Scilab Textbook Companion for
Programming In Ansi C
by E. Balagurusamy[1]

Created by
Devender Thakur
computer science
Computer Engineering
Shoolini University, Solan
College Teacher
Mr. Amit Nayyar
Cross-Checked by

October 7, 2013

[1] Funded by a grant from the National Mission on Education through ICT, http://spoken-tutorial.org/NMEICT-Intro. This Textbook Companion and Scilab codes written in it can be downloaded from the "Textbook Companion Project" section at the website http://scilab.in

Book Description

Title: Programming In Ansi C

Author: E. Balagurusamy

Publisher: Tata McGraw - Hill Education, New Delhi

Edition: 4

Year: 2008

ISBN: 978-0-07-064822-7

Scilab numbering policy used in this document and the relation to the above book.

Exa Example (Solved example)

Eqn Equation (Particular equation of the above book)

AP Appendix to Example(Scilab Code that is an Appednix to a particular Example of the above book)

For example, Exa 3.51 means solved example 3.51 of this book. Sec 2.3 means a scilab code whose theory is explained in Section 2.3 of the book.

Contents

List of Scilab Codes

Chapter 1

Overview of C

Scilab code Exa 1.3 Printing a mesage

```
//              Example 1.3
// SAMPLE PROGRAM 1: PRINTING A MESSAGE

//Printing Begins
printf(”I see,I remember”) //Printing using printf() function
//Printing ends


//We can also print a message using disp() function in scilab
disp(”I see,I remember”) //Printing using disp() function
```

Scilab code Exa 1.4 Adding two numbers

```
//              Example 1.4
// SAMPLE PROGRAM 2: ADDING TWO NUMBERS

number=100;
amount=30.75+75.35 ;      //Addition of two numbers
disp(number) ;            //Display  value of number variable
//Display value of amount in floating point with five places in all and two places to the right of decimalpoint
printf(”%5.2f”,amount);
```

Scilab code Exa 1.5 Interest Calculation

```
//              Example 1.5
// SAMPLE PROGRAM 3: INTEREST CALCULATION

PRINCIPAL=5000.00;
PERIOD=10;
//Assignment Statements
amount=PRINCIPAL;
inrate=0.11;
year=0;
//Computation using while loop
while(year<=PERIOD)
    printf(”%2d %8.2f\n”,year,amount);
    value=amount+inrate*amount;
    year=year+1;
    amount=value;
end //End of while loop
```

Scilab code Exa 1.6 Use of subroutines

```
//               Example 1.6
// SAMPLE PROGRAM 4: USE OF SUBROUTINES

//Program using function

function []=mul(a,b)    // mul()function starts (i.e. definition starts)
   y=a*b;
   printf("Multiplication of %d and %d is %d",a,b,y);
endfunction         // mul()function ends
a=5;b=10;
//Calling mul() function
mul(a,b)
```

Scilab code Exa 1.7 Use of math functions

```
//               Example 1.7
// SAMPLE PROGRAM 5: USE OF MATH FUNCTIONS

//Program using cosine function
angle=0;MAX=180;
printf("            Angle        Cos(angle)\n");
  while(angle<=MAX)
    x=(%pi/MAX)*angle;
    y=cos(x);              //Use of cosine function
    printf(”%15d %13.4f\n”,angle,y);
    angle=angle+10;
  end
```

Chapter 2

Constants Variables and Data Types

Scilab code Exa 2.1 Representation of integer constants on a 16 bit computer

```
//          Example 2.1
// Representation of integer constants on a 16-bit computer.

disp("Integer values");
//Integer values larger than 32767 are not stored properly on 16-bit machine
printf("%d %d %d \n",int16(32767),int16(32767+1),int16(32767+10));

disp("Long integer values");
//To store long integers properly, use int32 integer type
printf("%ld %ld %ld \n",int32(32767),int32(32767+1),int32(32767+10));
//The same result as from above statement can be achieved directly from below commented statement
//printf("%ld %ld %ld\n",32767,32767+1,32767+10);
```

Scilab code Exa 2.1cs Case study 1 avg of numbers

```
//    Case Study:-Chapter 2 Page No.-47
//   1.Calculation of Average of numbers

N=10;sum1=0;count=0;     //Initialization of variables
printf(” Enter ten numbers”);
while(count<N)
     number=scanf(”%f”);  //Reading number(using scanf() function)
     sum1=sum1+number;
     count=count+1;
 end
 average=sum1/N;         //Avarage is calculated
printf(” N = %d Sum1 = %f”,N,sum1 );
printf(” Average = %f”,average );
```

Scilab code Exa 2.2 typical declarations assignments and values stored in various types of variables

```
//      Example 2.2
// Program shows typical declarations,assignments and values stored in various types of variables.

//Declarations and Assignments
m=int16(54321);
```

```
n=int32(1234567890);
k=uint16(54321);
//Assignments
x=1.234567890000;        //Bydefault type is double in scilab
y=9.87654321;            //------------------------------------
p=1.0;q=1.0;
//Printing
printf(” m=%d\n”,m)
printf(” n=%ld\n”,n)
printf(” x=%.12f\n”,x)
printf(” x=%f\n”,x)
printf(” y=%.12f\n”,y)
printf(” y=%f\n”,y)
printf(” k=%u p=%f q=%.12f\n”,k,p,q)
```

Scilab code Exa 2.2cs Case study temprature in Farenheit and Celsius

```
//          Case Study:-Chapter 2    Page no.-48
//   2.Solution of temprature in Farenheit and Celsius
F_LOW=0;
F_MAX=250;
STEP=25;
fahrenheit=F_LOW;       //Initialization
printf(”Fahrenheit       Celsius\n\n”);
while(fahrenheit <= F_MAX)
    celsius=(fahrenheit-32.0)/1.8;      //conversion from Farenheit to Celsius
    printf(”%6.2f           %7.2f\n”,fahrenheit,celsius);
    fahrenheit=fahrenheit+STEP;
```

```
end
```

Scilab code Exa 2.3 use of scanf function

```
//                 Example 2.3
// The program illustrate the use of scanf() function
disp(”Enter an interger number:”);
number=scanf(”%d”);           //Read from keyboard
if(number<100) then
    disp(”Your number is smaller than 100”);
else
    disp(”Your number contain more than two digits”);
end
```

Scilab code Exa 2.4 Interest calculation using scanf

```
//                 Example 2.4
// Sample program 3(exm1.5) discussed in chapter 1 can be convered in to a more flexible intractive program using scanf() function
disp(”Enter in single line separted by space”);
disp(”Input amount,interest rate, and period”);
[amount,inrate,period]=scanf(”%f %f %d”);      //use of scanf()
year=1;
//Computation using while loop
while(year<=period)
```

```
         value=amount+inrate*amount;
         printf("%2d Rs %8.2f\n",year,value)
        year=year+1;
        amount=value;
end
```

Chapter 3

Operators and Expressions

Scilab code Exa 3.1 Use of integer arithmetic

```
//                    Example 3.1
//The program shows the use of integer arithmetic to convert a given number-
//of days into months and days

days=input('Enter days:');
months=int16(days/30);          //Compute for months
days =int16(pmodulo(days,30));  //compute for days
disp(days,"Days =",months,"Months =");
```

Scilab code Exa 3.1cs case study 1 salesmans salary

```
//        Case Study:-Chapter 3,Page No:76
//         1.Salesman's Salary


BASE_SALARY=1500.00;    //Minimum base salary
```

```
BONUS_RATE=200.00;      //Bonus for every computer sold
COMMISSION=0.02;        //Commission on total monthly sales
printf("Input number sold and price\n[Enter in single line separated by space]");
[quantity,price] = scanf("%d %f");    //Input quantity and price
//Computation for bonus,commission and gross_salary of a salesman
bonus = BONUS_RATE*quantity;
commission = COMMISSION*quantity*price;
gross_salary =BASE_SALARY + bonus + commission;
printf("Bonus                  = %6.2f\n",    bonus);
printf("Commission             = %6.2f\n",    commission);
printf("Gross  salary          = %6.2f\n",    gross_salary);
```

Scilab code Exa 3.2 Sequence of squares of numbers

```
//                    Example 3.2
//Program to print a sequence of squares of numbers.

N=100;A=2;
a=A;
while(a<N)
    disp(a);     //Prints square of number
    a=a^2;       //compute square of number
end
```

Scilab code Exa 3.2cs case study 2 solution of the quadratic equation

```
//            Case Study:-Chapter 3,Page No:77
//          2.Solution of the Quadratic equation


printf("Input values of a, b, and c \n");
a=input("a =");
b=input("b =");
c=input("c =");
discriminant = b^2-4*a*c;
if(discriminant<0)
    printf("\n\nROOTS ARE IMAGINARY\n");
else
    //Computes root1 and root2
    root1 = (-b + sqrt(discriminant))/(2.0*a);
    root2 = (-b - sqrt(discriminant))/(2.0*a);
    printf("\n\nRoot1 = %5.2f\n\nRoot2 = %5.2f\n",root1,root2 );
end
```

Scilab code Exa 3.3 Different kind of operators

```
//                          Example 3.3
//The program employs diffrent kind of operators.The results of their evaluation are also shown for comparison

```

```
//Increment(i.e. ++)/Decrement(--) operators are unavailable in Scilab
a=int16(15);
b=int16(10);
a=a+1;  // Replacement for ++a
c=a-b;
printf("a = %d b = %d c = %d\n",a,b,c);
b=b+1;  // Replacement for b++
d=b+a;
printf("a = %d b = %d d = %d\n",a,b,d);
printf("a/b = %d\n",a/b);                        // Division operator
printf("pmodulo(a,b) = %d\n",pmodulo(a,b));  // Modulus operatior
printf("a*b = %d\n",a*b);                        // Multiplication
//In scilab ther is no conditional operator(?:), hence 'if' can be used in place of ?:
if(c>d) then
    disp(1);
end
if(c<d) then
    disp(0);
end
```

Scilab code Exa 3.4 Use of variables in expressions

```
//                    Example 3.4
//The program illustrates the use of variables in expressions and their evaluation.

a=9;b=12;c=3;
//Expressions and their evaluations
```

```
x=a-b/3+c*2-1;
y=a-b/(3+c*(2-1));
z=a-(b/(3+c)*2)-1;

printf(”x=%f\n”,x)
printf(”y=%f\n”,y)
printf(”z=%f\n”,z)
// disp can be used..
//disp(x,”x=”)
//disp(y,”y=”)
//disp(z,”z=”)
```

Scilab code Exa 3.5 Round off errors

```
//                      Example 3.5
//Output of program shows round-off errors that can occur in computation of floating point numbers

//Sum of n terms of 1/n
count=1;
sum1=0;
n=input(”Enter value of n:”);
term=1.0/n;
while(count<=n)
    sum1=sum1+term;
    count=count+1;
end
printf(”Sum= %f”,sum1);
```

