

Scilab Textbook Companion for
Programming In Ansi C
by E. Balagurusamy¹

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

July 31, 2019

¹Funded 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	4
1 Overview of C	5
2 Constants Variables and Data Types	10
3 Operators and Expressions	17
4 Managing Input and Output Operations	25
5 Decision Making and Branching	39
6 Decision Making and Looping	51
7 Arrays	67
8 Character Arrays and Strings	87
9 User Defined Functions	102
10 Structures and Unions	115
12 File Management in C	125
13 Dynamic Memory Allocation and linked Lists	137

List of Scilab Codes

Exa 1.3	Printing a message	5
Exa 1.4	Adding two numbers	5
Exa 1.5	Interest Calculation	6
Exa 1.6	Use of subroutines	7
Exa 1.7	Use of math functions	8
Exa 2.1	Representation of integer constants on a 16 bit computer	10
Exa 2.1cs	Case study 1 avg of numbers	11
Exa 2.2	typical declarations assignments and values stored in various types of variables	11
Exa 2.2cs	Case study temprature in Farenheit and Cel- sius	13
Exa 2.3	use of scanf function	13
Exa 2.4	Interest calculation using scanf	14
Exa 3.1	Use of integer arithmetic	17
Exa 3.1cs	case study 1 salesmans salary	17
Exa 3.2	Sequence of squares of numbers	18
Exa 3.2cs	case study 2 solution of the quadratic equation	19
Exa 3.3	Different kind of operators	20
Exa 3.4	Use of variables in expressions	22
Exa 3.5	Round off errors	22
Exa 3.6	Cast to evaluate the equation	23
Exa 4.1	Use of getchar function	25
Exa 4.1cs	Case study 1 Inventory report	26
Exa 4.2	Whether character is alphabet or digit or spe- cial character	27
Exa 4.2cs	Case study 2 Reliability graph	30
Exa 4.3	Print character in reverse case	30

Exa 4.4	Input formatting options	31
Exa 4.5	Reading of real numbers	33
Exa 4.6	Reading of strings	33
Exa 4.8	Testing for correctness	34
Exa 4.9	Output of integer numbers	35
Exa 4.10	Printing a real number	36
Exa 4.11	Printing of character and strings	37
Exa 5.1	Ratio	39
Exa 5.1cs	Case study 1 range of numbers	40
Exa 5.2	counts the number of boys	41
Exa 5.2cs	Case study 2 Pay Bill Calculations	42
Exa 5.3	Evaluate the power series	45
Exa 5.4	Largest of the three numbers	45
Exa 5.5	Reads the customer number and power consumed	47
Exa 5.6	Loan applications and to sanction loans	48
Exa 5.7	square root for five numbers	49
Exa 6.1	evaluate the equation	51
Exa 6.1cs	Case study 1 Table of Binomial Coefficients	51
Exa 6.2	Multiplication table	53
Exa 6.2cs	Case study 2 Histogram	54
Exa 6.3	Uses a for loop	55
Exa 6.3cs	Case study 3 Minimum Cost	57
Exa 6.4	Read the marks and print total marks	58
Exa 6.4cs	Case study 4 Plotting of two Functions	59
Exa 6.5	Use of the break statement	62
Exa 6.6	Evaluate the series	63
Exa 6.7	Use of continue statement	65
Exa 1.cs	Case study 1 Median of list of numbers	67
Exa 2.cs	Case study 2 Calculation of standard deviation	68
Exa 3.cs	Case study 3 Evaluating a Test	70
Exa 4.cs	Case study 4 Production and sales analysis	71
Exa 7.1	Sum of squares of 10 numbers	76
Exa 7.2	Count the number of students	76
Exa 7.3	Compute and print	80
Exa 7.4	Multiplication table	82
Exa 7.5	Popularity of various cars	83
Exa 1.cs	Case study 1 Counting words in a text	87

Exa 2.cs	Case study 2 Processing of a customer list . . .	88
Exa 8.1	Read a series of words	90
Exa 8.2	Read a line of text	92
Exa 8.3	Copy one string into another	92
Exa 8.4	Display the string under various format specifications	93
Exa 8.5	Program using for loop	94
Exa 8.6	Print the alphabet set a to z and A to Z . . .	96
Exa 8.7	Concatenate the three parts into one string . .	98
Exa 8.8	Compare whether strings are equal	99
Exa 8.9	Sort a list of names in alphabetical order . .	100
Exa 1.cs	Case study 1 Calculation of Area under a Curve	102
Exa 9.1	Multiple functions	103
Exa 9.2	Include arguments in function calls	105
Exa 9.3	Return result	106
Exa 9.4	Computes x raised to the power y	107
Exa 9.5	Calculate standard deviation of an array values	108
Exa 9.6	Sort an array	110
Exa 9.7	Automatic variables	111
Exa 9.8	Global variables	112
Exa 9.16	Factorial of a number using recursion	113
Exa 1.cs	Case study 1 Book Shop Inventory	115
Exa 10.1	Define a structure type	117
Exa 10.2	Comparison of structure variables	119
Exa 10.3	Calculate the subject wise and student wise totals	120
Exa 10.4	Array member to represent the three subjects	122
Exa 10.5	structure as a parameter to a function	123
Exa 12.1	Read data from keyboard and write it to a file	125
Exa 12.2	Read and write odd and even numbers	126
Exa 12.3	Read and write data to and from the file INVENTORY	128
Exa 12.4	Error handling in file operations	130
Exa 12.5	use of function ftell or mtell and fseek or mseek	131
Exa 12.6	Append additional items to the file INVENTORY	133

Exa 1.cs	Case study 1 Insertion in a sorted list	137
Exa 2.cs	Case study 2 Building a Sorted List	140
Exa 13.3	Create a linear linked list	142
Exa 13.4	Insert the item before the specified key node	144
Exa 13.5	Delete a specified node in the list	147

Chapter 1

Overview of C

Scilab code Exa 1.3 Printing a message

```
1 //           Example 1.3
2 // SAMPLE PROGRAM 1: PRINTING A MESSAGE
3
4 //Printing Begins
5 printf("I see ,I remember") //Printing using printf()
   function
6 //Printing ends
7
8
9 //We can also print a message using disp() function
   in scilab
10 disp("I see ,I remember") //Printing using disp()
   function
```

Scilab code Exa 1.4 Adding two numbers

```

-->exec('C:\Users\Devender Thakur\Desktop\scilab\scilab programs\1.introduction\exml.3.sce', -1)
I see,I remember
I see,I remember

-->|

```

Figure 1.1: Printing a message

```

-->exec('C:\Users\Devender Thakur\Desktop\scilab\scilab programs\1.introduction\exml.4.sce', -1)

100.
106.10
-->

```

Figure 1.2: Adding two numbers

```

1 //           Example 1.4
2 // SAMPLE PROGRAM 2: ADDING TWO NUMBERS
3
4 number=100;
5 amount=30.75+75.35 ;           //Addition of two numbers
6 disp(number) ;               //Display value of number
   variable
7 //Display value of amount in floating point with
   five places in all and two places to the right of
   decimalpoint
8 printf("%5.2 f", amount);

```

Scilab code Exa 1.5 Interest Calculation

```

1 //           Example 1.5
2 // SAMPLE PROGRAM 3: INTEREST CALCULATION
3
4 PRINCIPAL=5000.00;
5 PERIOD=10;
6 //Assignment Statements
7 amount=PRINCIPAL;

```

```

-->exec('C:\Users\Devender Thakur\Desktop\scilab\scilab programs\1.introduction\exml1.5.sce', -1)
0 5000.00
1 5550.00
2 6160.50
3 6838.15
4 7590.35
5 8425.29
6 9352.07
7 10380.80
8 11522.69
9 12790.18
10 14197.10

```

Figure 1.3: Interest Calculation

```

8 inrate=0.11;
9 year=0;
10 //Computation using while loop
11 while(year<=PERIOD)
12     printf("%2d %8.2f\n",year,amount);
13     value=amount+inrate*amount;
14     year=year+1;
15     amount=value;
16 end //End of while loop

```

Scilab code Exa 1.6 Use of subroutines

```

1 //           Example 1.6
2 // SAMPLE PROGRAM 4: USE OF SUBROUTINES
3
4 //Program using function
5
6 function []=mul(a,b) // mul() function starts (i.e.
7     definition starts)
8     y=a*b;
9     printf(" Multiplication of %d and %d is %d",a,b,y)
10    ;

