 0
program_title = CENTRE("Number game", 79)
under_line = CENTRE("=============", 79)
SAY program_title
SAY under_line
SAY
DO FOREVER
   SAY "Please Guess the number (1-100) :"
   PARSE PULL player_guess
   go_counter = go_counter + 1
   IF player_guess = game_number THEN DO
      LEAVE
   END
   ELSE DO
      IF player_guess > game_number THEN DO
         SAY "Too high"
      END
      ELSE DO
         SAY "Too low"
      END
   END
END
SAY
SAY "Hurrah you guessed the number in "go_counter" guesses."
```

Work Section 8

1.

```
/****************************** REXX ***********************************/
/* Program                                                             */
/* rx10181                                                             */
/* Description                                                         */
/* REXX program to test the use of site written routines               */
/* Author       : Michaelangelo DeParma                                */
/* Date         : 3rd February 2000                                    */
/*----------------------- Amendment History -------------------------*/
/***********************************************************************/
CLEAR
CALL addup 1 2 3 4 5 6 7 8
EXIT

addup:
   ARG numbers
   number_count = WORDS(numbers)
   total = 0
   DO loop_counter = 1 TO number_count
      total = total + WORD(numbers, loop_counter)
   END
   SAY "The total of : "numbers
   SAY "is : "total
   RETURN
```

2.

```
/****************************** REXX ***********************************/
/* Program                                                             */
/* rx10182                                                             */
/* Description                                                         */
/* REXX program to test the use of site written routines               */
/* Author       : Michaelangelo DeParma                                */
/* Date         : 3rd February 2000                                    */
/*---------------------- Amendment History -------------------------*/
/***********************************************************************/
CLEAR
total = addup(1 2 3 4 5 6 7 8)
SAY "The total is : "total
```

```
/****************************** REXX ***********************************/
/* Program                                                             */
/* addup                                                               */
/* Arguments                                                           */
/* All the numbers to added up.                                        */
/* Description                                                         */
/* REXX program to test the use of site written routines               */
/* Author       : Michaelangelo DeParma                                */
/* Date         : 3rd February 2000                                    */
/*---------------------- Amendment History -------------------------*/
/***********************************************************************/
ARG numbers
number_count = WORDS(numbers)
total = 0
DO loop_counter = 1 TO number_count
  total = total + WORD(numbers, loop_counter)
END
RETURN total
```

Additional Program

```
/****************************** REXX ***********************************/
/* Program                                                             */
/* rx10182a                                                            */
/* Description                                                         */
/* REXX program to test the use of site written routines               */
/* Author       : Michaelangelo DeParma                                */
/* Date         : 3rd February 2000                                    */
/*----------------------- Amendment History -------------------------*/
/*********************************************************************/
CLEAR
low = 1
high = 100
SAY "Please enter the number you wish to check : "
PARSE PULL num
PARSE VAR num number .
number_check = RANGECHK(low high number)
IF number_check = 1 THEN DO
   SAY "The number is in the range "low" to "high
END
ELSE DO
   SAY "The number is out of range "low" to "high
END
```

```
/****************************** REXX ***********************************/
/* Program                                                             */
/* rangechk                                                            */
/* Arguments                                                           */
/* low - smallest number, high the highest number and                  */
/* the number to check.                                                */
/* Description                                                         */
/* REXX program to test the use of site written routines               */
/* Author       : Michaelangelo DeParma                                */
/* Date         : 3rd February 2000                                    */
/*----------------------- Amendment History -------------------------*/
/*********************************************************************/
ARG low high number
IF number >= low & number <= high THEN DO
   check_result = 1
END
ELSE DO
   check_result = 0
END
RETURN check_result
```

Additional Program

```
/****************************** REXX ***********************************/
/* Program                                                             */
/* rx10184                                                             */
/* Description                                                         */
/* REXX program to test the use of REVWORD function                    */
/* Author       : Michaelangelo DeParma                                */
/* Date         : 3rd February 2000                                    */
/*---------------------- Amendment History -------------------------*/
/*********************************************************************/
CLEAR
SAY "Please enter your list of words : "
PARSE PULL list_of_words
list_of_words = STRIP(list_of_words)
SAY "The list is : "||REVWORD(list_of_words)
```