Scilab code Exa 3.6 Cast to evaluate the equation

```
//                    Example 3.6
//Program using a cast to evaluate the equation.

sum1=0;
for n=int8(1:10)
    sum1=sum1+1/double(n);   //conversion from ’int’ to ’double’ or ’float’
     printf(”%2d %6.4f\n”,n,sum1);
end
```

Chapter 4

Managing Input and Output Operations

Scilab code Exa 4.1 Use of getchar function

```
//                    Exaymple 4.1
// The program shows the use of getchar function in an intractive environment.
//In Scilab in place of getchar function scanf function can be used to get
//character as there is no getchar function in Scilab.

disp(”Would you like to know my name?”);
disp(”Type Y for YES and N for NO:”);
answer=scanf(”%c”);                            //Reading character
if (answer==’Y’)|(answer==’y’) then            //Test for answer
    disp(”My name is BUSY BEE”);
else
    disp(”You are good for nothing”)
end
```

Scilab code Exa 4.1cs Case study 1 Inventory report

```
//                    Case Study:-Chapter 4 ,Page No:106
//                    1.Inventory Report

ITEMS=4;
i=1;
printf("[Enter in single line seperated be spaces]\n");
while(i <= 4)
    printf("Enter code,quantity, and rate:");
     [code(i),quantity(i),rate(i)]=scanf("%s %d %f");
     i=i+1;
end
printf("INVENTORY REPORT\n");
printf("-----------------------------------------------------\n");
printf("Code          Quantity      Rate        Value\n");
printf("-----------------------------------------------------\n");
total_value=0;
i=1;
while(i<=ITEMS)
    value=quantity(i)*rate(i);
    printf("%6s %10d %10.2f     %e\n",code(i),quantity(i),rate(i),value);
    total_value= total_value+value;
    i=i+1;
```

```
end
    printf("--------------------------------------------------------\n");
    printf("                  Total Value = %e\n", total_value);
    printf("--------------------------------------------------------\n");
```

Scilab code Exa 4.2 Whether character is alphabet or digit or special character

```
//                    Example 4.2
// The program requests the user to enter a character and display a message on
// the screen telling the user whether the character is an alphabet or digit,
// or any other special character.

disp("Press any key");
character=scanf("%c");                          // Reading character
if (isletter(character)) then                   //Test for letter
    disp("The character is a letter");
    elseif (isdigit(character)) then            //Test for digit
            disp("The character is a digit");
        else
            disp("The character is not alphanumeric");
 end
```

Scilab code Exa 4.2cs Case study 2 Reliability graph

```
//            Case study:chapter 4
//           2.Reliability Graph

LAMBDA=0.001;
for i=1:27
    printf(”--”);
end
printf(”\n”);
for t=0:150:3000
    r=exp(-LAMBDA*t);
    R=int32(50*r+0.5);
    printf(”|”);
    for i=1:R
          printf(”*”);
    end
    printf(”#\n”);
end
for i=1:2
    printf(”|\n”);
end
```

Scilab code Exa 4.3 Print character in reverse case

```
//                   Example 4.3
// A program that reads a character from the
   keyboard and then print in reverse
```

```
//case ,that is,if input is in upper case,the output will be lower case and vice-versa
disp(”Enter an alphabet”);
alphabet=scanf(”%c”);                    //Reading character
if((ascii(alphabet))>=97) then
    disp(convstr(alphabet,”u”));        //Reverse and display
else
    disp(convstr(alphabet,”l”));        //Reverse and display
end
```

Scilab code Exa 4.4 Input formatting options

```
//                  Example 4.4
// The program illustrates the various options for reading
//integers are experimented in this program


printf(”Enter three integer numbers\n”);
[n,a,b,c]=mscanf(”%d %*d %d”);
disp(c,b,a);
printf(”Enter two 4-digit numbers\n”);
[n,x,y]=mscanf(”%2d %4d”);
printf('%d %d\n',x,y);

printf(”Enter two integers\n”);
[n,a,x]=mscanf(”%d %d”);
printf('%d %d\n',a,x);

printf(”Enter a nine digit number\n”);
```

```
[n,p,q,r]=mscanf(”%3d %4d %3d”);
printf(’%d %d %d\n’,p,q,r);

printf(”Enter two three digit numbers\n”);
[n,x,y]=mscanf(”%d %d”);
printf(’%d %d \n’,x,y);
```

Scilab code Exa 4.5 Reading of real numbers

```
//                Example 4.5
//Reading of real numbers(in both decimal point and exponential notation

printf(”Values of x and y:”);
[x,y]=scanf(”%f %e”); //reading x [decimal point] and y[exponential] from standard input
printf(”\n”);
printf(”x = %f\ny = %f\n”,x,y);
printf(”Values of p and q:”);
[p,q]=scanf(”%lg %lg”);
printf(”\np = %.12f\nq = %.12e\n”,p,q);
```

Scilab code Exa 4.6 Reading of strings

```
//                  Example 4.6
//  Reading of strings using %wc and %ws.

//scanf() can only read one character at a time and return it to name1, therefore
```

```
//mscanf() is used that can read more than one character
printf("Enter serial number and name one:");
[n,no,name1]=mscanf("%d %15c");
printf("%d %15s\n",no,name1);

//Read and display a string
printf("Enter serial number and name two:");
[no,name2]=scanf("%d %s");
printf("%d %15s\n",no,name2);

//Read and display a string of width 15
printf("Enter serial number and name three:");
[no,name3]=scanf("%d %15s");
printf("%d %15s",no,name3);
```

Scilab code Exa 4.8 Testing for correctness

```
//                 Example 4.8
// The program illustrates the testing for correctness of reading data by scanf function

//In Scilab mscanf function is similar to scanf function of c
printf("Enter values of a,b and c");
[n,a,b,c]=mscanf("%d %f %c");
if(n==3) then             //Test condition
    printf("a = %d b = %f c = %c",a,b,c);
else
    printf("Error in input.");
end
```

Scilab code Exa 4.9 Output of integer numbers

```
//              Example 4.9
// The program illustrates the output of integer numbers under various formats

m=int16(12345);         //integer variable m
n=int32(987654);        //integer variable n
printf('%d\n',m);
printf('%10d\n',m);
printf('%010d\n',m);
printf('%-10d\n',m);
printf('%10d\n',n);
printf('%10d\n',-n);
```

Scilab code Exa 4.10 Printing a real number

```
//              Example 4.10
// The program illustrates all the options of printing a real number.

y=98.7654;          //real number
//Various options of printing a real number
printf('%7.4f\n',y);
printf('%f\n',y);
printf('%7.2f\n',y);
printf('%-7.2f\n',y);
printf('%07.2f\n',y);
printf('%7.2f\n',y);
printf('\n');
printf('%10.2e\n',y);
printf('%12.4e\n',-y);
printf('%-10.2e\n',y);
printf('%e\n',y);
```

Scilab code Exa 4.11 Printing of character and strings

```
//                   Example 4.11
// The program illustrates the printing of character and strings.

x='A';                              //x variable has character
name='ANIL KUMAR GUPTA';   //name variable has string
disp("OUTPUT OF CHARACTERS");
printf('%c\n%3c\n%5c\n',x,x,x);
printf('%3c\n%c\n',x,x);
printf('\n');
disp("OUTPUT OF STRINGS"); //disp can also be used for display
printf('%s\n',name);
printf('%20s\n',name);
printf('%20.10s\n',name);
printf('%.5s\n',name);
printf('%-20.10s\n',name);
printf('%5s\n',name);
```

Chapter 5

Decision Making and Branching

Scilab code Exa 5.1 Ratio

```
//                          Example 5.1
// The program reads four values a,b,c and d from the terminal and
//evaluates the ratio of (a+b) to (c+d) and prints the result,if c-d
//is not equal to zero

disp("Enter four integer values");
a=int8(input("a="));
b=int8(input("b="));
c=int8(input("c="));
d=int8(input("d="));

if(c-d ~= 0) then       //Execute statement block
    ratio=double(a+b)/double(c-d);
    printf("Ratio=%f\n",ratio);
end
```

Scilab code Exa 5.1cs Case study 1 range of numbers

```
//                Case Study:-Chapter 5, Page No:139
//                 1.Range of numbers

sum1=0;
count=0;
f=0;value=1;
printf("Enter numbers[press enter after each number]:\n input a NEGATIVE number to end\n");
while(value)
    [value]=scanf("%f");  //Read data
    if(value<0) then
        break;
    end
    count=count+1;
    //Calculating height and lowest value
    if(count==1) then
        high=value;
        low=value;
     elseif(value>high) then
         high=value;
     elseif(value<low) then
         low=value;
     end
     //Calculate sum
     sum1=sum1+value;
end
average=sum1/count;     //Average cost
range1=high-low;        //Range of values
//Print the results
printf("Total values: %d\n",count);
printf("Highest-values: %f\nLowest-value: %f\n",high,low);
printf("Range: %f\n Avarage: %f\n",range1,average);
```

Scilab code Exa 5.2 counts the number of boys

```
//                          Example 5.2
// The program counts the number of boys whose weight is less than 50Kg
//and height is greater than 170cm.

count=int(0);
disp("Enter weight and height for 10 boys");
for i=1:10
    [weight,height]=scanf("%f %f");
    if(weight<50&height>170) then      // Test for weight and height
        count=count+1;
    end
end
disp("Number of boys with weight <50 kg and height >170 cm =");
disp(count)
```

Scilab code Exa 5.2cs Case study 2 Pay Bill Calculations

```
//                      Case Study:-Chapter 5
//                2.Pay-Bill Calculations
CA1=1000;
CA2=750;
CA3=500;
CA4=250;
EA1=500;
EA2=200;
EA3=100;
EA4=0;
level=1;
while(level)
    printf(”Enter 0[zero] for level to end”);
    //Read data
    level=input(”Enter level:”);
    if(level==0)
        break;
    end
    printf(”Enter job number, and basic pay\n”);
    //Read data
    [jobnumber,basic]=scanf(”%d %f”);
    //Decide level number and calculate perks
    select level
    case 1 then perks=CA1+EA1;
    case 2 then perks=CA2+EA2;
    case 3 then perks=CA3+EA3;
    case 4 then perks=CA4+EA4;
    else
        printf(”Error in level code”);
        return;
    end

    house_rent=0.25*basic;
    //Calculate gross salary
    gross=basic+house_rent+perks;
    //Calculate income tax
    if (gross<=2000) then
        incometax=0;
    elseif(gross<=4000)
```

```
        incometax=0.03*gross;
    elseif(gross<=5000)
        incometax=0.05*gross;
    else
        incometax=0.08*gross;
    end
    //Compute the net salary
    net=gross-incometax;
    //Print the results
    printf(”%d %d %.2f\n”,level,jobnumber,net);
end
printf(”END OF THE PROGRAM”);
```