```

```
-->exec('C:\Users\Devender Thakur\Desktop\scilab\scilab programs\1.introduction\exml.6.sci', -1)
Multiplication of 5 and 10 is 50
-->|
```

Figure 1.4: Use of subroutines

```
9 endfunction          // mul() function ends
10 a=5;b=10;
11 // Calling mul() function
12 mul(a,b)
```

Scilab code Exa 1.7 Use of math functions

```
1 //          Example 1.7
2 // SAMPLE PROGRAM 5: USE OF MATH FUNCTIONS
3
4 //Program using cosine function
5 angle=0;MAX=180;
6 printf("          Angle          Cos(angle)\n");
7 while(angle<=MAX)
8     x=(%pi/MAX)*angle;
9     y=cos(x);          //Use of cosine function
10    printf("%15d %13.4f\n",angle,y);
11    angle=angle+10;
12 end
```

```
-->exec('C:\Users\Devender Thakur\Desktop\scilab\scilab programs\1.introduction\exml.7.sce', -1)
  Angle      Cos (angle)
    0         1.0000
   10         0.9848
   20         0.9397
   30         0.8660
   40         0.7660
   50         0.6428
   60         0.5000
   70         0.3420
   80         0.1736
   90         0.0000
  100        -0.1736
  110        -0.3420
  120        -0.5000
  130        -0.6428
  140        -0.7660
  150        -0.8660
  160        -0.9397
  170        -0.9848
  180        -1.0000
```

Figure 1.5: Use of math functions

Chapter 2

Constants Variables and Data Types

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

```
1 //           Example 2.1
2 // Representation of integer constants on a 16-bit
  computer.
3
4 disp("Integer values");
5 //Integer values larger than 32767 are not stored
  properly on 16-bit machine
6 printf("%d %d %d \n",int16(32767),int16(32767+1),
  int16(32767+10));
7
8 disp("Long integer values");
9 //To store long integers properly, use int32 integer
  type
10 printf("%ld %ld %ld \n",int32(32767),int32(32767+1),
  int32(32767+10));
11 //The same result as from above statement can be
  achieved directly from below commented statement
12 //printf("%ld %ld %ld\n",32767,32767+1,32767+10);
```

```

-->exec('C:\Users\Devender Thakur\Desktop\scilab\scilab programs\2.Datatype\exm2.1.sce', -1)

Integer values
32767 -32768 -32759

Long integer values
32767 32768 32777

```

Figure 2.1: Representation of integer constants on a 16 bit computer

Scilab code Exa 2.1cs Case study 1 avg of numbers

```

1 // Case Study:-Chapter 2 Page No.-47
2 // 1.Calculation of Average of numbers
3
4 N=10;sum1=0;count=0; //Initialization of
   variables
5 printf(" Enter ten numbers");
6 while(count<N)
7     number=scanf("%f"); //Reading number(using
   scanf() function)
8     sum1=sum1+number;
9     count=count+1;
10 end
11 average=sum1/N; //Avarage is calculated
12 printf(" N = %d Sum1 = %f",N,sum1 );
13 printf(" Average = %f",average );

```

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

```

-->exec('C:\Users\Devender Thakur\Desktop\scilab\scilab programs\2.Datatype\Casestudy2.1.sce', -1)
Enter ten numbers
-->1
-->2.3
-->4.67
-->1.42
-->7
-->3.67
-->4.08
-->2.2
-->4.25
-->8.21
N = 10 Sum1 = 38.800000 Average = 3.880000
-->

```

Figure 2.2: Case study 1 avg of numbers

```

1 //      Example 2.2
2 // Program shows typical declarations ,assignments
   and values stored in various types of variables.
3
4 //Declarations and Assignments
5 m=int16(54321);
6 n=int32(1234567890);
7 k=uint16(54321);
8 //Assignments
9 x=1.234567890000;           //Bydefault type is double
   in scilab
10 y=9.87654321;           //


---


11 p=1.0; q=1.0;
12 //Printing
13 printf(" m=%d\n", m)
14 printf(" n=%ld\n", n)
15 printf(" x=%0.12 f\n", x)
16 printf(" x=%f\n", x)
17 printf(" y=%0.12 f\n", y)
18 printf(" y=%f\n", y)
19 printf(" k=%u p=%f q=%0.12 f\n", k, p, q)


---



```



```

-->exec('C:\Users\Devender Thakur\Desktop\scilab\scilab programs\2.Datatype\exm2.2.sce', -1)
m=-11215
n=1234567890
x=1.234567890000
x=1.234568
y=9.876543210000
y=9.876543
k=54321 p=1.000000 q=1.000000000000

```

Figure 2.3: typical declarations assignments and values stored in various types of variables

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

```

1 //          Case Study:-Chapter 2   Page no.-48
2 //  2.Solution of temprature in Farenheit and
   Celsius
3 F_LOW=0;
4 F_MAX=250;
5 STEP=25;
6 fahrenheit=F_LOW;      //Initialization
7 printf("Fahrenheit      Celsius\n\n");
8 while(fahrenheit < = F_MAX)
9     celsius=(fahrenheit-32.0)/1.8;      //conversion
   from Farenheit to Celsius
10    printf("%6.2 f          %7.2 f\n", fahrenheit,
   celsius);
11    fahrenheit=fahrenheit+STEP;
12    end

```

Scilab code Exa 2.3 use of scanf function

```
-->exec('C:\Users\Devender Thakur\Desktop\scilab\scilab programs\2.Datatype\Casestudy2.2.sce', -1)
Fahrenheit      Celsius
0.00            -17.78
25.00           -3.89
50.00           10.00
75.00           23.89
100.00          37.78
125.00          51.67
150.00          65.56
175.00          79.44
200.00          93.33
225.00         107.22
250.00         121.11
```

Figure 2.4: Case study temperature in Fahrenheit and Celsius

```
1 //          Example 2.3
2 // The program illustrate the use of scanf()
  function
3 disp("Enter an interger number:");
4 number=scanf("%d");          //Read from keyboard
5 if(number<100) then
6     disp("Your number is smaller than 100");
7 else
8     disp("Your number contain more than two digits")
9     ;
9 end
```

Scilab code Exa 2.4 Interest calculation using scanf

```
1 //          Example 2.4
2 // Sample program 3(exm1.5) discussed in chapter 1
  can be convered in to a more flexible intractive
  program using scanf() function
3 disp("Enter in single line separted by space");
4 disp("Input amount, interest rate , and period");
```

```

-->exec('C:\Users\Devender Thakur\Desktop\scilab\scilab programs\2.Datatype\exm2.3.sce', -1)

Enter an interger number:
-->54

Your number is smaller than 100

-->exec('C:\Users\Devender Thakur\Desktop\scilab\scilab programs\2.Datatype\exm2.3.sce', -1)

Enter an interger number:
-->108

Your number contain more than two digits

```

Figure 2.5: use of scanf function

```

5 [amount , inrate , period]=scanf("%f %f %d");           // use
   of scanf()
6 year=1;
7 //Computation using while loop
8 while(year<=period)
9     value=amount+inrate*amount;
10     printf("%2d Rs %8.2f\n",year,value)
11     year=year+1;
12     amount=value;
13 end

```

```

-->exec('C:\Users\Devender Thakur\Desktop\scilab\scilab programs\2.Datatype\exm2.4.sce', -1)

Enter in single line separted by space

Input amount,interest rate, and period
-->10000 0.14 5
1 Rs 11400.00
2 Rs 12996.00
3 Rs 14815.44
4 Rs 16889.60
5 Rs 19254.15

-->exec('C:\Users\Devender Thakur\Desktop\scilab\scilab programs\2.Datatype\exm2.4.sce', -1)

Enter in single line separted by space

Input amount,interest rate, and period
-->20000 0.12 7
1 Rs 22400.00
2 Rs 25088.00
3 Rs 28098.56
4 Rs 31470.39
5 Rs 35246.83
6 Rs 39476.45
7 Rs 44213.63

```

Figure 2.6: Interest calculation using scanf

Chapter 3

Operators and Expressions

Scilab code Exa 3.1 Use of integer arithmetic

```
1 //                               Example 3.1
2 //The program shows the use of integer arithmetic to
   convert a given number-
3 //of days into months and days
4
5 days=input('Enter days:');
6 months=int16(days/30);           //Compute for months
7 days =int16(pmodulo(days,30));  //compute for days
8 disp(days,"Days =",months,"Months =");
```

