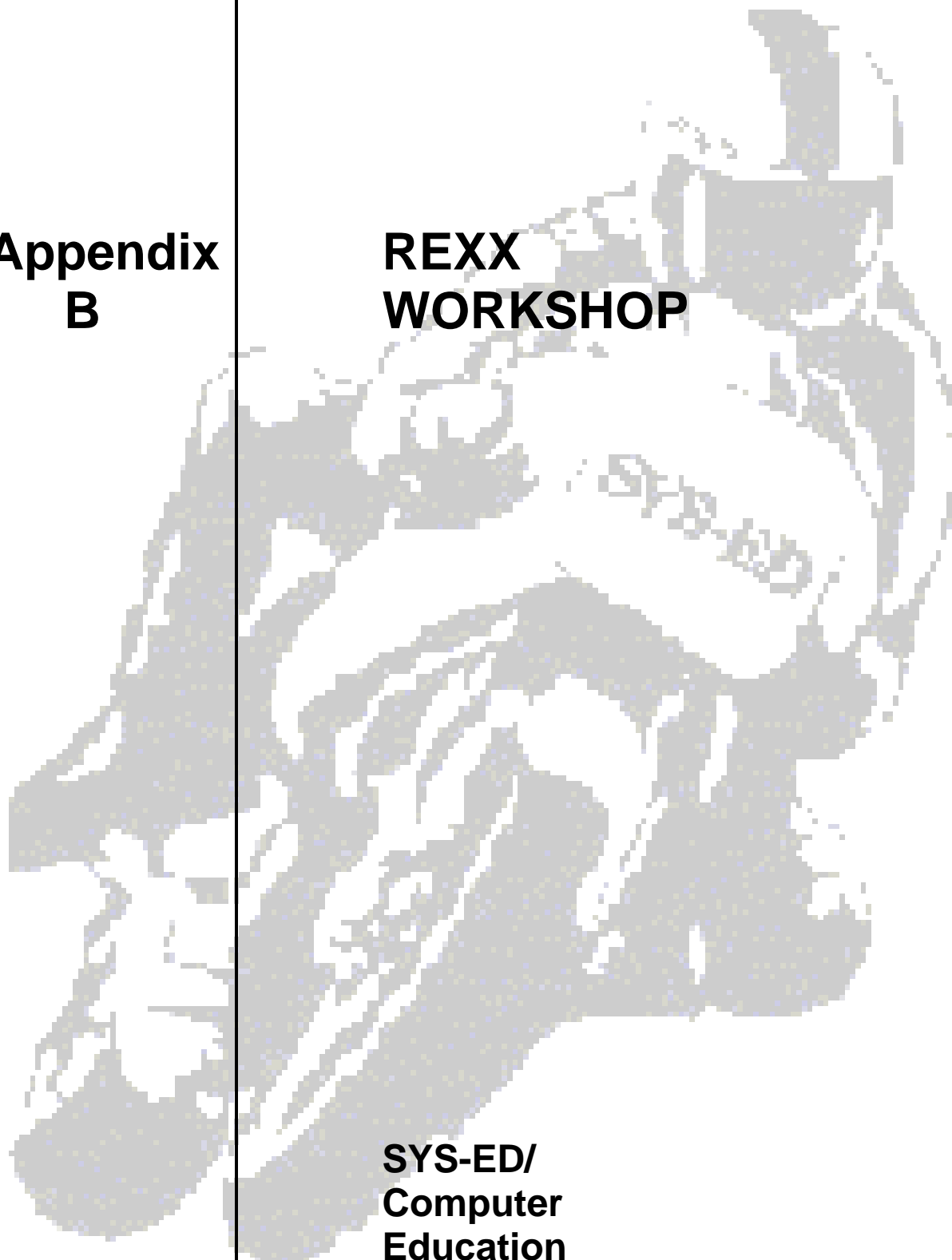


**Appendix  
B**

**REXX  
WORKSHOP**

**SYS-ED/  
Computer  
Education  
Techniques, Inc.**



---

**1 REXX I/O**
**Description:****Step 1**

Code a data entry program that inputs the following fields: SSN, NAME and SALARY. Assume that all the data is correct; it is not necessary to edit the data.

Write the inputted data to an output dataset. The record should be in the following format:

Field	Starting	Ending
SSN	1	9
NAME	10	35
SALARY	40	47

**Step 2**

Code another REXX program that reads the dataset created in Step 1 and displays a report in the following format:

```

999-99-9999  XXXXXXXXXXXXXXXXXXXXXXXX  99999999
999-99-9999  XXXXXXXXXXXXXXXXXXXXXXXX  99999999
999-99-9999  XXXXXXXXXXXXXXXXXXXXXXXX  99999999
999-99-9999  XXXXXXXXXXXXXXXXXXXXXXXX  99999999

```

**Step 3**

Code another REXX program that allows the user to enter a dataset name and copies the file to a backup. The program will check if the dataset exists before backing up the file. The backup dataset will use the input dataset name with a suffix of BMMDDYY.