Scilab code Exa 5.3 Evaluate the power series

```
//                          Example 5.3
// A program to evaluate the power series. It uses if...else to test accuracy.
// e^x=1+x+x^2/2! + x^3/3! +    +x^n/n!,0<x<1

ACCURACY=0.0001;
x=input(”Enter value of x:”);
n=1;term=1;sum1=1;count=int8(1);
while(n<=100)
    term=term* x/n;
    sum1=sum1+term;
    count=count+1;
    if(term<ACCURACY) then      //Test for accuracy
        n=999;
    else
        n=n+1;
     end
end
//Print results
printf(”Term=%d Sum=%f”,count,sum1);
```

Scilab code Exa 5.4 Largest of the three numbers

```
//                         Example 5.4
// The program selects and prints the largest of the three numbers
//using nested if...else statement

disp("Enter three values");
A=input("A=");
B=input("B=");
C=input("C=");
disp("Largest value is:");
if(A>B) ,                   //Test for largest between A&B
    if(A>C) ,               //Test for largest between A&C
        disp(A);
     else
         disp(C);
     end
else
    if(C>B) ,              //Test for largest between C&B
        disp(C);
    else
        disp(B);
     end
end
```

Scilab code Exa 5.5 Reads the customer number and power consumed

```
//                              Example 5.5
// The program reads the customer number and power consumed and prints
//the amount to be paid by the customer

disp("Enter CUSTOMER NO. and UNITS consumed");
//Read data
custnum=input("CUSTOMER NO.:");
units=input("UNITS:");
//Use of else...if ladder
//Calculates charges
if(units<=200) ,
    charges=0.5*units;
elseif(units<=400) ,
    charges=100+0.65*(units-200);
elseif(uints<=600) ,
    charges=230+0.8*(units-400);
else
    charges=390+(units-600);
end
//Print results
printf("Customer No:%d  Charges=%.2f",custnum,charges);
```

Scilab code Exa 5.6 Loan applications and to sanction loans

```
//                          Example 5.6
// A program to  process loan applications and to sanction loans.

MAXLOAN=50000;
disp("Enter the values of previous two loans");
loan1=int32(input("Enter first loan:"));
loan2=int32(input("Enter second loan:"));
loan3=int32(input("Enter the values of new loan:"));
sum23=loan2+loan3;
//Calculate the sanction loan
if(loan1>0),
    sancloan=0;
 elseif(sum23>MAXLOAN),
     sancloan=MAXLOAN-loan2;
 else
     sancloan=loan3;
end
//Print the results
printf("Previous loans pending:%d %d\n",loan1,loan2);
printf("Loan requested =%d\n",loan3);
printf("Loan sanctioned =%d\n",sancloan);
```

Scilab code Exa 5.7 square root for five numbers

```
//                          Example 5.7
// The  program evaluates the square root for five numbers.

count=1;
printf("Enter FIVE real values\n");
while(count<=5)
```

```
    x=scanf("%f");
    if(x<0) then
        printf("Value - %d is negative\n",count);
    else
        y=sqrt(x);            //Calculate square root
        printf("%f\t%f\n",x,y); //Print result
    end
    count=count+1;
end
printf("End of computation");
```

Chapter 6

Decision Making and Looping

Scilab code Exa 6.1 evaluate the equation

```
//               Example 6.1
//A program to evaluate the equation y=x^n when n is a non-negative integer.

x=input("Enter the value of x:");
n=input("Enter the value of n:");
y=1.0;count=1;     //intialization
//Loop begins
while(count<=n)    //Testing
    y=y*x;
    count=count+1;   //Increamenting
end
//End of loop
printf("x = %f; n = %d; x to power n = %f\n",x,n,y);
```

Scilab code Exa 6.1cs Case study 1 Table of Binomial Coefficients

```
//                    Case Study:-Chapter 6, Page No :176
//               1.Table of Binomial Coefficients

MAX=10;
printf(”mx”);
for m=0:10
    printf(”%4d”,m);
end
printf(”\n————————————————————————————————————————————————\n”);
m=0;
//print the table of binomial coefficients for m=10
//Computation using while loop
while(m<=MAX)
    printf(”%2d”,m);
    x=0;
    binom=1;
    while(x<=m)
        if(m==0|x==0)
            printf(”%4d”,binom);    //Print the result i.e. binom
        else
            binom=binom*(m-x+1)/x;  //compute the binomial coefficient
            printf(”%4d”,binom);    //Print the result i.e. binom
        end
        x=x+1;
    end
    printf(”\n”);
    m=m+1;
end
printf(”————————————————————————————————————————————————\n”);
```

Scilab code Exa 6.2 Multiplication table

```
//                Example 6.2
//A program to print multiplication table from 1*1 to 12*10.


disp("                MULTIPLICATION TABLE                 ");
disp("-------------------------------------------------------");
row=1;
while(row<=12)             //Outer loop begins
    column=1;
    while(column<=10)      //Inner loop begins
        y=row*column;
        printf("%4d",y);
        column=column+1;
     end
     row=row+1;
     printf("\n");
end
```

Scilab code Exa 6.2cs Case study 2 Histogram

```
//                       Case Study:-Chapter 6
//                            2.Histogram

N=5;
for n=1:N
```

```
    printf("Enter employees in Group-%d:",n);
    value(n)=scanf("%d");            //Read data in the array named value
    printf("%d\n",value(n));         //Print number which is at position n
end
printf("\n");
printf("          |\n");
//Computation using for loop and draw a histogram
for n=1:N
    for i=1:3
         if(i==2) then
                printf("Group-%1d |",n);
          else
                printf("          |");
          end
          for j=1:value(n)
                printf("*");
          end
          if(i==2)
                printf("(%d)\n",value(n));
           else
                printf("\n");
           end
     end
                printf("          |\n");
end
```

Scilab code Exa 6.3 Uses a for loop

```
//              Example 6.3
//The program uses a for loop to print the "Power of 2" table for the
```

```
//power 0 to 20, both positive and negative.

disp("-------------------------------------------------");
printf(" 2 to power n        n       2 to power -n\n");
disp("-------------------------------------------------");
for n=0:20            //Loop begins
    p=2^n;
    q=2^-n;
    printf("%7d      %9d       %10.12f\n",p,n,q);
end                   //Loop ends
```

Scilab code Exa 6.3cs Case study 3 Minimum Cost

```
//                  Case Study:-Chapter 6
//                   3.Minimum Cost

for p=0:0.1:10
    cost=48-8*p+p^2;
    if(p==0) ,
        cost1=cost;
        continue;         //Use of continue statement
    end

    if(cost>=cost1) ,
        break;            //Use of break statement
    end
    cost1=cost;
    p1=p;
end
p =(p+p1)/2.0;
cost=40-8*p+p^2;        //Computes the cost
//print the result
```

```
printf("MINIMUM COST=%.2f AT p=%.1f\n",cost,p);
```

Scilab code Exa 6.4 Read the marks and print total marks

```
//                         Example 6.4
//A class of n students take an annual examination in m subjects.
// A program to read the marks obtained by each student in various subjects
// and to compare and print the total marks obtained by each of them.

FIRST=360;SECOND=240;
disp("Enter number of students and subjects");
[n,m]=scanf("%d %d");
for i=1:n
    roll_number=input("Enter roll_number:");
    total=0;
    printf("Enter marks of %d subjects for ROLL NO %d",m,roll_number);
    printf("[Enter each in newline]");
    for j=1:m
        marks=scanf("%d");
        total=total+marks;    //Compute the total
    end
    //print the total marks
    printf("TOTAL MARKS =%d",total);
    //Test  for division and display it
    if(total>=FIRST) ,
        disp("First Division");
    elseif(total>=SECOND)
        disp("Second Division");
    else
```

```
        disp("***F A I L ***")
    end
end
```

Scilab code Exa 6.4cs Case study 4 Plotting of two Functions

```
//                    Case Study:-Chapter 6
//    4.Plotting of two Functions i.e. y1=exp(-ax) and y2=exp(-ax^2/2)

a=0.4;
printf("                              y-------->                              \n");
printf("0 ----------------------------------------------------------------------\n");
for x=0:0.25:4
    //Evaluation of functions
    y1=int32(50*exp(-a*x)+0.5);
    y2=int32(50*exp(-a*(x^2)/2)+0.5);
    //plotting when y1=y2
    if(y1==y2) ,
        if(x==2.5) ,
            printf("x |");
        else
            printf("  |");
        end
        for i=1:(y1-1)
            printf(" ");
        end
        printf("#\n");
        continue;
    end

    //Plotting when y1>y2
    if(y1>y2) ,
        if(x==2.5) ,
            printf("x |");
        else
            printf("  |");
        end
        for i=1:y2-1
            printf(" ");
        end
        printf("*");
        for i=1:(y1-y2-1)
            printf("-");
        end
        printf("0\n");
        continue;
     end

     //Plotting when y2>y1
    if(y2>y1) ,
        if(x==2.5)
            printf("x |");
        else
            printf("  |");
        end
        for i=1:(y1-1)
            printf(" ");
        end
        printf("0");
        for i=1:(y2-y1-1)
            printf("-");
        end
        printf("*\n");
     end
end
     printf("  |\n");
```

Scilab code Exa 6.5 Use of the break statement

```
//                  Example 6.5
//Program illustrate use of the break statement

disp("This program computes the avarage of set of numbers");
disp("Enter values and enter a NEGATIVE value at the end");
sum1=0;
for m=1:1000
    x=scanf("%f");      //Read data
    if(x<0) then
        break;         //EXIT FROM LOOP
    end
    sum1=sum1+x;       //Computes sum
end
average=sum1/(m-1);   //Computes Average
//Print the results
printf("Number of values =%d\n",m-1);
printf("sum1=%f\n",sum1);
printf("Avarage =%f\n",average);
```

Scilab code Exa 6.6 Evaluate the series

```
//                  Example 6.6
//Program to evaluate the series i.e.
// 1/1-x = 1+x+x^2+x^3+.....+x^n
```

```

x=input("Input value of x:");    //Read value of x
LOOP=100; ACCURACY=0.0001;       //Initialization
sum1=0;term=1;flag=0;
//Computation using for loop
for n=1:LOOP
    sum1=sum1+term;
    if(term<=ACCURACY) then      //Test for accuracy
        flag=1;
        break;
    end
    term=term*x;

end
//Print the results
if(flag==1) then
    printf("  EXIT FROM LOOP\n");
    printf("  Sum =%f ; No. of terms =%d",sum1,n);
else
    disp("FINAL VALUE OF N IS NOT SUFFICIENT TO ACHIEVE DESIRED ACCURCY");
end
```