Scilab code Exa 3.1cs case study 1 salesmans salary

```
1 //           Case Study:-Chapter 3,Page No:76
2 //           1.Salesman 's Salary
3
4
5 BASE_SALARY=1500.00;           //Minimum base salary
```

```

-->exec('C:\Users\Devender Thakur\Desktop\scilab\scilab programs\3.operator\exm3.1.sce', -1)
Enter days:265

Months =

      8

Days =

     25

```

Figure 3.1: Use of integer arithmetic

```

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

```

Scilab code Exa 3.2 Sequence of squares of numbers

```

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

```

```

-->exec('C:\Users\Devender Thakur\Desktop\scilab\scilab programs\3.operator\Casestudy3_1.sce', -1)
Input number sold and price
[Enter in single line separated by space]
-->5 20450.00
Bonus          = 1000.00
Commission     = 2045.00
Gross salary   = 4545.00

```

Figure 3.2: case study 1 salesmans salary

```

-->exec('C:\Users\Devender Thakur\Desktop\scilab\scilab programs\3.operator\exm3.2.sce', -1)

2.

4.

16.

```

Figure 3.3: Sequence of squares of numbers

```

3
4 N=100; A=2;
5 a=A;
6 while(a<N)
7     disp(a); //Prints square of number
8     a=a^2; //compute square of number
9 end

```

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

```

1 //          Case Study:-Chapter 3,Page No:77
2 //          2.Solution of the Quadratic equation
3
4
5 printf("Input values of a, b, and c \n");
6 a=input("a =");
7 b=input("b =");
8 c=input("c =");

```

```

-->exec('C:\Users\Devender Thakur\Desktop\scilab\scilab programs\3.operator\Casestudy3_2.sce', -1)
Input values of a, b, and c
a =2
b =4
c =-16

Root1 = 2.00

Root2 = -4.00

-->exec('C:\Users\Devender Thakur\Desktop\scilab\scilab programs\3.operator\Casestudy3_2.sce', -1)
Input values of a, b, and c
a =1
b =2
c =3

ROOTS ARE IMAGINARY

```

Figure 3.4: case study 2 solution of the quadratic equation

```

9 discriminant = b^2-4*a*c;
10 if(discriminant<0)
11     printf("\n\nROOTS ARE IMAGINARY\n");
12 else
13     //Computes root1 and root2
14     root1 = (-b + sqrt(discriminant))/(2.0*a);
15     root2 = (-b - sqrt(discriminant))/(2.0*a);
16     printf("\n\nRoot1 = %5.2f\n\nRoot2 = %5.2f\n",
17           root1,root2 );
17 end

```

Scilab code Exa 3.3 Different kind of operators

```

1 //                               Example 3.3
2 //The program employs diffrent kind of operators.
  The results of their evaluation are also shown
  for comparison
3

```



```

-->exec('C:\Users\Devender Thakur\Desktop\scilab\scilab programs\3.operator\exm3.3.sce', -1)
a = 16 b = 10 c = 6
a = 16 b = 11 d = 27
a/b = 1
pmodulo(a,b) = 5
a*b = 176

0.

```

Figure 3.5: Different kind of operators

```

4 //Increment(i.e. ++)/Decrement(--) operators are
   unavailable in Scilab
5 a=int16(15);
6 b=int16(10);
7 a=a+1; // Replacement for ++a
8 c=a-b;
9 printf("a = %d b = %d c = %d\n",a,b,c);
10 b=b+1; // Replacement for b++
11 d=b+a;
12 printf("a = %d b = %d d = %d\n",a,b,d);
13 printf("a/b = %d\n",a/b); //
   Division operator
14 printf("pmodulo(a,b) = %d\n",pmodulo(a,b)); //
   Modulus operator
15 printf("a*b = %d\n",a*b); //
   Multiplication
16 //In scilab ther is no conditional operator(?:),
   hence 'if' can be used in place of ?:
17 if(c>d) then
18     disp(1);
19 end
20 if(c<d) then
21     disp(0);
22 end

```

```
-->exec('C:\Users\Devender Thakur\Desktop\scilab\scilab programs\3.operator\exm3.4.sce', -1)
x=10.000000
y=7.000000
z=4.000000
```

Figure 3.6: Use of variables in expressions

Scilab code Exa 3.4 Use of variables in expressions

```
1 //                               Example 3.4
2 //The program illustrates the use of variables in
   expressions and their evaluation.
3
4 a=9;b=12;c=3;
5 //Expressions and their evaluations
6 x=a-b/3+c*2-1;
7 y=a-b/(3+c*(2-1));
8 z=a-(b/(3+c)*2)-1;
9
10 printf("x=%f\n",x)
11 printf("y=%f\n",y)
12 printf("z=%f\n",z)
13 // disp can be used..
14 //disp(x,"x=")
15 //disp(y,"y=")
16 //disp(z,"z=")
```

Scilab code Exa 3.5 Round off errors

```
1 //                               Example 3.5
2 //Output of program shows round-off errors that can
   occur in computation of floating point numbers
3
4 //Sum of n terms of 1/n
```

```

-->exec('C:\Users\Devender Thakur\Desktop\scilab\scilab programs\3.operator\exm3.5.sce', -1)
Enter value of n:99
Sum= 1.000000
-->

```

Figure 3.7: Round off errors

```

5 count=1;
6 sum1=0;
7 n=input("Enter value of n:");
8 term=1.0/n;
9 while(count<=n)
10     sum1=sum1+term;
11     count=count+1;
12 end
13 printf("Sum= %f",sum1);

```

Scilab code Exa 3.6 Cast to evaluate the equation

```

1 //                               Example 3.6
2 //Program using a cast to evaluate the equation.
3
4 sum1=0;
5 for n=int8(1:10)
6     sum1=sum1+1/double(n); //conversion from 'int'
7     printf("%2d %6.4 f\n",n,sum1);
8     to 'double' or 'float'
9
10    end

```

```
-->exec('C:\Users\Devender Thakur\Desktop\scilab\scilab programs\3.operator\exm3.6.sce', -1)
1 1.0000
2 1.5000
3 1.8333
4 2.0833
5 2.2833
6 2.4500
7 2.5929
8 2.7179
9 2.8290
10 2.9290
```

Figure 3.8: Cast to evaluate the equation

Chapter 4

Managing Input and Output Operations

Scilab code Exa 4.1 Use of getchar function

```
1 //                               Exaymple 4.1
2 // The program shows the use of getchar function in
  an interactive environment.
3 //In Scilab in place of getchar function scanf
  function can be used to get
4 //character as there is no getchar function in
  Scilab.
5
6 disp("Would you like to know my name?");
7 disp("Type Y for YES and N for NO:");
8 answer=scanf("%c"); //Reading
  character
9 if (answer=='Y')|(answer=='y') then //Test for
  answer
10     disp("My name is BUSY BEE");
11 else
12     disp("You are good for nothing")
13 end
```

```

-->exec('C:\Users\Devender Thakur\Desktop\scilab\scilab programs\4.IO\exm4.1.sce', -1)

Would you like to know my name?

Type Y for YES and N for NO:
-->y

My name is BUSY BEE

-->exec('C:\Users\Devender Thakur\Desktop\scilab\scilab programs\4.IO\exm4.1.sce', -1)

Would you like to know my name?

Type Y for YES and N for NO:
-->n

You are good for nothing

```

Figure 4.1: Use of getchar function

Scilab code Exa 4.1cs Case study 1 Inventory report

```

1 //          Case Study:-Chapter 4 ,Page No:106
2 //          1.Inventory Report
3
4 ITEMS=4;
5 i=1;
6 printf("[Enter in single line seperated be spaces]\n
   ");
7 while(i <= 4)
8     printf("Enter code, quantity , and rate:");
9     [code(i),quantity(i),rate(i)]=scanf("%s %d %f")
   ;
10    i=i+1;
11 end
12 printf("INVENTORY REPORT\n");
13 printf("