```
/****************************** REXX ***********************************/
/* Program                                                             */
/* revword                                                             */
/* Arguments                                                           */
/* a word list.                                                        */
/* Description                                                         */
/* REXX program to reverse a list of words.                            */
/* Author       : Michaelangelo DeParma                                */
/* Date         : 3rd February 2000                                    */
/*---------------------- Amendment History -------------------------*/
/*********************************************************************/
ARG word_list
new_list = ""
no_of_words = WORDS(word_list)
DO loop_counter = no_of_words TO 1 BY -1
   new_list = new_list||" "||WORD(word_list, loop_counter)
END
RETURN new_list
```

Work Section 9.1

```
/****************************** REXX ***********************************/
/* Program                                                             */
/* rx20111                                                             */
/* Description                                                         */
/* REXX program to display the result of the "LU"                      */
/* Author       : Michaelangelo DeParma                                */
/* Date         : 1st January 2000                                     */
/*----------------------- Amendment History -------------------------*/
/***********************************************************************/
ADDRESS "TSO"
"CLEAR"
SAY "The result of the LU command is :"
"LU"
```

Work Section 9.2

```
/***************************** REXX ***********************************/
/* Program                                                            */
/* rx20112                                                            */
/* Description                                                        */
/* REXX program to send a TSO message.                                */
/* Author      : Michaelangelo DeParma                                */
/* Date        : 1st January 2000                                     */
/*----------------------- Amendment History -------------------------*/
/**********************************************************************/
ADDRESS "TSO"
"CLEAR"
CALL get_message
CALL get_id
CALL send_message
EXIT

get_message:
   SAY "Please enter the message to send:"
   PARSE PULL tso_message
   tso_message = STRIP(tso_message)
   RETURN

Get_id:
   SAY "Please enter the TSO ID :"
   PARSE PULL tso_id
   tso_id = STRIP(tso_id)
   RETURN

Send_message:
   "SE '"tso_message"' USER("tso_id") LOGON"
   RETURN
```

Work Section 9.3

```
/***************************** REXX ***********************************/
/* Program                                                            */
/* rx20113                                                            */
/* Description                                                        */
/* REXX program to display a userid's datasets.                       */
/* Author      : Michaelangelo DeParma                                */
/* Date        : 1st January 2000                                     */
/*----------------------- Amendment History -------------------------*/
/**********************************************************************/
ADDRESS "TSO"
"CLEAR"
CALL get_id
CALL display_dataset
EXIT

get_id:
   SAY "Please enter the TSO ID :"
   PARSE PULL tso_id
   tso_id = STRIP(tso_id)
   RETURN

Display_dataset:
   un_used = OUTTRAP("line.", "*", "NOCONCAT")
      "LISTC LEVEL("tso_id")"
   un_used = OUTTRAP("OFF")
   DO line_counter = 1 TO line.0 BY 2
      SAY WORD(line.line_counter,3)
   END
   RETURN
```

Additional Program

```
/****************************** REXX ***********************************/
/* Program                                                             */
/* rx20114                                                             */
/* Description                                                         */
/* REXX program to display a datasets's volume.                        */
/* Author       : Michaelangelo DeParma                                */
/* Date         : 1st January 2000                                     */
/*---------------------- Amendment History -------------------------*/
/*********************************************************************/
ADDRESS "TSO"
"CLEAR"
CALL get_dataset
CALL display_dataset
EXIT

get_dataset:
   SAY "Please enter the dataset:"
   PARSE PULL dsn_name
   dsn_name = STRIP(dsn_name)
   RETURN

display_dataset:
   un_used = OUTTRAP("line.", "*", "NOCONCAT")
      "LISTDS '"dsn_name"'"
   un_used = OUTTRAP("OFF")
   SAY "The dataset is on volume : "||line.5
   RETURN
```

Work Section 10.1

```
/****************************** REXX ***********************************/
/* Program                                                             */
/* rx20121                                                             */
/* Description                                                         */
/* REXX program to check that a dataset is valid REXX                  */
/* Author      : Michaelangelo DeParma                                 */
/* Date        : 2nd February 2000.                                    */
/*----------------------- Amendment History -------------------------*/
/*********************************************************************/
dsn_name = "crone90.crone.rexx(test)"
CALL check_dsn
CALL read_dsn
CALL check_member
EXIT

check_dsn:
   dsn_check = SYSDSN("'"dsn_name"'")
   IF dsn_check \= "OK" THEN DO
   IF dsn_check \= "OK" THEN DO
      SAY dsn_check
      EXIT
   END
   ELSE DO
      NOP
   END
   RETURN