Scilab code Exa 6.7 Use of continue statement

```
//              Example 6.7
//The program illustrate the use of continue statement

disp("Enter 9999 to STOP");
count=0;
negative=0;
while(count<=100)
```

```
    number=input(”Enter a number:”);
    if(number==9999) then
        break;         //EXIT FROM THE LOOP
    end
    if(number<0),
        disp(”Number is negative”);
        negative =negative+1;
        continue;    //SKIP REST OF LOOP
     end
     sqrot=sqrt(number);  //COMPUTE SQUARE ROOT
     printf(”Number = %f\n”,number);
     printf(”Square root = %f”,sqrot);
     count=count+1;
end
//PRINT RESULTS
printf(”Number of items done = %d\n”,count);
printf(”Negative items = %d\n”,negative);
disp(”END OF DATA”);
```

Chapter 7

Arrays

Scilab code Exa 1.cs Case study 1 Median of list of numbers

```
//            Case Study: Chapter 7, Page No:210
//             1. Median of list of numbers

N=10;
disp("Enter the number of items");
n=scanf("%d");
//Reading items into array a
printf("Input %d values[One at a time]\n",n);
for i=1:n
    a(i)=scanf("%f");
end
//Sorting begins
for i=1:n-1
   for j=1:n-i
     if(a(j)<=a(j+1))
         t=a(j);
         a(j)=a(j+1);
         a(j+1)=t;
     else
         continue;
     end
```

```
 end
end  //sorting ends
//Calculation of median
if(n/2==0) then
    median1=(a(n/2)+a(n/2+1))/2.0;
else
    median1=a(n/2+1);
end
//Printing
for i=1:n
    printf("%f  ",a(i));
end
printf("\nMedian is %f\n",median1);
```

Scilab code Exa 2.cs Case study 2 Calculation of standard deviation

```
//              Case Study: Chapter-7
//        2.Calculation of standard deviation

MAXSIZE=100;
sum1=0;n=0;sumsqr=0;
disp("Input values:input -1 to end");
for i=1:MAXSIZE
    value(i)=scanf("%f");  //Entering values in the array named value
    if(value(i)==-1)
        break;
    end
    sum1=sum1+value(i);
    n=n+1;
end
mean1=sum1/n;                //Computes mean
for i=1:n
```

```
    deviation=value(i)-mean1;
    sumsqr=sumsqr+deviation^2;
end
variance1=sumsqr/n;
stddeviation=sqrt(variance1);       //Computes standard deviation
//Printing items,Mean and Standard deviation
printf("Number of items: %d\n",n);
printf("Mean: %f\n",mean1);
printf("Standard deviation: %f\n",stddeviation);
```

Scilab code Exa 3.cs Case study 3 Evaluating a Test

```
//                Case Study: Chapter-7
//                 3.Evaluating a Test

STUDENTS=3;
ITEMS=5;
//Reading of correct answers
printf("Input key to the items\n");
for i=1:ITEMS
    key(i)=read(%io(1),1,1,'(a)'); //Read data using read function
    // key(i)=scanf("%c"); It can be used to read data
end
//Evaluation begins

for student=1:STUDENTS
    //Reading students responses and counting correct ones
    count=0;
    printf("\nInput responses of student-%d",student
      );
    for i=1:ITEMS
        response(i)=scanf("%c");
    end
    correct=zeros(1,ITEMS)
//Commented code can be used to replace above line i.e. correct=zeros(1,ITEMS)
    //for i=1:ITEMS
    //    correct(i)=0;
    //end
    for i=1:ITEMS
        if(response(i)==key(i)) then
            count=count+1;
            correct(i)=1;
        end
    end
    //Printing of results
    printf("Student-%d\n",student);
    printf("Score is %d out of %d\n",count,ITEMS);
    printf("Response to the items below are wrong\n");
    n=0;
    for i=1:ITEMS
        if(correct(i)==0)
            printf("  %d",i);
            n=n+1;
        end
    end
    if(n==0) then
        printf("NIL\n");
    end
end
```

Scilab code Exa 4.cs Case study 4 Production and sales analysis

```
//                  Case Study: Chapter-7
//              4.Production and sales analysis

//Input Data
disp("Enter products manufactured week_wise");
disp("M11,M12,--,M21,M22,--etc");
for i=1:2
    for j=1:5
        M(i,j)=scanf("%d");
    end
end
disp("Enter products sold week_wise");
disp("S11,S12,--,S21,S22,--etc");
for i=1:2
    for j=1:5
        S(i,j)=scanf("%d");
    end
end
disp("Enter cost of each product");
for j=1:5
    C(j)=scanf("%d");
end
//Values matrices of production and sales
for i=1:2
    for j=1:5
        Mvalue(i,j)=M(i,j)*C(j);
        Svalue(i,j)=S(i,j)*C(j);
    end
end
//Total value of weekly production and sales
for i=1:2
    Mweek(i)=0;
    Sweek(i)=0;
    for j=1:5
        Mweek(i)=Mweek(i)+Mvalue(i,j);
        Sweek(i)=Sweek(i)+Svalue(i,j);
    end
end
//Monthly value of product_wise production and sales
for j=1:5
    Mproduct(j)=0;
    Sproduct(j)=0;
    for i=1:2
        Mproduct(j)=Mproduct(j)+Mvalue(i,j);
        Sproduct(j)=Sproduct(j)+Svalue(i,j);
    end
end
//Grand total of production and sales values
Mtotal=0;Stotal=0;
for i=1:2
    Mtotal=Mtotal+Mweek(i);
    Stotal=Stotal+Sweek(i);
end

//*********************************************
//Selection and printing of information required
//*********************************************
disp("Following is the list of things you request for");
disp("Enter appropriate number and press return key");

disp("1.Value matrices of production and sales");
disp("2.Total value of weekly production and sales");
disp("3.Production_wise monthly value of production and sales");
disp("4.Grand total value of production and sales");
disp("5.Exit")

number=0;
while(1)
    //Begining of while loop
    number=input("ENTER YOUR CHOICE:");
    if(number==5) then
        disp("Good Bye");
        break;
    end
    select number
        //Value Matices
    case 1 then
        disp("VALUE MATRIX OF PRODUCTION");
        for i=1:2
            printf("Week(%d)\t",i);
            for j=1:5
               printf("%7d",Mvalue(i,j));
            end
            printf("\n");
        end
       disp("VALUE MATRIX OF SALES");
        for i=1:2
           printf("Week(%d)\t",i);
            for j=1:5
               printf("%7d",Svalue(i,j));
            end
            printf("\n");
        end
       //Weekly Analysis
    case 2 then
        disp("TOTAL WEEKLY PRODUCTION AND SALES");
        disp("                PRODUCTION       SALES");
        disp("                ——————————      ———————");
        for i=1:2
            printf("Week(%d)\t",i);
            printf("%7d\t%9d\n",Mweek(i),Sweek(i));
        end
      //Product wise Analysis
    case 3 then
        disp("PRODUCTWISE TOTAL PRODUCTION AND SALES
           ");
        disp("                     PRODUCTION       SALES");
        disp("                     ——————————      ———————");
        for i=1:5
            printf("Product(%d)\t",i);
            printf("%7d\t%7d\n",Mproduct(i),Sproduct(i));
        end
     //Grand Totals
    case 4 then
       disp("GRAND TOTAL OF PRODUCTION AND SALES");
       printf(" Total production = %d\n",Mtotal);
       printf(" Total sales = %d\n",Stotal);
      //Default
    else
       printf("Wrong choicce,select again\n");
 end //End of select
end  //End of while
disp("Exit from the program");
```

Scilab code Exa 7.1 Sum of squares of 10 numbers

```
//                   Example:7.1
//Write a program using single-subscripted variable to evaluate:
// sum of squares of 10 numbers.The values x1,x2,... are read from the terminal.

//Reading values into array
disp("ENTER 10 REAL NUMBERS[Each in newline]");
total=0;
for i=1:10
    x(i)=input(" ");
    total=total+x(i)^2; //Computation of total
end
//Printing of x(i) values and total
```

```
for i=1:10
    printf("x(%2d) =%5.2f\n",i,x(i));
end
printf("Total =%.2f",total);
```

Scilab code Exa 7.2 Count the number of students

```
//                          Example:7.2
//Given below is the list of marks obtained by a class of 50 students in an
//annual examination. 43 65 51 27 79 11 56 61 82 09 25 36 07 49 55 63 74 81 49
//37 40 49 16 75 87 91 33 24 58 78 65 56 76 67 45 54 36 63 12 21 73 49 51 19 39
//49 68 93 85 59
//Write a program to count the number of students belonging to each of
// following groups of marks :0-9,10-19,20-29,.......100.

//This program computes for 10 students. We could compute for 50 students by
//changing MAXVAL=50.

MAXVAL=10;COUNTER=11;
disp("Input Data[Marks of 10 students]");
group1=zeros(1,11);
//Reading and counting
for i=1:MAXVAL
    //Reading of values
    value(i)=input(" ");
    //Counting frequency of groups
    a=int16((value(i)/10));
    if(a==0) then
      group1(a+1)=group1(a+1)+1;
     else
      group1(a+1)=group1(a+1)+1;
    end

end
//Printing of frequency table
printf("Group       Range     Frequency\n");
for i=0:COUNTER-1
    if(i==0) ,
        low=0;
    else
        low=i*10;
    end
    if(i==10),
        high=100;
     else
          high=low+9;
    end
    printf("%2d %8d to %3d %5d\n",i+1,low,high,group1(i+1));
end
```

Scilab code Exa 7.3 Compute and print

```
//                 Example:7.3
//Write a program using two dimensional array to compute print following
//information from the table of data discussed:
//(a)Total value of sales by each girl.
//(b)Total value of each item sold
//(c)Grand total of all sales of all items by all
```

```
   girls.

MAXGIRLS=4;MAXITEMS=3;
frequency=zeros(1,5);
disp("Input data");
//Reading values and computing girl_total
disp("Enter values,one at a time");

for i=1:MAXGIRLS
    girl_total(i)=0;
    for j=1:MAXITEMS
    value(i,j)=scanf("%d");
    girl_total(i)=girl_total(i)+value(i,j);
    end
end
//Computing item total
for j=1:MAXITEMS
    item_total(j)=0;
    for i=1:MAXGIRLS
    item_total(j)=item_total(j)+value(i,j);
    end
end
//Computing grand total
grand_total=0;
for i=1:MAXGIRLS
    grand_total=grand_total+girl_total(i);
end
//Printing of result
disp("GIRLS TOTALS");
for i=1:MAXGIRLS
    printf("Salesgirl(%d)=%d\n",i,girl_total(i));
end

disp("ITEM TOTALS");
for j=1:MAXITEMS
    printf("Item(%d)=%d\n",j,item_total(j));
end
printf("Grand Total=%d\n",grand_total);
```

Scilab code Exa 7.4 Multiplication table

```
//                              Example:7.4
//Write a program to compute and print a multiplication table for numbers 1 to 5

ROWS=5;COLUMNS=5;
disp("MULTIPLICATION TABLE");
printf(" *|")
for j=1:COLUMNS
    printf("%4d",j);
end
disp("________________________");
for i=1:ROWS
    printf("%1d|",i);
    for j=1:COLUMNS
        product(i,j)=i*j;           //Calculate the product
    printf("%4d",product(i,j));   //Print the product
    end
    printf("\n");
end
```