```

```

    -----\n"
);
14 printf(" Code          Quantity      Rate      Value\n")
;
15 printf("
    -----\n"
);
16 total_value=0;
17 i=1;
18 while(i<=ITEMS)
19     value=quantity(i)*rate(i);
20     printf("%6s %10d %10.2f      %e\n",code(i),
            quantity(i),rate(i),value);
21     total_value= total_value+value;
22     i=i+1;
23 end
24     printf("
    -----\
n");
25     printf("                Total Value = %e\n",
            total_value);
26     printf("
    -----\
n");
-----

```

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

```

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

```

```

-->exec('C:\Users\Devender Thakur\Desktop\scilab\scilab programs\4.IO\Casestudy4_1.sce', -1)
[Enter in single line seperated be spaces]
Enter code,quantity, and rate:
-->F105 275 575.00
Enter code,quantity, and rate:
-->H220 107 99.95
Enter code,quantity, and rate:
-->I019 321 215.50
Enter code,quantity, and rate:
-->M315 89 725.00
INVENTORY REPORT
-----
Code      Quantity  Rate      Value
-----
F105      275       575.00    1.581250e+005
H220      107       99.95     1.069465e+004
I019      321       215.50    6.917550e+004
M315      89        725.00    6.452500e+004
-----
Total Value = 3.025202e+005
-----

```

Figure 4.2: Case study 1 Inventory report

```

5
6 disp(" Press any key");
7 character=scanf("%c"); //
  Reading character
8 if (isletter(character)) then //Test
  for letter
9     disp("The character is a letter");
10    elseif (isdigit(character)) then //Test
    for digit
11        disp("The character is a digit");
12    else
13        disp("The character is not alphanumeric"
14        );
14 end

```

Scilab code Exa 4.2cs Case study 2 Reliability graph

```
1 //          Case study:chapter 4
2 //          2. Reliability Graph
3
4 LAMBDA=0.001;
5 for i=1:27
6     printf("—");
7 end
8 printf("\n");
9 for t=0:150:3000
10    r=exp(-LAMBDA*t);
11    R=int32(50*r+0.5);
12    printf("|");
13    for i=1:R
14        printf("*");
15    end
16    printf("#\n");
17 end
18 for i=1:2
19    printf("|\n");
20 end
```

Scilab code Exa 4.3 Print character in reverse case

```
1 //          Example 4.3
2 // A program that reads a character from the
3 // keyboard and then print in reverse
4 // case ,that is ,if input is in upper case ,the output
5 // will be lower case and vice-versa
6 disp("Enter an alphabet");
```

```

-->exec('C:\Users\Devender Thakur\Desktop\scilab\scilab programs\4.IO\exm4_3.sce', -1)

Enter an alphabet
-->a

A

-->exec('C:\Users\Devender Thakur\Desktop\scilab\scilab programs\4.IO\exm4_3.sce', -1)

Enter an alphabet
-->Q

q

```

Figure 4.5: Print character in reverse case

```

5 alphabet=scanf("%c"); //Reading
   character
6 if((ascii(alphabet))>=97) then
7     disp(convstr(alphabet,"u")); //Reverse and
   display
8 else
9     disp(convstr(alphabet,"l")); //Reverse and
   display
10 end

```

Scilab code Exa 4.4 Input formatting options

```

1 // Example 4.4
2 // The program illustrates the various options for
   reading
3 //integers are experimented in this program
4
5
6 printf("Enter three integer numbers\n");
7 [n,a,b,c]=mscanf("%d %*d %d");
8 disp(c,b,a);

```

```

-->exec('C:\Users\Devender Thakur\Desktop\scilab\scilab programs\4.IO\exm4_4.sce', -1)
Enter three integer numbers
-->1 2 3

    1.

    3.

    []
Enter two 4-digit numbers
-->6789 4321
67 89
Enter two integers
-->44 66
44 66
Enter a nine digit number
-->123456789
123 4567 89
Enter two three digit numbers
-->123 456
123 456

```

Figure 4.6: Input formatting options

```

9  printf("Enter two 4-digit numbers\n");
10 [n,x,y]=mscanf("%2d %4d");
11 printf( '%d %d\n', x,y);
12
13 printf("Enter two integers\n");
14 [n,a,x]=mscanf("%d %d");
15 printf( '%d %d\n', a,x);
16
17 printf("Enter a nine digit number\n");
18 [n,p,q,r]=mscanf("%3d %4d %3d");
19 printf( '%d %d %d\n', p,q,r);
20
21 printf("Enter two three digit numbers\n");
22 [n,x,y]=mscanf("%d %d");
23 printf( '%d %d \n', x,y);

```

```

-->exec('C:\Users\Devender Thakur\Desktop\scilab\scilab programs\4.IO\exm4_5.sce', -1)
Values of x and y:
-->12.345600 17.5e-2

x = 12.345600
y = 0.175000
Values of p and q:
-->4.142857142857 18.5678901234567890

p = 4.142857142857
q = 1.8567890123456e+001

```

Figure 4.7: Reading of real numbers

Scilab code Exa 4.5 Reading of real numbers

```

1 //           Example 4.5
2 //Reading of real numbers(in both decimal point and
   exponential notation
3
4 printf(" Values of x and y:");
5 [x,y]=scanf("%f %e"); //reading x [decimal point]
   and y[exponential] from standard input
6 printf("\n");
7 printf("x = %f\ny = %f\n",x,y);
8 printf(" Values of p and q:");
9 [p,q]=scanf("%lg %lg");
10 printf("\np = %.12f\nq = %.12e\n",p,q);

```

Scilab code Exa 4.6 Reading of strings

```

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

```

```

3
4 //scanf() can only read one character at a time and
   return it to name1, therefore
5 //mscanf() is used that can read more than one
   character
6 printf("Enter serial number and name one:");
7 [n,no,name1]=mscanf("%d %15c");
8 printf("%d %15s\n",no,name1);
9
10 //Read and display a string
11 printf("Enter serial number and name two:");
12 [no,name2]=scanf("%d %s");
13 printf("%d %15s\n",no,name2);
14
15 //Read and display a string of width 15
16 printf("Enter serial number and name three:");
17 [no,name3]=scanf("%d %15s");
18 printf("%d %15s",no,name3);

```

Scilab code Exa 4.8 Testing for correctness

```

1 //           Example 4.8
2 // The program illustrates the testing for
   correctness of reading data by scanf function
3
4 //In Scilab mscanf function is similar to scanf
   function of c
5 printf("Enter values of a,b and c");
6 [n,a,b,c]=mscanf("%d %f %c");
7 if(n==3) then           //Test condition
8     printf("a = %d b = %f c = %c",a,b,c);
9 else
10     printf("Error in input.");

```

```

-->exec('C:\Users\Devender Thakur\Desktop\scilab\scilab programs\4.IO\exm4_6.sce', -1)
Enter serial number and name one:
-->1 123456789012345
1 123456789012345
Enter serial number and name two:
-->2 New York
2          New
Enter serial number and name three:
-->3 York
3          York
-->exec('C:\Users\Devender Thakur\Desktop\scilab\scilab programs\4.IO\exm4_6.sce', -1)
Enter serial number and name one:
-->1 123456789012
1 123456789012
Enter serial number and name two:
-->2 New_York
2          New_York
Enter serial number and name three:
-->3 London
3          London

```

Figure 4.8: Reading of strings

11 **end**

Scilab code Exa 4.9 Output of integer numbers

```

1 //          Example 4.9
2 // The program illustrates the output of integer
  numbers under various formats
3
4 m=int16(12345);          //integer variable m
5 n=int32(987654);        //integer variable n
6 printf( '%d\n', m);
7 printf( '%10d\n', m);
8 printf( '%010d\n', m);
9 printf( '%-10d\n', m);
10 printf( '%10d\n', n);
11 printf( '%10d\n', -n);

```

```

-->exec('C:\Users\Devender Thakur\Desktop\scilab\scilab programs\4.IO\exm4_8.sce', -1)
Enter values of a,b and c
-->12 3.45 A
a = 12 b = 3.450000 c = A
-->exec('C:\Users\Devender Thakur\Desktop\scilab\scilab programs\4.IO\exm4_8.sce', -1)
Enter values of a,b and c
-->23 78 9
a = 23 b = 78.000000 c = 9
-->exec('C:\Users\Devender Thakur\Desktop\scilab\scilab programs\4.IO\exm4_8.sce', -1)
Enter values of a,b and c
-->8 A 5.25
Error in input.
-->exec('C:\Users\Devender Thakur\Desktop\scilab\scilab programs\4.IO\exm4_8.sce', -1)
Enter values of a,b and c
-->15.75 23 X
a = 15 b = 0.750000 c = 2
-->