Read_dsn:
   "ALLOC DD(ddname) DSN('"dsn_name"') SHR REU"
   "EXECIO 1 DISKR ddname (STEM row. FINIS)"
   PARSE UPPER VAR row.1 row.1
   RETURN

Check_member:
   IF POS("/*", row.1) = 0 | POS("*/", row.1) = 0 |,
                             POS("REXX", row.1) = 0 THEN DO
      SAY "This does not contain the correct first line to be REXX."
   END
   ELSE DO
     SAY "This contains the correct elements for a REXX program."
   END
   RETURN
```

Work Section 10.2

```
/***************************** REXX ***********************************/
/* Program                                                            */
/* rx20122                                                            */
/* Description                                                        */
/* REXX program to create a REXX program                              */
/* Author      : Michaelangelo DeParma                                */
/* Date        : 2nd February 2000.                                   */
/*----------------------- Amendment History -------------------------*/
/**********************************************************************/
CALL create_rexx
CALL allocate
CALL write_dsn
EXIT

Create_rexx:
   dsn_name = "crone90.crone.rexx(test)"
   line.1 = "/* REXX */"
   line.2 = "SAY 'Enter your name'"
   line.3 = "PARSE PULL name"
   line.4 = "SAY 'Hello '||name"
   RETURN

Allocate:
   "ALLOC DD(ddname) DSN('"dsn_name"') SHR REU"
   RETURN

Write_dsn:
   "EXECIO * DISKW ddname (STEM line. FINIS)"
   RETURN
```

Additional Program

```
/***************************** REXX ***********************************/
/* Program                                                            */
/* rx20123                                                            */
/* Description                                                        */
/* REXX program to check customer numbers                             */
/* Author       : Michaelangelo DeParma                               */
/* Date         : 2nd February 2000.                                  */
/*----------------------- Amendment History -------------------------*/
/**********************************************************************/
CALL read_dsn
CALL check_data
EXIT

read_dsn:
   dsn_name = "crone90.crone.rexx(customer)"
   "ALLOC DD(tstname) DSN('"dsn_name"') SHR"
   "EXECIO * DISKR tstname (STEM line. FINIS)"
   RETURN

check_data:
   DO line_counter = 1 TO line.0
      IF LENGTH(line.line_counter) <> 6 | line.line_counter < 0|,
                                        line.line_counter > 999999 THEN DO
         SAY line.line_counter||" is not a valid record"
      END
   END
   RETURN
```

Work Section 11.1

```
/***************************** REXX ***********************************/
/* Program                                                            */
/* rx20131                                                            */
/* Description                                                        */
/* REXX program to demonstrate stacks                                 */
/* Author       : Michaelangelo DeParma                               */
/* Date         : 2nd February 2000.                                  */
/*----------------------- Amendment History -------------------------*/
/**********************************************************************/
ADDRESS "TSO"
"CLEAR"
CALL get_data
CALL display_data
EXIT

get_data:
   DO FOREVER
      SAY "Please enter your name :"
      SAY "Please enter your name :"
      PARSE UPPER EXTERNAL full_name un_used
      IF full_name = "STOP" THEN DO
         LEAVE
      END
      ELSE DO
         QUEUE full_name
      END
   END
   RETURN

Display_data:
   no_in_stack = QUEUED()
   DO no_in_stack
      PARSE PULL name
      SAY name
   END
   RETURN
```

Work Section 11.2

```
/***************************** REXX ***********************************/
/* Program                                                            */
/* rx20132                                                            */
/* Description                                                        */
/* REXX program to demonstrate stacks                                 */
/* Author      : Michaelangelo DeParma                                */
/* Date        : 2nd February 2000.                                   */
/*----------------------- Amendment History -------------------------*/
/**********************************************************************/
ADDRESS "TSO"
"CLEAR"
CALL get_name
CALL get_dob
CALL display_stack
EXIT

get_name:
   DO FOREVER
      SAY "Please enter your name :"
      PARSE UPPER EXTERNAL full_name un_used
      IF full_name = "STOP" THEN DO
         LEAVE
      END
      ELSE DO
         PUSH full_name
      END
   END
   RETURN

get_dob:
   "NEWSTACK"
   DO FOREVER
      SAY "Please enter your DOB :"
      PARSE UPPER EXTERNAL dob un_used
      IF dob = "STOP" THEN DO
         LEAVE
      END
      ELSE DO
         QUEUE dob
      END
   END
   RETURN