Scilab code Exa 7.5 Popularity of various cars

```
//                              Example:7.5
//A survey to know the popularity of four cars(Ambassador,fait,Dolphin and
//Maruti) was conducted in four cities(Bombay,Calcutta,Delhi and Madras).
//Each person surveyed was asked to give his city and type of car he was using.
//Write a program to prouce a table showing the popularity of various cars in
//four cities.

frequency=zeros(5,5);
printf("For each person,enter the city code[B,C,D,M]\n");
printf("followed by the car code[1,2,3,4].\n");
printf("Enter the letter X 0(zero)to indicate end.\n");

//Tabulation begins
for i=1:99
    [n,city,car]=mscanf("%c %d");
    if(city=='X') then
        break;
    end
    select city
     case 'B' then frequency(1,car)=frequency(1,car)+1;
     case 'C' then frequency(2,car)=frequency(2,car)+1;
     case 'D' then frequency(3,car)=frequency(3,car)+1;
     case 'M' then frequency(4,car)=frequency(4,car)+1;
    end

end

//Tabulation completed and Printing begins
disp("                 POPULATORY TABLE");
printf("----------------------------------------------\n"
);
printf("City        Ambasseador   fait   Dolphin   Maruti   \n");
printf("----------------------------------------------\n");

for i=1:4
    select i
    case 1 then printf(" Bombay   ");
    case 2 then printf(" Calcutta");
    case 3 then printf(" Delhi    ");
    case 4 then printf(" Madras   ");
    end
    for j=1:4
    printf("%8d",frequency(i,j));
    end
    printf("\n");
end
printf("----------------------------------------------");
//                Printing ends
```

Chapter 8

Character Arrays and Strings

Scilab code Exa 1.cs Case study 1 Counting words in a text

```
//                    Csae study: Chapter-8, Page No :253
//                          1.Counting words in a text

characters=0;words=0;lines1=0;
printf("      KEY IN THE TEXT.\n");
printf("GIVE ONE SPACE AFTER EACH WORD.\n");
printf("WHEN COMPLETED,ENTER end\n");

l=' ';
while(l~='end')
     l=read(%io(1),1,1,'(a)'); //Reading a line of text
     if(l=='end') then
        break;
     end
     line=[ascii(l)];          //Array of ascii values of line l
     len=length(line);         //compute length of line
     for i=1:len
```

```
        //ascii value of ' '(i.e.space) is 32
        if(line(i)==32) then
            words=words+1;      //Count the number of words
        end
    end
    lines1=lines1+1;            //Count the number of lines
    characters=characters+len; //Count the number of characters
end
//Printing results
printf("Number of lines = %d\n",lines1);
//Number of lines are added to words because last word of each line remains-
//uncounted as words are incremented at the occurence of space.
printf("Number of words = %d\n",words+lines1);
printf("Number of characters = %d\n",characters);
```

Scilab code Exa 2.cs Case study 2 Processing of a customer list

```
//                  Csae study: Chapter-8, Page No :253
//                   2.Processing of a customer list


CUSTOMERS=10;
printf("    Input names and telephone numbers\n");
printf(" [Names must have First,Second and Last_name ]\n");
for i=1:CUSTOMERS
    //Read data
    [first_name(i),second_name(i),surname(i),telephone(i)]=scanf(”%s %s %s %s”);
    //Converting full name to surname with initials
    l1=length(surname(i));            //Compute length of surname at i
    name(i)=strncpy(surname(i),l1) ;
    name(i)=strcat([name(i),’,’]);
    dummy(1)=part(first_name(i,1),1);
    name(i)=strcat([name(i),dummy]);
    name(i)=strcat([name(i),’.’]);
    dummy(1)=part(second_name(i,1),1);
    name(i)=strcat([name(i),dummy]);
end
//Alphabetical odering of surnames
for i=1:CUSTOMERS                 //Outer loop begins
    for j=2:CUSTOMERS-i+1         //Inner loop begins
        k=strcmp(name(j-1),name(j));
        if(k>0) then

            //Swaping names
            l1=length(name(j-1));
            l2=length(name(j));
            dummy=strncpy(name(j-1),l1);
            name(j-1)=strncpy(name(j),l2);
            l3=length(dummy);
            name(j)=strncpy(dummy,l3);

            //Swapping telephone numbers
            l3=length(telephone(j-1));
            l4=length(telephone(j));
            dummy=strncpy(telephone(j-1),l3);
            telephone(j-1)=strncpy(telephone(j),l4);
            telephone(j)=strncpy(dummy,l3);
        end
    end  //Inner loop ends
end  //Outer loop ends
```

```
//Printing alphabetical list
disp(”CUSTOMER LIST IN ALPHABETICAL ORDER”);
for i=1:CUSTOMERS
    printf(”%–20s\t %–10s\n”,name(i),telephone(i));
end
```

Scilab code Exa 8.1 Read a series of words

```
//                    Exampple 8.1
//Write a program to read a series of words from terminal using scanf function.

//Read data using scanf function
disp(”Enter text:”)
[word1,word2,word3,word4]=scanf(”%s %s %s %s”);
//Printing the results
printf(”word1 = %s\nword2 = %s\n”,word1,word2);
printf(”word3 = %s\nword4 = %s\n”,word3,word4);
```

Scilab code Exa 8.2 Read a line of text

```
//                    Example 8.2
//Write a program to read a line of text containing a series of–
//words from the terminal.

disp(”Enter text. Press <Return> at end”);
line=read(%io(1),1,1,’(a)’);      //Read a line
disp(line);                       //Display line
```

Scilab code Exa 8.3 Copy one string into another

```
//                          Example 8.3
//Write a program to copy one string into another and count the number
//of characters copied.

//Read data using scanf function
disp("Enter a string:")
[string2]=scanf("%s");        //Read string
l=length(string2);            //Compute the length
string1=' ';                  //string1 is empty
for i=1:l
    string1=string1+ part(string2,i);
end
//Printing the results
printf(" %s\n",string1);
printf(" Number of characters = %d\n",l);
```

Scilab code Exa 8.4 Display the string under various format specifications

```
//                          Exampple 8.4
//Write a program to store the string "United Kingdom" in the array country-
//and display the string under various format specifications.

```

```

country='United Kingdom';
printf("\n");
printf("*123456789012345*\n");
printf("--------\n");
printf("%15s\n",country);
printf("%5s\n",country);
printf("%15.6s\n",country);
printf("%-15.6s\n",country);
printf("%15.0s\n",country);
printf("%.3s\n",country);
printf("%s\n",country);
printf("--------\n");
```

Scilab code Exa 8.5 Program using for loop

```
//                    Example 8.5
//Write a program using for loop to print the following output:
//    C
//    CP
//    ....
//    ....
//    CProgrammimg
//    CProgrammimg
//    ....
//    ....
//    CPr
//    CP
//    C
string1='CProgramming';
printf(" -------------\n");
f=' ';
```

```
for i=1:12
    f=f+part(string1,i);

    printf("|%-13s|\n",f);
end
printf("|-------------|\n");
for j=0:11
    s=' ';
    for i=1:12-j
        s=s+part(string1,i);
     end
     printf("|%-13s|\n",s);
end
printf(" -------------");
//for c=0:11
//     d=c+1;
//     mprintf("|%-12.*s|\n",d,string1);
//end
//disp("----------------------");
//for c=11:c-1:0
//     d=c+1;
//     printf("|%-12.*s|\n",d,string1);
//end
//disp("-----------------------");
```

Scilab code Exa 8.6 Print the alphabet set a to z and A to Z

```
//                  Example 8.6
//Write a program which would print the alphabet set
    a to z A to Z in decimal-
//character form.

for c=65:122
```

```
    if(c>90&c<97) then
        continue;           //Terminate current iteration
    end
    c1=ascii(c);            //Convert ascii value to character
    printf("|%4d  -  %c\",c,c1);
end
printf("|\n")
```

Scilab code Exa 8.7 Concatinate the three parts into one string

```
//                         Example 8.7
//The name of employees of an organization are stored in three arrays namely-
//first_name,second_name and last_name.
//Write a program to concatinate the three parts into one string called name.


//Store the name in the three arrays
first_name=['VISWANATH'];
second_name=['PRATAP'];
last_name=['SINGH'];

//Concatinate three parts into one
name=[first_name second_name last_name];
// Print the result
for i=1:3
    printf("%s ",name(i));
end
//Statement below can also be used to print the result
```

```
//disp(name);
```

Scilab code Exa 8.8 Compare whether strings are equal

```
//                                Example 8.8
//s1,s2 and s3 are three string variables. Write a program to read two string-
//constants in to s1 and s2 and compare whether they are equal or not,join-
//them together. Then copy contents of s1 to variable s3. At the end program-
// should print all three variables and their lengths


//Read data
printf("Enter two string constants\n");
[s1 s2]=scanf("%s %s");

//Comparing two strings
x=strcmp(s1,s2);
if x~=0 then
    printf("String are not equal\n");
    //Concatinate two strings s1 and s2
    s1=strcat([s1,s2]);
else
     printf("String are equal\n");
end

l1=length(s1);
//Coping s1 to s3
s3=strncpy(s1,l1);
//finding length of strings
```

```
l2=length(s2);
l3=length(s3);
//Output
printf("s1 =  %s\t length = %d characters\n",s1,l1);
printf("s2=  %s\t length = %d characters\n",s2,l2);
printf("s3=  %s\t length = %d characters\n",s3,l3);
```

Scilab code Exa 8.9 Sort a list of names in alphabetical order

```
//                              Example 8.9
//Write a program that would sort a list of names in alphabetical order.

ITEMS=5;
//Reading the list
printf("Enter names of %d items\n",ITEMS);
i=1;
while(i<=ITEMS)
    string1(i)=scanf("%s");
    i=i+1;
end
//Sorting begins
for i=1:ITEMS                   //Outer loop begins
    for j=2:ITEMS-i+1           //Inner loop begins
        k=strcmp(string1(j-1),string1(j))
        if(k>0) then
            //Compute length and Exchange of contents
            l1=length(string1(j-1));
            l2=length(string1(j));
            dummy=strncpy(string1(j-1),l1);
            string1(j-1)=strncpy(string1(j),l2);
            l3=length(dummy);
            string1(j)=strncpy(dummy,13);
        end
    end  //Inner loop ends
end  //Outer loop ends
//Sorting completed
disp(”Alphabetical list”);
for i=1:ITEMS
    printf(”%s\n”,string1(i));
end
```

Chapter 9

User Defined Functions

Scilab code Exa 1.cs Case study 1 Calculation of Area under a Curve

```
//            Case Study:Chapter-9[page no:310]
//         1.Calculation of Area under a Curve
funcprot(0);
 //global variables
 global start_point
 global end_point;
 global total_area;
 global numtraps;
function[]=input1()
    global start_point;
     global end_point;
     global total_area;
     global numtraps;
    total_area=0;
    start_point=input("Enter lower limit:");
    end_point=input("Enter upper limit:");
    numtraps=input("Enter number of trapezoids:");
endfunction
function[total_area]=find_area(a,b,n)
    global total_area;
    base=(b-1)/n;                 //base is local
```

```
variable
    lower=a;                              //lower is local variable
    for lower=a:(lower+base):(b-base)
        h1=function_x(lower);
        h2=function_x(lower+base);
        total_area=total_area+trap_area(h1,h2,base);    //total area is calculated
    end
endfunction
function[area]=trap_area(height_1,height_2,base)
    area =0.5*(height_1+height_2)*base;      //area is local variable
endfunction
function[x] =function_x(x)
    x=(x^2)+1;
endfunction

//calling functions
disp(”AREA UNDER CURVE”);
input1();                                 //calling input1() function
total_area=find_area(start_point,end_point,numtraps);//calling find_area() function
printf(”TOTAL AREA = %f”,total_area);
```

Scilab code Exa 9.1 Multiple functions

```
//                        Example 9.1
//Write a program with multiple functions that do not communicate-
//data between them.