```

Figure 4.9: Testing for correctness

```

-->exec('C:\Users\Devender Thakur\Desktop\scilab\scilab programs\4.IO\exm4_9.sce', -1)
12345
    12345
0000012345
12345
    987654
   -987654

```

Figure 4.10: Output of integer numbers

Scilab code Exa 4.10 Printing a real number

```

1 //           Example 4.10
2 // The program illustrates all the options of
  printing a real number.
3
4 y=98.7654;           //real number
5 //Various options of printing a real number
6 printf( '%7.4 f\n', y);

```



```

-->exec('C:\Users\Devender Thakur\Desktop\scilab\scilab programs\4.IO\exm4_10.sce', -1)
98.7654
98.765400
  98.77
98.77
0098.77
  98.77

  9.88e+001
-9.8765e+001
  9.88e+001
  9.876540e+001

```

Figure 4.11: Printing a real number

```

7 printf( '%f\n', y);
8 printf( '%7.2 f\n', y);
9 printf( '%-7.2 f\n', y);
10 printf( '%07.2 f\n', y);
11 printf( '%7.2 f\n', y);
12 printf( '\n');
13 printf( '%10.2 e\n', y);
14 printf( '%12.4 e\n', -y);
15 printf( '%-10.2 e\n', y);
16 printf( '%e\n', y);

```

Scilab code Exa 4.11 Printing of character and strings

```

1 //           Example 4.11
2 // The program illustrates the printing of character
  and strings.
3
4 x= 'A';           //x variable has character
5 name= 'ANIL KUMAR GUPTA'; //name variable has string
6 disp("OUTPUT OF CHARACTERS");
7 printf( '%c\n%3c\n%5c\n', x, x, x);

```

```

-->exec('C:\Users\Devender Thakur\Desktop\scilab\scilab programs\4.IO\exm4_11.sce', -1)

OUTPUT OF CHARACTERS
A
  A
    A
      A
        A

OUTPUT OF STRINGS
ANIL KUMAR GUPTA
  ANIL KUMAR GUPTA
    ANIL KUMAR
      ANIL
        ANIL KUMAR
          ANIL KUMAR GUPTA

```

Figure 4.12: Printing of character and strings

```

8 printf( '%3c\n%c\n', x, x );
9 printf( '\n' );
10 disp("OUTPUT OF STRINGS"); //disp can also be used
    for display
11 printf( '%s\n', name );
12 printf( '%20s\n', name );
13 printf( '%20.10s\n', name );
14 printf( '%.5s\n', name );
15 printf( '%-20.10s\n', name );
16 printf( '%5s\n', name );

```

Chapter 5

Decision Making and Branching

Scilab code Exa 5.1 Ratio

```
1 //                               Example 5.1
2 // The program reads four values a,b,c and d from
   the terminal and
3 //evaluates the ratio of (a+b) to (c+d) and prints
   the result ,if c-d
4 //is not equal to zero
5
6 disp("Enter four integer values");
7 a=int8(input(" a="));
8 b=int8(input(" b="));
9 c=int8(input(" c="));
10 d=int8(input(" d="));
11
12 if(c-d ~= 0) then //Execute statement block
13     ratio=double(a+b)/double(c-d);
14     printf(" Ratio=%f\n",ratio);
15 end
```

```
Enter four integer values
a=12
b=23
c=34
d=45
Ratio=-3.181818
```

Figure 5.1: Ratio

Scilab code Exa 5.1cs Case study 1 range of numbers

```
1 // Case Study:-Chapter 5, Page No:139
2 // 1.Range of numbers
3
4 sum1=0;
5 count=0;
6 f=0;value=1;
7 printf("Enter numbers[press enter after each number
   ]:\n input a NEGATIVE number to end\n");
8 while(value)
9     [value]=scanf("%f"); //Read data
10    if(value<0) then
11        break;
12    end
13    count=count+1;
14    //Calculating height and lowest value
15    if(count==1) then
16        high=value;
17        low=value;
18    elseif(value>high) then
19        high=value;
```

```

-->exec('C:\Users\Devender Thakur\Desktop\scilab\scilab programs\5.branching\Casestudy5_1.sce', -1)
Enter numbers[press enter after each number]:
input a NEGATIVE number to end
-->35
-->40.50
-->25
-->31.25
-->68.15
-->47
-->26.65
-->29
-->53.45
-->62.50
-->-1
Total values: 10
Highest-values: 68.150000
Lowest-value: 25.000000
Range: 43.150000
Avarage: 41.850000

```

Figure 5.2: Case study 1 range of numbers

```

20     elseif(value<low) then
21         low=value;
22     end
23     //Calculate sum
24     sum1=sum1+value;
25 end
26 average=sum1/count;           //Average cost
27 range1=high-low;             //Range of values
28 //Print the results
29 printf("Total values: %d\n",count);
30 printf("Highest-values: %f\nLowest-value: %f\n",high
    ,low);
31 printf("Range: %f\n Avarage: %f\n",range1,average);

```

Scilab code Exa 5.2 counts the number of boys

```

1 //                               Example 5.2
2 // The program counts the number of boys whose

```

```

    weight is less than 50Kg
3 //and height is greater than 170cm.
4
5 count=int(0);
6 disp("Enter weight and height for 10 boys");
7 for i=1:10
8     [weight,height]=scanf("%f %f");
9     if(weight<50&height>170) then // Test for
        weight and height
10         count=count+1;
11     end
12 end
13 disp("Number of boys with weight <50 kg and height
    >170 cm =");
14 disp(count)

```

Scilab code Exa 5.2cs Case study 2 Pay Bill Calculations

```

1 //          Case Study:-Chapter 5
2 //          2.Pay-Bill Calculations
3 CA1=1000;
4 CA2=750;
5 CA3=500;
6 CA4=250;
7 EA1=500;
8 EA2=200;
9 EA3=100;
10 EA4=0;
11 level=1;
12 while(level)
13     printf("Enter 0[zero] for level to end");

```

```

Enter weight and height for 10 boys
-->45 176.5
-->55 174.2
-->47 168.0
-->49 170.7
-->54 169.0
-->53 170.5
-->49 167.0
-->48 175.0
-->47 167
-->51 170

Number of boys with weight <50 kg and height>170 cm =

3.

```

Figure 5.3: counts the number of boys

```

-->exec('C:\Users\Devender Thakur\Desktop\scilab\scilab programs\5.branching\casestudy5_2.sce', -1)
Enter 0[zero] for level to end
Enter level:1
Enter job number, and basic pay
-->1111 4000
1 1111 5980.00
Enter 0[zero] for level to end
Enter level:2
Enter job number, and basic pay
-->2222 3000
2 2222 4465.00
Enter 0[zero] for level to end
Enter level:3
Enter job number, and basic pay
-->3333 2000
3 3333 3007.00
Enter 0[zero] for level to end
Enter level:4
Enter job number, and basic pay
-->4444 1000
4 4444 1500.00
Enter 0[zero] for level to end
Enter level:0
END OF THE PROGRAM
-->

```

Figure 5.4: Case study 2 Pay Bill Calculations

```

14 //Read data
15 level=input("Enter level:");
16 if(level==0)
17     break;
18 end
19 printf("Enter job number, and basic pay\n");
20 //Read data
21 [jobnumber,basic]=scanf("%d %f");
22 //Decide level number and calculate perks
23 select level
24 case 1 then perks=CA1+EA1;
25 case 2 then perks=CA2+EA2;
26 case 3 then perks=CA3+EA3;
27 case 4 then perks=CA4+EA4;
28 else
29     printf("Error in level code");
30     return;
31 end
32
33 house_rent=0.25*basic;
34 //Calculate gross salary
35 gross=basic+house_rent+perks;
36 //Calculate income tax
37 if (gross<=2000) then
38     incometax=0;
39 elseif(gross<=4000)
40     incometax=0.03*gross;
41 elseif(gross<=5000)
42     incometax=0.05*gross;
43 else
44     incometax=0.08*gross;
45 end
46 //Compute the net salary
47 net=gross-incometax;
48 //Print the results
49 printf("%d %d %.2f\n",level ,jobnumber ,net);
50 end
51 printf("END OF THE PROGRAM");