Display_stack:
   no_in_stack = QUEUED()
   DO no_in_stack
      PARSE PULL name
      SAY name
   END
   "DELSTACK"
   no_in_stack = QUEUED()
   DO no_in_stack
      PARSE PULL name
      SAY name
   END
   RETURN
```

Additional Program

```
/****************************** REXX ***********************************/
/* Program                                                             */
/* rx20133                                                             */
/* Description                                                         */
/* REXX program to demonstrate stacks                                  */
/* Author       : Michaelangelo DeParma                                */
/* Date         : 2nd February 2000.                                   */
/*----------------------- Amendment History --------------------------*/
/***********************************************************************/
ADDRESS "TSO"
"CLEAR"
CALL define_data
CALL load_data
CALL display_data
EXIT

define_data:
   dsn_name = "crone90.crone.rexx(test)"
   dsn_check = SYSDSN("'"dsn_name"'")
   IF dsn_check <> "OK" THEN DO
      SAY dsn_check
      EXIT
   END
   ELSE DO
      RETURN
   END

Load_data:
   "ALLOC DD(ddname) DSN('"dsn_name"') SHR REU"
   "EXECIO * DISKR ddname (FINIS)"
   RETURN

Display_data:
   no_in_stack = QUEUED()
   DO no_in_stack
      SAY "Do you wish to see a line from the stack (Yes/end) ?"
      PARSE UPPER EXTERNAL answer
      IF answer = "END" THEN DO
         "DELSTACK"
         RETURN
      END
      ELSE DO
         PARSE PULL line
         SAY line
      END
   END
   RETURN
```

Work Section 12.1

```
/***************************** REXX ***********************************/
/* Program                                                            */
/* rx20141                                                            */
/* Description                                                        */
/* REXX program to display a panel                                    */
/* Author       : Michaelangelo DeParma                               */
/* Date         : 2nd February 2000.                                  */
/*----------------------- Amendment History --------------------------*/
/**********************************************************************/

/*--------------------------------------------------------------------*/
/* Initialise any variables and libraries                             */
/*--------------------------------------------------------------------*/
CALL initialise

/*--------------------------------------------------------------------*/
/* Start main program display loop                                    */
/*--------------------------------------------------------------------*/
DO FOREVER
   CALL display_panel
   msg1 = ""
   return_code = RC
   IF return_code = 8 THEN DO
      EXIT
   END
   ELSE DO
      CALL options
   END
END
EXIT

Initialise:
   /*-----------------------------------------------------------*/
   /* Initialise panels and variables.                          */
   /*-----------------------------------------------------------*/
   ADDRESS ISPEXEC "LIBDEF ISPPLIB DATASET ID('crone90.rx201.ispplib')"
   RETURN

Display_panel:
   /*-----------------------------------------------------------*/
   /* Display the choice selection panel.                       */
   /*-----------------------------------------------------------*/
   ADDRESS ISPEXEC "DISPLAY PANEL(rx20141)"
   RETURN

Options:
   /*-----------------------------------------------------------*/
   /* This will display the choice.                             */
   /*-----------------------------------------------------------*/
   IF msg1 <> "" THEN DO
      RETURN
```

```
END
SELECT
   WHEN choice = 1 THEN DO
   WHEN choice = 1 THEN DO
      msg1 = "You chose 1"
   END
   WHEN choice = 2 THEN DO
      msg1 = "You chose 2"
   END
   WHEN choice = 3 THEN DO
      msg1 = "You chose 3"
   END
   WHEN choice = 4 THEN DO
      msg1 = "You chose 4"
   END
   WHEN choice = 5 THEN DO
      msg1 = "You chose 5"
   END
   OTHERWISE DO
      msg1 = "You must enter one of the options listed."
   END
END
RETURN
```

Work Section 12.2

```
/****************************** REXX ***********************************/
/* Program                                                             */
/* rx20142                                                             */
/* Description                                                         */
/* REXX program to use a panel to collect names                        */
/* Author       : Michaelangelo DeParma                                */
/* Date         : 2nd February 2000.                                   */
/*---------------------- Amendment History -------------------------*/
/***********************************************************************/

/*---------------------------------------------------------------------*/
/* Initialise any variables and libraries                              */
/*---------------------------------------------------------------------*/
CALL initialise