```

```
//Function1:printline()
funcprot(0);
function []=printline()     //contains no argument
    for i=1:35
        printf("%c",'-');
    end
    printf("\n");
endfunction
//Function2: value()
function []=value()         //contains no argument
    principal=input("Principal amount?");
    inrate=input("Interest rate?");
    period=input("Period?");
    sum1=principal;
    year=1;
    //Computation using while loop
    while(year<=period)
        sum1=sum1*(1+inrate);  //calculates principal amount after certain years
        year=year+1;
     end
     printf("%8.2f  %5.2f %5d %12.2f\n",principal,inrate,period,sum1);
endfunction
//Calling  functions
printline();
value();
printline();
```

Scilab code Exa 9.2 Include arguments in function calls

```
//                Example 9.2
//Modify Example 9.1 to include arguments in
```

```
function calls.

funcprot(0);
function []=printline(ch)          //function with argument ch
  for i=1:52
      printf("%c",ch);
  end
  printf("\n");
endfunction
function []=value(p,r,n)          //function with argument p,r,n
  sum1=p;
  year=1;
  while(year<=n)
      sum1=sum1*(1+r);
      year=year+1;
   end
   printf("%f\t%f\t%d\t%f\n",p,r,n,sum1);
endfunction
printf("Enter principal amount,interest rate,and period \n[Enter in single line seperated by space]");
[principal,inrate,period]=scanf("%f %f %d");   //read from standard input
//Calling functions
printline('z');
value(principal,inrate,period);
printline('c');
```

Scilab code Exa 9.3 Return result

```
//                    Example 9.3
```

```
//Modify Example 9.2's function value() to return result and extend versatility of the function
//printline by taking the lenth of line as an argument
function []=printline(ch,len)
    for i=1:len
        printf("%c",ch);
    end
    printf("\n");
endfunction
function [amount]=value(p,r,n)   //returns amount
    sum1=p;
    year=1;
    while(year<=n)
        sum1=sum1*(1+r);
        year=year+1;
     end
     amount=sum1;
endfunction
printf("Enter principal amount,interest rate,and period\n[Enter in single line seperated by space]");
[principal,inrate,period]=scanf("%f %f %d");
//Calling  functions
printline('*',52);
amount=value(principal,inrate,period);
printf("%f\t%f\t%d\t%f\n",principal,inrate,period,amount);
printline('=',52);
```

Scilab code Exa 9.4 Computes x raised to the power y

```
//                    Example 9.4
```

```
//Write a program with function power that computes x raised to the power y.
funcprot(0);
function p = power(x,y)
     p=1.0;               // x to power 0
    if((y>=0)) then
        while(y)         //computes positive powers
            p=p*x;
            y=y-1;
        end
    else
        while(y)       //computes negative powers
            p=p/x;
            y=y+1;
        end
    end
endfunction
disp("Enter x,y:");
disp("[Enter in single line seperated by space] ");
[x,y]=scanf("%d %d");    //input using scanf function

//calling power() function and printing its output
printf("%d to the power %d is %f",x,y,power(x,y));
```

Scilab code Exa 9.5 Calculate standard deviation of an array values

```
//             Example 9.5
//Write a program to calculate standard deviation of an array values.
//Array elements are read from terminal.Use functions to calulate-
//standard deviation and mean
```

```
funcprot(0);
//passing array named 'value' to function std_dev at 'a'
function[std]=std_dev(a,n)
    sum1=0;
    x=mean1(a,n);                          //calling mean1() function
    for i=1:n
      sum1=sum1+(x-a(i))^2;
      std=sqrt(sum1/double(n));            //computes standard deviation
    end
endfunction
function[x]=mean1(a,n)
    sum1=0;
    for i=1:n
      sum1=sum1+a(i);
      x=sum1/double(n);                    //x contain mean value
    end
endfunction
SIZE=int8(5);                              //size of array
printf("Enter %d float values",SIZE);
for i=1:SIZE
    value(i)=input(" ");                   //entering values in the array
end
printf("Std.deviation is %f",std_dev(value,SIZE));  //calling std_dev() function
```

Scilab code Exa 9.6 Sort an array

```
//             Example 9.6
//Write a program that uses a function to sort an array of integers.
funcprot(0);
function[x]=sort(m,x)           //Passing an array i.e. marks to function sort()
    for i=1:m                   // i repesents number of passes
        for j=2:m-i+1           // j represents number of comperision in each pass
            if(x(j-1)>=x(j)) then
                t=x(j-1);
                x(j-1)=x(j);
                x(j)=t;
            end
        end
    end
endfunction
marks=int16([40,90,73,81,35]);   //creating an array named marks of 5 integers
disp("Marks before sorting");
disp(marks);
x=sort(5,marks);                 //calling sort() function
disp("Marks after sorting");
disp(x);
```

Scilab code Exa 9.7 Autometic variables

```
//             Example 9.7
//Write a multifunction how autometic variables work.
funcprot(0);
```

```
function[]=function1()
    m=int32(10);            //Local Variable
    disp(m);                //First Output
endfunction
function[]=function2()
    m= int32(100);          //Local Variable
    function1();            //Calling function1()
    disp(m);                //Second Output
endfunction
function[]=funcmain()
m=int32(1000);
function2();        //calling function2()
disp(m);            //Third output
endfunction
funcmain()          //calling funcmain() function
```

Scilab code Exa 9.8 Global variables

```
//              Example 9.8
//Write a multifunction to illustrate the properties of global variables.
funcprot(0);
function[x]=fun1()
    global x;
    x=x+10;             //global x
endfunction
function[x]=fun2()
    x=1                //Local x
endfunction
function[x]=fun3()
    global x;
    x=x+10;             //global x
endfunction
```

```
    global x;
    x=10;
    printf("x=%d\n",x)
    //calling fun1(),fun2(),fun3() functions
    printf("x=%d\n",fun1());
    printf("x=%d\n",fun2());
    printf("x=%d\n",fun3());
```

Scilab code Exa 9.16 Factorial of a number using recursion

```
//            Topic 9.16 RECURSION
//            Page no. 288
//Write a program to calculate factorial of a number using recursion
function[fact1]=factorial1(n)
    fact1=-1
    if(n<0) then
        disp("Please enter positive value[i.e. 0 or greater than 0]  ");
        return;                  //Quits the current function
    end
    if((n==0)|(n==1)) then
        fact1=1;
    else
        fact1=n*factorial1(n-1);   //recursive call to factorial1()
    end
endfunction
n=input("Enter number:");
//calling factorial1() function inside printf()
printf("Factorial of %d  = %d",n,factorial1(n));
```

Chapter 10

Structures and Unions

Scilab code Exa 1.cs Case study 1 Book Shop Inventory

```
//                  Case study: Chapter-10, Page No :341
//                       Book Shop Inventory

funcprot(0);
//Defining functions
function [string1]=get1()
    string1=read(%io(1),1,1,'(a)');
endfunction
function [i] =look_up(table,s1,s2,m)
    for i=1:m
        x=strcmp(s1,table(i).title);
        y=strcmp(s2,table(i).author);
        if x==0 & y==0 then
            return i;   //Book found
        end
     end
      i=-1;  //Book not found
endfunction

//Creates array of structures
```

```
book=[struct('author','Ritche','title','C Language','price',45.00,'month','May','year',1977,'publisher','PHI','quantity',10)
      struct('author','Kochan','title','Programming in C','price',75.50,'month','July','year',1983,'publisher','Hayden','quantity',5)
      struct('author','Balagurusamy','title','BASIC','price',30.00,'month','January','year',1984,'publisher','TMH','quantity',0)
      struct('author','Balagurusamy','title','COBOL','price',60.00,'month','December','year',1988,'publisher','Macmillan','quantity',25)
         ];
n=size(book);
no_of_records=n(1);
response=' ', a=1;
while ((response=='Y' | response=='y')|a==1)
    //Read data
    printf("Enter title and author name as per the list:\n");
    printf("Title: \n");
    title1=get1();
    printf("Author:\n");
    author1=get1();
    //Calling index() function and
    //Passing structure book to function look_up()
    index=look_up(book,title1,author1,no_of_records);
    //If book found then print the book detail otherwise not in list
    if index~=-1 & index then    //Book found
      printf("%s  %s",book(index).author, book(index).title);
      printf("% .2f  %s",book(index).price, book(index).month);
      printf("% d  %s\n",book(index).year, book(index).publisher);
```

```
        quantity=input("Enter number of copies :");
        if quantity<book(index).quantity then
         printf("Cost of %d copies = %.2f\n",quantity,book(index).price*quantity);
        else
           printf("Required copies not in stock\n");
        end
      else
           printf("Book not in list\n");
      end
      printf("\nDo you want any other book?(YES/NO):");
      response=get1();a=2;
end
printf("   Thank you.     Good Bye");
```

Scilab code Exa 10.1 Define a structure type

```
//                    Example 10.1
//Define a structure type,struct personal that would contain person name,-
// date of joining and salary. Write a program to read this information from
// keyboard and print same on the screen.

funcprot(0);
function [ ]=struc(n,d,m,y,s)
    //Defining structure members
    personal=struct('name',n,'day',d,'month',m,'year',y,'salary',s);
    person=personal;
   //Accessing structure members
printf("   %s %d %s %d %.2f",person.name,person.day,
    person.month,person.year,person.salary);
endfunction
disp("Input values[Name day month year and salary]");
//Reading data
[name,day,month,year,salary]=scanf("%s %d %s %d %f");
//Calling function struc()
struc(name,day,month,year,salary);
```