```

Scilab code Exa 5.3 Evaluate the power series

```
1 //                               Example 5.3
2 // A program to evaluate the power series. It uses
   if...else to test accuracy.
3 //  $e^x=1+x+x^2/2! + x^3/3! + \dots +x^n/n!, 0 < x < 1$ 
4
5 ACCURACY=0.0001;
6 x=input("Enter value of x:");
7 n=1;term=1;sum1=1;count=int8(1);
8 while(n<=100)
9     term=term* x/n;
10    sum1=sum1+term;
11    count=count+1;
12    if(term<ACCURACY) then           //Test for accuracy
13        n=999;
14    else
15        n=n+1;
16    end
17 end
18 //Print results
19 printf("Term=%d Sum=%f" ,count ,sum1);
```

Scilab code Exa 5.4 Largest of the three numbers

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

```

-->exec('C:\Users\Devender Thakur\Desktop\scilab\scilab programs\5.branching\exm5_3.sce', -1)
Enter value of x:0
Term=2 Sum=1.000000
-->exec('C:\Users\Devender Thakur\Desktop\scilab\scilab programs\5.branching\exm5_3.sce', -1)
Enter value of x:0.5
Term=7 Sum=1.648720
-->exec('C:\Users\Devender Thakur\Desktop\scilab\scilab programs\5.branching\exm5_3.sce', -1)
Enter value of x:0.75
Term=8 Sum=2.116997
-->exec('C:\Users\Devender Thakur\Desktop\scilab\scilab programs\5.branching\exm5_3.sce', -1)
Enter value of x:0.99
Term=9 Sum=2.691232
-->exec('C:\Users\Devender Thakur\Desktop\scilab\scilab programs\5.branching\exm5_3.sce', -1)
Enter value of x:1
Term=9 Sum=2.718279
-->

```

Figure 5.5: Evaluate the power series

```

4
5 disp("Enter three values");
6 A=input("A=");
7 B=input("B=");
8 C=input("C=");
9 disp("Largest value is:");
10 if(A>B) , //Test for largest between A
    &B
11     if(A>C) , //Test for largest between A
        &C
12         disp(A);
13     else
14         disp(C);
15     end
16 else
17     if(C>B) , //Test for largest between C&
        B
18         disp(C);
19     else
20         disp(B);
21     end
22 end

```

```

-->exec('C:\Users\Devender Thakur\Desktop\scilab\scilab programs\5.branching\exm5_4.sce', -1)

Enter three values
A=23445
B=67379
C=88843

Largest value is:

88843.

```

Figure 5.6: Largest of the three numbers

Scilab code Exa 5.5 Reads the customer number and power consumed

```

1 // Example 5.5
2 // The program reads the customer number and power
  consumed and prints
3 //the amount to be paid by the customer
4
5 disp("Enter CUSTOMER NO. and UNITS consumed");
6 //Read data
7 custnum=input("CUSTOMER NO.:");
8 units=input("UNITS:");
9 //Use of else...if ladder
10 //Calculates charges
11 if(units<=200) ,
12     charges=0.5*units;
13 elseif(units<=400) ,
14     charges=100+0.65*(units-200);
15 elseif(units<=600) ,
16     charges=230+0.8*(units-400);
17 else
18     charges=390+(units-600);
19 end

```

```

-->exec('C:\Users\Devender Thakur\Desktop\scilab\scilab programs\5.branching\exm5_5.sce', -1)

Enter CUSTOMER NO. and UNITS consumed
CUSTOMER NO.:101
UNITS:150
Customer No:101 Charges=75.00
-->exec('C:\Users\Devender Thakur\Desktop\scilab\scilab programs\5.branching\exm5_5.sce', -1)

Enter CUSTOMER NO. and UNITS consumed
CUSTOMER NO.:202
UNITS:225
Customer No:202 Charges=116.25
-->exec('C:\Users\Devender Thakur\Desktop\scilab\scilab programs\5.branching\exm5_5.sce', -1)

Enter CUSTOMER NO. and UNITS consumed
CUSTOMER NO.:303
UNITS:375
Customer No:303 Charges=213.75
-->

```

Figure 5.7: Reads the customer number and power consumed

```

20 //Print results
21 printf("Customer No:%d Charges=%.2f", custnum,
charges);

```

Scilab code Exa 5.6 Loan applications and to sanction loans

```

1 // Example 5.6
2 // A program to process loan applications and to
sanction loans.
3
4 MAXLOAN=50000;
5 disp("Enter the values of previous two loans");
6 loan1=int32(input("Enter first loan:"));
7 loan2=int32(input("Enter second loan:"));
8 loan3=int32(input("Enter the values of new loan:"));
9 sum23=loan2+loan3;
10 //Calculate the sanction loan

```

```

-->exec('C:\Users\Devender Thakur\Desktop\scilab\scilab programs\5.branching\exm5_6.sce', -1)

Enter the values of previous two loans
Enter first loan:0
Enter second loan:20000
Enter the values of new loan:45000
Previous loans pending:0 20000
Loan requested =45000
Loan sanctioned =30000

-->exec('C:\Users\Devender Thakur\Desktop\scilab\scilab programs\5.branching\exm5_6.sce', -1)

Enter the values of previous two loans
Enter first loan:1000
Enter second loan:15000
Enter the values of new loan:25000
Previous loans pending:1000 15000
Loan requested =25000
Loan sanctioned =0

```

Figure 5.8: Loan applications and to sanction loans

```

11 if(loan1>0),
12     sancloan=0;
13 elseif(sum23>MAXLOAN),
14     sancloan=MAXLOAN-loan2;
15 else
16     sancloan=loan3;
17 end
18 //Print the results
19 printf("Previous loans pending:%d %d\n",loan1,loan2)
20     ;
21 printf("Loan requested =%d\n",loan3);
22 printf("Loan sanctioned =%d\n",sancloan);

```

Scilab code Exa 5.7 square root for five numbers

```

1 // Example 5.7
2 // The program evaluates the square root for five

```

```

-->exec('C:\Users\Devender Thakur\Desktop\scilab\scilab programs\5.branching\exm5_7.sce', -1)
Enter FIVE real values
-->50.70
50.700000      7.120393
-->40
40.000000      6.324555
-->-36
Value - 3 is negative
-->75
75.000000      8.660254
-->11.25
11.250000      3.354102
End of computation
-->

```

Figure 5.9: square root for five numbers

```

    numbers .
3
4 count=1;
5 printf("Enter FIVE real values\n");
6 while(count <=5)
7     x=scanf("%f");
8     if(x<0) then
9         printf("Value - %d is negative\n",count);
10    else
11        y=sqrt(x);           //Calculate square
                                root
12        printf("%f\t%f\n",x,y); //Print result
13    end
14    count=count+1;
15 end
16 printf("End of computation");

```

Chapter 6

Decision Making and Looping

Scilab code Exa 6.1 evaluate the equation

```
1 //          Example 6.1
2 //A program to evaluate the equation  $y=x^n$  when n is
   a non-negative integer.
3
4 x=input("Enter the value of x:");
5 n=input("Enter the value of n:");
6 y=1.0; count=1;    //initialization
7 //Loop begins
8 while(count<=n)    //Testing
9     y=y*x;
10    count=count+1; //Incrementing
11 end
12 //End of loop
13 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

```

-->exec('C:\Users\Devender Thakur\Desktop\scilab\scilab programs\6.lopping\exm6_1.sce', -1)
Enter the value of x:2.5
Enter the value of n:4
x = 2.500000; n = 4; x to power n = 39.062500

```

Figure 6.1: evaluate the equation

```

1 //          Case Study:-Chapter 6, Page No
   :176
2 //          1.Table of Binomial Coefficients
3
4 MAX=10;
5 printf("mx");
6 for m=0:10
7     printf("%4d",m);
8 end
9 printf("\n
   -----\n
   ");
10 m=0;
11 //print the table of binomial coefficients for m=10
12 //Computation using while loop
13 while(m<=MAX)
14     printf("%2d",m);
15     x=0;
16     binom=1;
17     while(x<=m)
18         if(m==0|x==0)
19             printf("%4d",binom); //Print the
   result i.e. binom
20         else
21             binom=binom*(m-x+1)/x; //compute the
   binomial coefficient
22             printf("%4d",binom); //Print the
   result i.e. binom
23         end
24         x=x+1;
25     end
26     printf("\n");

```