/*---------------------------------------------------------------------*/
/* Start main program display loop                                     */
/*---------------------------------------------------------------------*/
DO FOREVER
   CALL display_panel
   return_code = RC
   IF return_code = 8 THEN DO
      LEAVE
   END
   ELSE DO
      CALL store_name
   END
END
CALL display_names
EXIT

Initialise:
   /*-----------------------------------------------------------*/
   /* Initialise panels and variables.                          */
   /*-----------------------------------------------------------*/
   ADDRESS ISPEXEC "LIBDEF ISPPLIB DATASET ID('crone90.rx201.ispplib')"
   msg1 = ""
   RETURN

Display_panel:
   /*-----------------------------------------------------------*/
   /* Display the name selection panel                          */
   /*-----------------------------------------------------------*/
   ADDRESS ISPEXEC "DISPLAY PANEL(rx20142)"
   RETURN

Store_name:
   /*-----------------------------------------------------------*/
   /* This will store the names on the stack                    */
   /*-----------------------------------------------------------*/
```

```
   QUEUE names
   msg1 = names||" stored."
   RETURN

Display_names:
   /*-------------------------------------------------------------*/
   /* This will display the names.                                */
   /*-------------------------------------------------------------*/
   no_in_stack = QUEUED()
   IF no_in_stack = 0 THEN DO
      SAY "No names stored."
   END
   ELSE DO
      DO no_in_stack
         PARSE PULL old_name
         SAY old_name
      END
   END
   RETURN
```

Additional Program

```
/****************************** REXX ***********************************/
/* Program                                                             */
/* rx20143                                                             */
/* Description                                                         */
/* REXX program to add a new customer to a customer file               */
/* Author       : Michaelangelo DeParma                                */
/* Date         : 2nd February 2000.                                   */
/*----------------------- Amendment History --------------------------*/
/***********************************************************************/

/*---------------------------------------------------------------------*/
/* Initialise any variables and libraries                              */
/*---------------------------------------------------------------------*/
CALL initialise

/*---------------------------------------------------------------------*/
/* Start main program display loop                                     */
/*---------------------------------------------------------------------*/
DO FOREVER
   CALL display_panel
   return_code = RC
   IF return_code = 8 THEN DO
      LEAVE
   END
   ELSE DO
      CALL store_name
   END
END
CALL save_data
EXIT

Initialise:
   /*-----------------------------------------------------------*/
   /* Initialise panels and variables.                          */
   /*-----------------------------------------------------------*/
   dsn_name = "crone90.crone.rexx(custmer1)"
   ADDRESS ISPEXEC "LIBDEF ISPPLIB DATASET ID('crone90.rx201.ispplib')"
   ADDRESS "TSO" "ALLOC DD(ddname) DSN('"dsn_name"') SHR REU"
   msg1 = ""
   RETURN

Display_panel:
   /*-----------------------------------------------------------*/
   /* Display the new customer panel.                           */
   /*-----------------------------------------------------------*/
   ADDRESS ISPEXEC "DISPLAY PANEL(rx20143)"
   RETURN

Store_name:
   /*-----------------------------------------------------------*/
```

```
   /* This will store the names on the stack                    */
   /*-----------------------------------------------------------*/
   QUEUE custname postcode
   msg1 = custname||" "||postcode||" accepted."
   RETURN

save_data:
   /*-----------------------------------------------------------*/
   /* This will check the stack and if it is required to        */
   /* be saved to the customer file.                            */
   /*-----------------------------------------------------------*/
   no_in_stack = QUEUED()
   IF no_in_stack = 0 THEN DO
      SAY "No customers to add."
   END
   ELSE DO
      SAY "Do you want to save the data (YES/NO) ?"
      PARSE UPPER EXTERNAL answer un_used
      IF answer = "YES" THEN DO
         /*--------------------------------------*/
         /* Add NULL character to stop EXECIO    */
         /* returning to the terminal for input. */
         /*--------------------------------------*/
         QUEUE ""
         "EXECIO * DISKW ddname (FINIS)"
         IF RC = 0 THEN DO
            SAY "Data stored successfully."
         END
      END
      ELSE DO
         NOP
      END
   END
   /*-----------------------------------------------------------*/
   /* Empty stack and free ddname.                              */
   /*-----------------------------------------------------------*/
   "DELSTACK"
   ADDRESS "TSO" "FREE DD("ddname")"
   RETURN
```