Scilab code Exa 10.2 Comparison of structure variables

```
//                    Example 10.2
// Write a program to illustrate the comparison of structure variables.

function []=class()
    //Defining structures
    student1=struct('number',111,'name','Rao','marks',72.50);
    student2=struct('number',222,'name','Raddy','marks',67.00);
    student3=struct('number',[],'name',[],'marks',[]);
    student3=student2;
    if(student3==student2) ,   //Logical operation on structures
        disp("Student2 and student 3 are same");
        printf(" %d %s %f",student3.number,student3.name,student3.marks);
    else
        disp("Student2 and student 3 are not same");
    end
```

```
endfunction
//calling function class
class()
```

Scilab code Exa 10.3 Calculate the subject wise and student wise totals

```
//                          Example 10.3
// Write a program to calculate the subject-wise and student-wise totals
//and store them as a part of the structue.

//Defining array of structures
student=[struct('sub1',45,'sub2',67,'sub3',81,'total',0)
         struct('sub1',75,'sub2',53,'sub3',69,'total',0)
         struct('sub1',57,'sub2',36,'sub3',71,'total',0)
         ];
total=struct('sub1',0,'sub2',0,'sub3',0,'total',0);

//Calculate the student-wise and subject-wise totals
for i=1:3
    student(i).total=student(i).sub1+student(i).sub2+student(i).sub3;
    total.sub1=total.sub1+student(i).sub1;
    total.sub2=total.sub2+student(i).sub2;
    total.sub3=total.sub3+student(i).sub3;
    total.total=total.total+student(i).total;
end
//Printing student-wise totals
printf("STUDENT          TOTAL\n");
for i=1:3
```

```
    printf("student(%d)          %d\n",i,student(i).total);
end
//Printing subject-wise totals
printf("SUBJECT          TOTAL\n");
printf(" %s        %d\n","Subject 1",total.sub1);
printf(" %s        %d\n","Subject 2",total.sub2);
printf(" %s        %d\n","Subject 3",total.sub3);
//Printing grand total
printf("Grand Total = %d",total.total);
```

Scilab code Exa 10.4 Array member to represent the three subjects

```
//                          Example 10.4
//Rewrite the  program of Example 10.3 to using an array member to represent
//the three subjects.

//Defining array of structures and array with in structure
student(1)=[struct('sub',[45 67 81],'total',0)];
student(2)=[struct('sub',[75 53 69],'total',0)];
student(3)=[struct('sub',[57 36 71],'total',0)];
total=student;
for i=1:3
    total.sub(i)=0;
end
total.total=0;
//Calculate the student-wise and subject-wise totals
for i=1:3
    for j=1:3
        student(i).total=student(i).total+student(i).sub(j);
        total.sub(j)=total.sub(j)+student(i).sub(j);
    end
    total.total=total.total+student(i).total;  //Grand  total
end
//Printing student-wise totals
printf("STUDENT          TOTAL\n");
for i=1:3
    printf("student(%d)          %d\n",i,student(i).total);
end
//Printing subject-wise totals
printf("SUBJECT          TOTAL\n");
for j=1:3
    printf("subject -(%d)          %d\n",j,total.sub(j));
end
//Printing grand total
printf("Grand Total = %d",total.total);
```

Scilab code Exa 10.5 structure as a parameter to a function

```
//                    Example 10.5
//Write a simple program to illustrate the method of sending an entire
//structure as a parameter to a function.

funcprot(0);
//Defining functions
function [item]=update(product,p,q)
    product.price=product.price+p;
    product.quantity=product.quantity+q;
    item=product;
```

```
endfunction
function [value] =mul(stock)
     value=stock.price*stock.quantity;
endfunction

//Creates structure item
item=struct('name','XYZ','price',25.75,'quantity',12);
//Read data
printf("Input increment values:");
printf("  price increment and quantity increment\n");
[p_increment,q_increment]=scanf("%f %d");

//Calling update() and mul() functions
//Passing structure item to functions update() and mul()
//----------------------------------------------------
item=update(item,p_increment,q_increment);
value=mul(item);
//----------------------------------------------------
//Printing Results
printf("Updated values of items\n");
printf("Name          :%s\n",item.name);
printf("Price         :%f\n",item.price);
printf("Quantity      :%d\n",item.quantity);
printf("Value of item = %f\n",value);
```

Chapter 12

File Management in C

Scilab code Exa 12.1 Read data from keyboard and write it to a file

```
//                          Example 12.1
// Write a program to read data from keyboard,write it to a file called INPUT,
// again read the same data from the INPUT file and display it on the screen.

warning('off');
disp("Data Input");

//Open the file INPUT
f1=mopen('INPUT.txt','w');
// Get character from keyboard
c=read(%io(1),1,1,'(a)');
mfprintf(f1,'%s',c);

//close the file input
mclose(f1);


disp("Data Output");
//Reopen the file INPUT
```

```
f1=mopen('INPUT.txt','r');
     txt=mgetl(f1);
     printf("%s",text);
//close the file input
mclose(f1);
```

Scilab code Exa 12.2 Read and write odd and even numbers

```
//                      Example12.2
//A  file named DATA contains a series of integer numbers. Code a program
//to read these numbers and then write all 'odd' numbers to a file to be
//called ODD and all 'even' numbers to a file to be called EVEN.

warning('off');
//Input numbers in the DATA.txt file
printf("Contents of DATA file\n");
f1=mopen('DATA.txt','wt');
for i=1:30
    number(i)=scanf("%d");
    if(number(i) == -1)
        break;
    end
     mfprintf(f1,'%d\n',number(i));
end
mclose(f1);

f2=mopen('ODD.txt','wt');
f3=mopen('EVEN.txt','wt');
f1=mopen('DATA.txt','rt');
//Read numbers from DATA file
```

```
EOF=length(number);
i=1;
even=0;
odd=0;
while (i<EOF)
    [n,number]=mfscanf(f1,” %d”)
     if(pmodulo(number,2)==0)
        mfprintf(f3,’%d\n’,number);
        even=even+1;
    else
        mfprintf(f2,’%d\n’,number);
        odd=odd+1;
    end
    i=i+1;
end
mclose(f1);
mclose(f2);
mclose(f3);
//Write odd numbers in the ODD.txt file
f2=mopen(’ODD.txt’,’rt’);
printf(”\nContents of ODD file\n”);
i=1;
while (i<=odd)
    [n,number]=mfscanf(f2,”%d”)
    printf(”%4d”,number);
    i=i+1;
end
//Write even numbers in the EVEN.txt file
f3=mopen(’EVEN.txt’,’rt’);
printf(”\nContents of EVEN file\n”);
i=1;
while (i<=even)
    [n,number]=mfscanf(f3,”%d”)
    printf(”%4d”,number);
    i=i+1;
end
//close the files
mclose(f2);
mclose(f3);
```

Scilab code Exa 12.3 Read and write data to and from the file INVENTORY

```
//                              Example12.3
//A  program to open a file named INVENTORY and store in it the following
// data: Item name    Number      Price     Quantity
//          AAA-1       111        17.50     115
//          BBB-2       125        36.00     75
//          C-3         247        31.75     104
//Extend the program to read this data from the file INVENTORY and display
//inventory table with the value of each item.

disp("Input file name");
filename=scanf("%s");    //Read file name that is 'INVENTORY'
fp=mopen(filename,'w');  //Open file in write mode, fp is file descriptor
disp("Input inventory data");
disp("Item name      Number      Price      Quantity");
for i=1:3
     //read data from terminal
  [n,item(i),number(i),price(i),quantity(i)]=mscanf("%s %d %f %d");
  //write data to the file
   mfprintf(fp,'%s\t%d\t%.2f\t%d\n',item(i),number(i),price(i),quantity(i));
end
mclose(fp);                  //close the file
fp=mopen(filename,'r');  //open file in read mode
```

```
disp("Item name    Number    Price    Quantity    Value");
for i=1:3
    //Read data from the file 'INVENTORY'
    [n,item,number,price,quantity]=mfscanf(fp,"%s %d %f %d");
     value=price*quantity;  //Computes value
     //Printing of the data
     printf('   %s    %7d %8.2f  %8d %11.2f\n',item,number,price,quantity,value);
end
mclose(fp);
```

Scilab code Exa 12.4 Error handling in file operations

```
//                      Example12.4
//Write a program to illustatre error handling in file operations.

warning('off');
fp1=mopen('TEST','w');   //Open file in write mode, fp1 is file descriptor
for i=10:10:100
   //write data to the file
   mfprintf(fp1,'%d\n',i);
end
mclose(fp1);
disp("Input file name");
filename='a';
while(filename~=' ')
    filename=scanf("%s");
    //Error handling
    try
```

```
        fp2=mopen(filename,'r');
        if(fp2>0) ,
            break;         //Terminates the loop if file exist or opened
        end

    catch
        //Messages to be displayed when error occured
        printf("Can not open file.\n");
        printf("Type file name again.\n");
    end
end
//Code below runs while there is no error
for i=1:20
    number = mfscanf(fp2,"%d");  //Read data from file 'TEST'
    if meof(fp2) then            //Test for end of file
       printf("Ran out of data");
       break;
     else
        printf("%d\n",number);   //prints the data
      end
end
mclose(fp2);
```

Scilab code Exa 12.5 use of function ftell or mtell and fseek or mseek

```
//                    Example12.5
//Write a program that uses function ftell(mtell) and fseek(mseek).