```

-->exec('C:\Users\Devender Thakur\Desktop\scilab\scilab programs\6.lopping\Casestudy6_1.sce', -1)
mx  0  1  2  3  4  5  6  7  8  9 10
-----
0  1
1  1  1
2  1  2  1
3  1  3  3  1
4  1  4  6  4  1
5  1  5 10 10  5  1
6  1  6 15 20 15  6  1
7  1  7 21 35 35 21  7  1
8  1  8 28 56 70 56 28  8  1
9  1  9 36 84 126 126 84 36  9  1
10 1 10 45 120 210 252 210 120 45 10  1
-----

```

Figure 6.2: Case study 1 Table of Binomial Coefficients

```

27     m=m+1;
28 end
29 printf("
-----\n
");
-----

```

Scilab code Exa 6.2 Multiplication table

```

1 //           Example 6.2
2 //A program to print multiplication table from 1*1
  to 12*10.
3
4
5 disp("           MULTIPLICATION TABLE           ");
6 disp("-----");
7 row=1;
8 while(row<=12)           //Outer loop begins
9     column=1;
10    while(column<=10)     //Inner loop begins
11        y=row*column;

```

```
-->exec('C:\Users\Devender Thakur\Desktop\scilab\scilab programs\6.lopping\exm6_2.sce', -1)
```

MULTIPLICATION TABLE

1	2	3	4	5	6	7	8	9	10
2	4	6	8	10	12	14	16	18	20
3	6	9	12	15	18	21	24	27	30
4	8	12	16	20	24	28	32	36	40
5	10	15	20	25	30	35	40	45	50
6	12	18	24	30	36	42	48	54	60
7	14	21	28	35	42	49	56	63	70
8	16	24	32	40	48	56	64	72	80
9	18	27	36	45	54	63	72	81	90
10	20	30	40	50	60	70	80	90	100
11	22	33	44	55	66	77	88	99	110
12	24	36	48	60	72	84	96	108	120

Figure 6.3: Multiplication table

```
12         printf("%4d", y);
13         column=column+1;
14     end
15     row=row+1;
16     printf("\n");
17 end
```

Scilab code Exa 6.2cs Case study 2 Histogram

```
1 //          Case Study:-Chapter 6
2 //          2. Histogram
3
4 N=5;
5 for n=1:N
6     printf("Enter employees in Group-%d:", n);
7     value(n)=scanf("%d");           //Read data in
        the array named value
8     printf("%d\n", value(n));       //Print number
```

```

          which is at position n
9  end
10 printf("\n");
11 printf("          |\n");
12 //Computation using for loop and draw a histogram
13 for n=1:N
14     for i=1:3
15         if(i==2) then
16             printf("Group-%1d |",n);
17         else
18             printf("          |");
19         end
20         for j=1:value(n)
21             printf("*");
22         end
23         if(i==2)
24             printf("( %d)\n",value(n));
25         else
26             printf("\n");
27         end
28     end
29     printf("          |\n");
30 end

```

Scilab code Exa 6.3 Uses a for loop

```

1 //          Example 6.3
2 //The program uses a for loop to print the "Power of
   2" table for the
3 //power 0 to 20, both positive and negative.
4
5 disp("-----");
6 printf(" 2 to power n          n          2 to power -n\n")

```

```

-->exec('C:\Users\Devender Thakur\Desktop\scilab\scilab programs\6.lopping\Casestudy6_2.sce', -1)
Enter employees in Group-1:
-->12
12
Enter employees in Group-2:
-->23
23
Enter employees in Group-3:
-->35
35
Enter employees in Group-4:
-->20
20
Enter employees in Group-5:
-->11
11

```

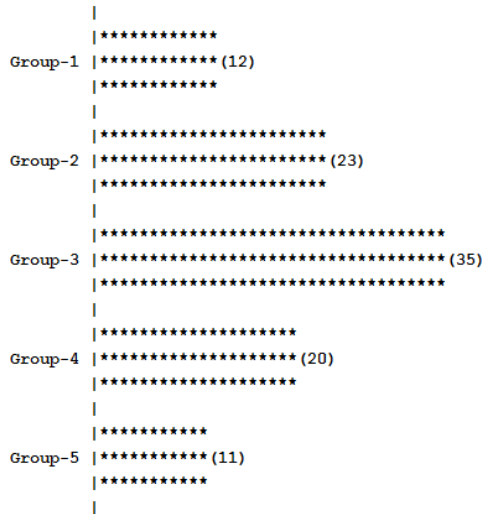


Figure 6.4: Case study 2 Histogram

```
-->exec('C:\Users\Devender Thakur\Desktop\scilab\scilab programs\6.lopping\exm6_3.sce', -1)
```

```
-----
2 to power n      n      2 to power -n
-----
   1           0      1.000000000000
   2           1      0.500000000000
   4           2      0.250000000000
   8           3      0.125000000000
  16           4      0.062500000000
  32           5      0.031250000000
  64           6      0.015625000000
 128           7      0.007812500000
 256           8      0.003906250000
 512           9      0.001953125000
1024          10      0.000976562500
2048          11      0.000488281250
4096          12      0.000244140625
8192          13      0.000122070313
16384         14      0.000061035156
32768         15      0.000030517578
65536         16      0.000015258789
131072        17      0.000007629395
262144        18      0.000003814697
524288        19      0.000001907349
1048576       20      0.000000953674
```

Figure 6.5: Uses a for loop

```

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

Scilab code Exa 6.3cs Case study 3 Minimum Cost

```
1 // Case Study:-Chapter 6
```

```
-->exec('C:\Users\Devender Thakur\Desktop\scilab\scilab programs\6.lopping\Casestudy6_3.sce', -1)
MINIMUM COST=24.00 AT p=4.1
```

```
-->
```

Figure 6.6: Case study 3 Minimum Cost

```
2 //          3.Minimum Cost
3
4 for p=0:0.1:10
5     cost=48-8*p+p^2;
6     if(p==0) ,
7         cost1=cost;
8         continue;          //Use of continue statement
9     end
10
11     if(cost>=cost1) ,
12         break;          //Use of break statement
13     end
14     cost1=cost;
15     p1=p;
16 end
17 p =(p+p1)/2.0;
18 cost=40-8*p+p^2;          //Computes the cost
19 //print the result
20 printf("MINIMUM COST=%0.2 f AT p=%0.1 f\n", cost , p);
```

Scilab code Exa 6.4 Read the marks and print total marks

```
1 //          Example 6.4
2 //A class of n students take an annual examination
   in m subjects.
3 // A program to read the marks obtained by each
   student in various subjects
```

```

4 // and to compare and print the total marks obtained
   by each of them.
5
6 FIRST=360;SECOND=240;
7 disp("Enter number of students and subjects");
8 [n,m]=scanf("%d %d");
9 for i=1:n
10     roll_number=input("Enter roll_number:");
11     total=0;
12     printf("Enter marks of %d subjects for ROLL NO
        %d",m,roll_number);
13     printf("[Enter each in newline]");
14     for j=1:m
15         marks=scanf("%d");
16         total=total+marks; //Compute the total
17     end
18     //print the total marks
19     printf("TOTAL MARKS =%d",total);
20     //Test for division and display it
21     if(total>=FIRST) ,
22         disp("First Division");
23     elseif(total>=SECOND)
24         disp("Second Division");
25     else
26         disp("***F A I L ***")
27     end
28 end

```

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

```

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

```

```

-->exec('C:\Users\Devender Thakur\Desktop\scilab\scilab programs\6.lopping\exm6.4.sce', -1)

Enter number of students and subjects
-->3 6
Enter roll_number:8701
Enter marks of 6 subjects for ROLL NO 8701[Enter each in newline]
-->81
-->75
-->83
-->45
-->61
-->59
TOTAL MARKS =404
First Division
Enter roll_number:8702
Enter marks of 6 subjects for ROLL NO 8702[Enter each in newline]
-->51
-->49
-->55
-->47
-->65
-->41
TOTAL MARKS =308
Second Division
Enter roll_number:8704
Enter marks of 6 subjects for ROLL NO 8704[Enter each in newline]
-->40
-->19
-->31
-->47
-->39
-->25
TOTAL MARKS =201
***F A I L ***