---

## 2 Jumble Program

```
/* REXX */
do forever
  say 'What do you want to do'
  say '    A. Play a game'
  say '    B. Do some work'
  say '    C. Sleep'
  say '    D. Exit'
  say 'Enter option'
  pull option
  if verify(option,'ABCD') = 0 then leave
  say 'Invalid option try again'
end
if verify(option, 'BCD') = 0 then exit

correctWord = 'fish'
correctWord = translate(correctWord) /* uppercase */
jumble = copies('*',length(correctWord))

do i = 1 to length(correctWord)
  do forever
    rand=random(1,length(correctword))
    if substr(jumble,rand,1) = '*' then do
      jumble=overlay(substr(correctWord,i,1),
        jumble, rand)
    end
  end
end

end
say 'The jumble word is ' jumble
say 'you have three guesses'
do 3
  say 'Enter your answer'
  pull answer
  if answer = correctWord then do
    say 'Good going'
    exit
  end
end
end
say 'Sorry Charlie'
```

---

### 3 REXX Compound Variable

---

**Description:**

1. Code a REXX routine that allows the user to enter salary for every employee in a department.
  - Salary must be a positive numeric field.
  - Store the salary information in a stem (compound variable).
2. Code an external subroutine that computes the average salary in the compound variable.
3. Code a function that returns the minimum salary in the compound variable.
4. Code a function that returns the maximum salary in the compound variable.

---

## 4 REXX Function

### Description:

1. Display the following menu:

```
WHAT DO YOU WANT TO DO
```

- A. PLAY A GAME
- B. DO SOME WORK
- C. SLEEP (MEDITATE)
- D. EXIT

```
ENTER OPTION
```

2. Check the option. It must be an A-D. Do not check with a SELECT or an IF that checks for an A, B C or D. Use one of the functions to check its value. If it is not valid, display an error, clear the screen and let the user try again.

3. If the user select options A, write the following game:

- Initialize a field with a word; for example FLD=GAME.
- Jumble up the word in order that every time the game is played, it looks different.

Use the RANDOM function.

- Display the jumbled word and allow the user to guess the word.

If he/she does not guess in three tries, relPL the system.

All other option in the menu should terminate the program.

---

## 5 Display and Customize JCL

**Description:**

The user wants to run the IEBGENER utility; but does not know JCL.

1. Write a REXX program that asks the user for the following information and displays the JCL with the appropriate portion filled in from the users input:

JOB NAME

INPUT DATASET NAME

OUTPUT DATASET NAME

2. All fields are required. If they are not entered, display an appropriate error message and have the use enter the information.

**Sample Output:**

```
//GENER1 PROC
//STEP01 EXEC PGM=IEBGENER
//SYSPRINT DD SYSOUT=*
//SYSUT1 DD DSN=$7101.DAVE.DATA,DISP=SHR
//SYSUT2 DD DSN=$7101.DAVE.BKUP,DISP=SHR
//SYSIN DD DUMMY
```

---

**6 REXX – Function/Subroutine Lab****Description:**

1. Write a function that accepts a 9 character Social Security Number and returns an 11 character Social Security Number in the format 999-99-9999.
2. Invoke the function as a function. For example:

```
SAY SSNFORMAT('123456789')
```

This should return 123-45-6789

3. Invoke the function with a call; for example `CALL SSNFORMAT(('123456789'))`. Display the results as 123-45-6789.

---

**7 Solution**

```
PROC 0
WRITE ENTER SSN
READ INPSSN
IF &DATATYPE(&INPSSN) = CHAR THEN EXIT

ALLOC DS(SYSED.CLIST(DATA)) DD(EMP) SHR
OPENFILE EMP INPUT
DO WHILE A = A
  GETFILE EMP
  SET RCD = &EMP
  SET SSN = &SUBSTR(1:9,&RCD)
  IF &SSN = &INPSSN THEN DO
    WRITENR &SUBSTR(1:3,&SSN)&STR(-)-
&SUBSTR(4:5,&SSN)&STR(-)&SUBSTR(6:9,&SSN)
    SET RCD = &SUBSTR(10:30,&RCD)
    SET I = 1
    DO WHILE &I < &LENGTH(&RCD)
      IF &SUBSTR(&I:&I,&RCD) = &STR( ) THEN DO
        SET POSSPACE = &I
        SET I = 999
      END
      SET I = &I + 1
    END
    WRITE &SUBSTR(&POSSPACE:&LENGTH(&RCD),&RCD), -
&SUBSTR(1:&POSSPACE,&RCD)

    EXIT
  END
  WRITE &SSN
END
CLOSFIE EMP
FREE DD(EMP)
```