```

```
warning('off');
//Open file 'RANDOM' in write mode,fp is file descriptor
fp=mopen('RANDOM','w');
c=read(%io(1),1,1,'(a)');
mfprintf(fp,'%s',c);    //write data to the file
printf("Number of characters entered = %d\n",mtell(fp));
mclose(fp);

//Open file 'RANDOM' in read mode
fp=mopen('RANDOM','r');
n=0;
while(meof(fp)==0)
//n is the offset from origin in number of bytes.
//The new position is at the signed distance given by n bytes from the beginning
    mseek(n,fp,'set');
    //Print the chracter  and its postion
    printf("Position of %c is %d\n",ascii(mget(1,'c',fp)),mtell(fp));
    n=n+5;
end
n=0;     //Initial offset
while(mtell(fp)>1)
 //New position is at the signed distance given by n bytes from the end
      mseek(n,fp,'end');
      //Print the characters from the end
      printf("%c",(ascii(mget(1,'c',fp))));
      n=n-1;
end
mclose(fp);
```

Scilab code Exa 12.6 Append additional items to the file INVENTORY

```
//                          Example 12.6
//Write a program to append additional items to the file INVENTORY
//created in Example 12.3 and print the total contents of the file.
funcprot(0);
warning('off');
function[item] =append(product,fp)
    printf("Item name:\n");
    product.name=scanf("%s");
    printf("Item number:.\n");
    product.number=scanf("%d");
    printf("Item price\n");
    product.price=scanf("%f");
    printf("Quantity:\n");
    product.quantity=scanf("%d");
    //Write data in the file
     mfprintf(fp,'%s %d %.2f %d\n',product.name,product.number,product.price,product.quantity);
     item=product;
endfunction
//Creating structure
item=struct('name','0','number','0','price','0','quantity','0');
//Read file name that is 'INVENTORY'
disp("Type file name");
filename=scanf("%s");
//Open file in append mode,fp is file descriptor
fp=mopen(filename,'a+');
b=0;response=-1;
//Read data
while(response==1|b==0)
    item=append(item,fp);  //calling append() function
    printf("Item %s appended.\n",item.name);
    printf(”Do you want to add another item\(1 for YES/0 for NO)?”);
    response=scanf(”%d”);
    b=1;
end
n=mtell(fp);      //position of last character
mclose(fp);

//Opening file in the read mode
fp=mopen(filename,’r’);
while (mtell(fp) < n-2)
  //read data from terminal
  [g,item.name,item.number,item.price,item.quantity]=mfscanf(fp,”%s %d %f %d”);
  //Print Data to screen
   printf(’%s %7d %8.2f %8d\n’,item.name,item.number,item.price,item.quantity);
end
mclose(fp);
```

Chapter 13

Dynamic Memory Allocation and linked Lists

Scilab code Exa 1.cs Case study 1 Insertion in a sorted list

```
//                  Case Study: Chapter:13 ,Page No.:434
//                     1.Insertion in a sorted list

funcprot(0);
//Create the list
function [List]=create(list1)
    global List;
    // Create the current node
    list1.number=input("Input a number(Type -999 to end); ");
    if list1.number==-999 then
        list1.next=NULL;
        list1.add=NULL;
     else
         list1.add=list1.add+1;
         list1.next=NULL;
         List(i)=list1;
         if(i==1) then

         else
            List(i-1).next=List(i).add
         end
         i=i+1;
         create(list1);// Create the next node
     end
     return;
endfunction
function []=print1(list1)
     if list1(i)(1).next~=NULL then
        printf(”%d--->”,list1(i)(1).number);//Print current item
        i=i+1;
        if list1(i)(1).next==NULL then
             printf(”%d”,list1(i)(1).number);
        end
        print1(list1);//Move to next item
     end
     return;
endfunction
function [List]=insert(list1)
    global List;
    x=input(”Input number to be inserted: ”);//Read the number
    //find the location so that number could be placed in sorted order
    while (list1(i)(1).next~=NULL)
        if(list1(i)(1).number>=x) then
              break;
        end
        i=i+1;
    end
     key=i;
     //Insetion at end
    if(list1(i)(1).next==NULL & list1(i)(1).number < x) then
        list1(i+1)(1).number=x;
        list1(i+1)(1).add=i+1;
        list1(i+1)(1).next=NULL;
        list1(i)(1).next=list1(i+1)(1).add;
        List=list1;
        return;
    end
    i=1;
    while (list1(i)(1).next~=NULL)
        i=i+1;
    end
    j=i+1;
    //Key node found and insert new node or item
    while(list1(i)(1).add~=key)
        list1(i+1)(1).number=list1(i)(1).number;
        i=i-1;
    end
    list1(i+1)(1).number=list1(i)(1).number
    list1(i)(1).number=x;
    list1(j)(1).add=j;
    list1(j)(1).next=NULL;
    list1(j-1)(1).next=list1(j)(1).add;
    List=list1;
endfunction

global List;
NULL=0;i=1;
//Create the structure i.e. node
node=struct('number',0,'add',0,'next',0);
head=node;
//Calling the functions
printf("Input a sorted(ascending) list");
List=create(head);
printf("\nOriginal List: ");
print1(List);
List=insert(List);
printf("\nNew List: ");
print1(List);
```

Scilab code Exa 2.cs Case study 2 Building a Sorted List

```
//                  Case Study: Chapter:13 ,Page No.:438
//                     2.Building a Sorted List

funcprot(0);
//Create the list
function [List]=create(list1)
    global List;
    // Create the current node
    list1.number=input("Input a number(Type -999 to end); ");
    if list1.number==-999 then
        list1.next=NULL;
        list1.add=NULL;
     else
         list1.add=list1.add+1;
         list1.next=NULL;
         List(i)=list1;
         if(i==1) then

         else
            List(i-1).next=List(i).add
         end
         i=i+1;
         create(list1);// Create the next node
     end
     return;
endfunction
function []=print1(list1)
     if list1(i)(1).next~=NULL then
```

```
        printf("%d--->",list1(i)(1).number);//Print current item
        i=i+1;
        if list1(i)(1).next==NULL then
             printf("%d",list1(i)(1).number);
        end
        print1(list1);//Move to next item
     end
     return;
endfunction
//Sorting of the numbers in the list
function [List]=insert_sort(list1)
    global List;
    j=1;
    while (list1(j)(1).next~=NULL)
        i=1;
        while (list1(i)(1).next~=NULL)
           if(list1(i)(1).number >list1(i+1)(1).number) then
              temp=list1(i)(1).number;
              list1(i)(1).number=list1(i+1)(1).number;
              list1(i+1)(1).number=temp;
           end
           i=i+1;
        end
        j=j+1;
    end
    List=list1;
endfunction

global List;
NULL=0;i=1;
//Create the structure i.e. node
node=struct('number',0,'add',0,'next',0);
head=node;
//Calling the functions
List=create(head);
```

```
printf(”\nOriginal List: ”);
print1(List);
List=insert_sort(List); //Sort the list
printf(”\nAfter sorting: ”);
print1(List);
```

Scilab code Exa 13.3 Create a linear linked list

```
//                    Example 13.3
//Write a program to create a linear linked list interactively
//and print the list and total number of items in the list.

funcprot(0);
NULL=0;i=1;
//Create the list
function [List]=create(list1)
    global List;
   //Create the current node in the list
    list1.number=input(”Input a number(Type −999 to end); ”)
    if list1.number==-999 then
        list1.next=NULL;
        list1.add=NULL;
     else
         //Create the next node in the list
         list1.add=list1.add+1;
         list1.next=NULL;
         List(i)=list1;
         if(i==1) then

         else
```

```
             List(i-1).next=List(i).add
         end
         i=i+1;
       create(list1); //Call create() function
     end
     return;
endfunction
//Function to print the numbers of list
function []=print1(list1)
     if list1(i)(1).next~=NULL then
        printf(”%d——>”,list1(i)(1).number);//Print current item
        i=i+1;
        if list1(i)(1).next==NULL then
             printf(”%d”,list1(i)(1).number);
        end
        print1(list1);//Move to next item
     end
     return;
endfunction
//Function to count the number of items in the list
function []=count(list1)
    global c;
     if list1(i)(1).next==NULL then
         return;
     else
        i=i+1;
        c=i;
        count(list1);
     end
     return;
endfunction
//Create the structure i.e. node
node=struct('number',0,'add',0,'next',0);
head=node;
global List;
//Calling the functions
List=create(head);
```

```
print1(List);
global c;
c=1;
count(List);
//Print the total number of items
printf("\nNumber of items = %d",c);
```

Scilab code Exa 13.4 Insert the item before the specified key node

```
//                    Example 13.4
//Write a function to insert a given item before a specified node known as
//key node.

funcprot(0);
//Create the list
function [List]=create(list1)
    global List;
   // Create the current node
    list1.number=input("Input a number(Type -999 to end); ");
    if list1.number==-999 then
        list1.next=NULL;
        list1.add=NULL;
     else
         list1.add=list1.add+1;
         list1.next=NULL;
         List(i)=list1;
         if(i==1) then

         else
              List(i-1).next=List(i).add
         end
         i=i+1;
       create(list1);// Creates the next node
     end
     return;
endfunction
//Function to insert the item before the specified key node
function [List]=insert(list1)
    x=input("Value of new item?");
    printf("Value of key item?(Before which you want to insert?)");
    key=scanf("%d");
    while list1(i)(1).next~=NULL
            i=i+1;
     end
     j=i+1;
     //Find the key node and insert the new node
     while(list1(i)(1).number~=key)
         list1(i+1)(1).number=list1(i)(1).number;
         i=i-1;
         if(i==0) then
             printf("Item not Found");
             return;
         end
      end
      list1(i+1)(1).number=list1(i)(1).number
      list1(i)(1).number=x; //Inset the new node before the key node
      list1(j)(1).add=j;
      list1(j)(1).next=NULL;
      list1(j-1)(1).next=list1(j)(1).add;
      List=list1;
endfunction
//Function to print the numbers of list
function []=print1(list1)
     if list1(i)(1).next~=NULL then
        printf("%d--->",list1(i)(1).number);//Print current item
        i=i+1;
        if list1(i)(1).next==NULL then
             printf("%d",list1(i)(1).number);
        end
        print1(list1);//Move to next item
     end
     return;
endfunction
global List;
NULL=0;i=1;
//Create the structure i.e. node
node=struct('number',0,'add',0,'next',0);
head=node;
//Calling the functions
List=create(head);
printf("\nOriginal List: ");
print1(List);
List=insert(List);
printf("\nNew List: ");
print1(List);
```

Scilab code Exa 13.5 Delete a specified node in the list

```
//                    Example 13.5
//Write a program/function to delete a specified node.

funcprot(0);
//Create the list
function [List]=create(list1)
    global List;
    // Create the current node
    list1.number=input("Input a number(Type -999 to
       end); ")//scanf("%d");
    if list1.number==-999 then
        list1.next=NULL;
        list1.add=NULL;
     else
         list1.add=list1.add+1;
         list1.next=NULL;
         List(i)=list1;
         if(i==1) then

         else
             List(i-1).next=List(i).add
         end
         i=i+1;
         create(list1);// Create the next node
     end
     return;
endfunction
//Function to print the numbers of list
function []=print1(list1)
     if list1(i)(1).next~=NULL then
        printf("%d——>",list1(i)(1).number);//Print
           current item
        i=i+1;
        if list1(i)(1).next==NULL then
             printf("%d",list1(i)(1).number);
        end
        print1(list1);//Move to next item
     end
     return;
endfunction
//Function to delete the specified node
function [List]=delet(list1)
    key=input("Value of item number to be deleted?")
       ;//Read value of key
    //Find and delete the key node
    while(list1(i)(1).number~=key) then
        if list1(i)(1).next==NULL then
```

```
            printf("Item not found in the list");
            return;
        end
        i=i+1;
     end
     while(list1(i).next~=NULL)
         list1(i)(1).number=list1(i+1)(1).number;
         i=i+1;
     end
     list1(i-1)(1).next=NULL;
     List=list1;
endfunction
global List;
NULL=0;i=1;
//Create the structure i.e. node
node=struct('number',0,'add',0,'next',0);
head=node;
//Calling the functions
List=create(head);
printf("\nOriginal List: ");
print1(List);
List=delet(List);
printf("\nAfter deletion List is: ");
print1(List)
```