```

Figure 6.7: Read the marks and print total marks


```

3
4 a=0.4;
5 printf("          y----->
          \n");
6 printf("0
-----\
n");
7 for x=0:0.25:4
8 //Evaluation of functions
9 y1=int32(50*exp(-a*x)+0.5);
10 y2=int32(50*exp(-a*(x^2)/2)+0.5);
11 //plotting when y1=y2
12 if(y1==y2) ,
13     if(x==2.5) ,
14         printf("x |");
15     else
16         printf(" |");
17     end
18     for i=1:(y1-1)
19         printf(" ");
20     end
21     printf("#\n");
22     continue;
23 end
24
25 //Plotting when y1>y2
26 if(y1>y2) ,
27     if(x==2.5) ,
28         printf("x |");
29     else
30         printf(" |");
31     end
32     for i=1:y2-1
33         printf(" ");
34     end
35     printf("*");
36     for i=1:(y1-y2-1)
37         printf("-");

```

```

38         end
39         printf("0\n");
40         continue;
41     end
42
43     //Plotting when y2>y1
44     if(y2>y1) ,
45         if(x==2.5)
46             printf("x |");
47         else
48             printf(" |");
49         end
50         for i=1:(y1-1)
51             printf(" ");
52         end
53         printf("0");
54         for i=1:(y2-y1-1)
55             printf("-");
56         end
57         printf("*\n");
58     end
59 end
60     printf(" |\n");

```

Scilab code Exa 6.5 Use of the break statement

```

1 //           Example 6.5
2 //Program illustrate use of the break statement
3
4 disp("This program computes the avarage of set of
   numbers");
5 disp("Enter values and enter a NEGATIVE value at the
   end");

```

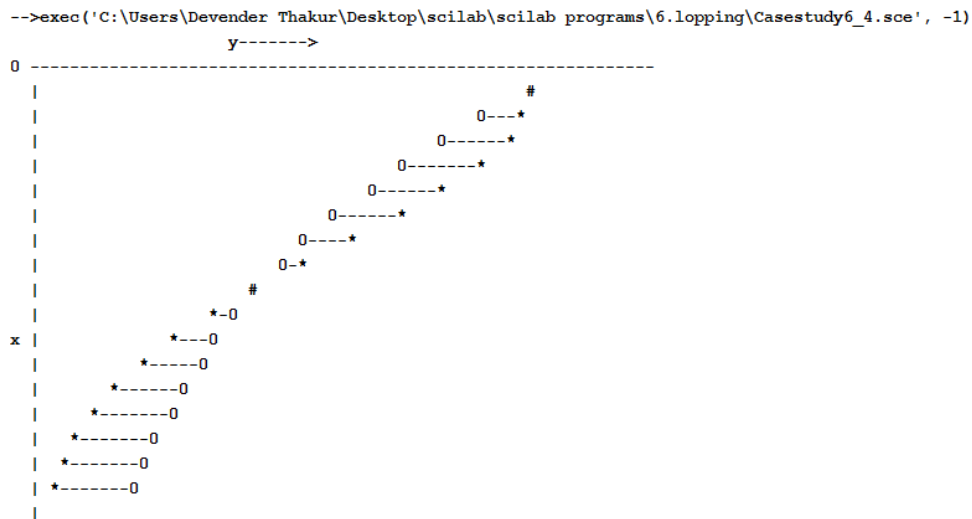


Figure 6.8: Case study 4 Plotting of two Functions

```

6 sum1=0;
7 for m=1:1000
8     x=scanf("%f"); //Read data
9     if(x<0) then
10        break; //EXIT FROM LOOP
11    end
12    sum1=sum1+x; //Computes sum
13 end
14 average=sum1/(m-1); //Computes Average
15 //Print the results
16 printf("Number of values =%d\n",m-1);
17 printf("sum1=%f\n",sum1);
18 printf("Avarage =%f\n",average);

```

Scilab code Exa 6.6 Evaluate the series

```
1 // Example 6.6
```

```

-->exec('C:\Users\Devender Thakur\Desktop\scilab\scilab programs\6.lopping\exm6_5.sce', -1)

This program computes the avarage of set of numbers

Enter values and enter a NEGATIVE value at the end
-->21
-->23
-->24
-->22
-->26
-->22
-->-1
Number of values =6
sum1=138.000000
Avarage =23.000000

```

Figure 6.9: Use of the break statement

```

2 //Program to evaluate the series i.e.
3 //  $1/1-x = 1+x+x^2+x^3+\dots+x^n$ 
4
5 x=input("Input value of x:"); //Read value of x
6 LOOP=100; ACCURACY=0.0001; //Initialization
7 sum1=0; term=1; flag=0;
8 //Computation using for loop
9 for n=1:LOOP
10     sum1=sum1+term;
11     if(term<=ACCURACY) then //Test for accuracy
12         flag=1;
13         break;
14     end
15     term=term*x;
16
17 end
18 //Print the results
19 if(flag==1) then
20     printf(" EXIT FROM LOOP\n");
21     printf(" Sum =%f ; No. of terms =%d",sum1,n);
22 else
23     disp("FINAL VALUE OF N IS NOT SUFFICIENT TO
24         ACHIEVE DESIRED ACCURCY");
25 end

```

```

-->exec('C:\Users\Devender Thakur\Desktop\scilab\scilab programs\6.1opping\exm6_6.sce', -1)
Input value of x:.21
EXIT FROM LOOP
Sum =1.265800 ; No. of terms =7
-->exec('C:\Users\Devender Thakur\Desktop\scilab\scilab programs\6.1opping\exm6_6.sce', -1)
Input value of x:.75
EXIT FROM LOOP
Sum =3.999774 ; No. of terms =34
-->exec('C:\Users\Devender Thakur\Desktop\scilab\scilab programs\6.1opping\exm6_6.sce', -1)
Input value of x:.99

FINAL VALUE OF N IS NOT SUFFICIENT TO ACHIEVE DESIRED ACCURCY

```

Figure 6.10: Evaluate the series

Scilab code Exa 6.7 Use of continue statement

```

1 //           Example 6.7
2 //The program illustrate the use of continue
  statement
3
4 disp("Enter 9999 to STOP");
5 count=0;
6 negative=0;
7 while(count<=100)
8     number=input("Enter a number:");
9     if(number==9999) then
10         break;           //EXIT FROM THE LOOP
11     end
12     if(number<0),
13         disp("Number is negative");
14         negative =negative+1;
15         continue;       //SKIP REST OF LOOP
16     end
17     sqrot=sqrt(number); //COMPUTE SQUARE ROOT

```

```

-->exec('C:\Users\Devender Thakur\Desktop\scilab\scilab programs\6.1opping\exm6_7.sce', -1)

Enter 9999 to STOP
Enter a number:25.0
Number = 25.000000
Square root = 5.000000
Enter a number:40.5
Number = 40.500000
Square root = 6.363961
Enter a number:-9

Number is negative
Enter a number:16
Number = 16.000000
Square root = 4.000000
Enter a number:-14.75

Number is negative
Enter a number:80
Number = 80.000000
Square root = 8.944272
Enter a number:9999
Number of items done = 4
Negative items = 2

END OF DATA

```

Figure 6.11: Use of continue statement

```

18     printf("Number = %f\n",number);
19     printf("Square root = %f",sqrot);
20     count=count+1;
21 end
22 //PRINT RESULTS
23 printf("Number of items done = %d\n",count);
24 printf("Negative items = %d\n",negative);
25 disp("END OF DATA");

```

Chapter 7

Arrays

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

```
1 //          Case Study: Chapter 7, Page No:210
2 //          1. Median of list of numbers
3
4 N=10;
5 disp("Enter the number of items");
6 n=scanf("%d");
7 //Reading items into array a
8 printf("Input %d values [One at a time]\n",n);
9 for i=1:n
10     a(i)=scanf("%f");
11 end
12 //Sorting begins
13 for i=1:n-1
14     for j=1:n-i
15         if(a(j)<=a(j+1))
16             t=a(j);
17             a(j)=a(j+1);
18             a(j+1)=t;
19         else
20             continue;
21         end
```

```

-->exec('C:\Users\Devender Thakur\Desktop\scilab\scilab programs\7.Array\Casestudy7_1.sce', -1)

Enter the number of items
-->5
Input 5 values[One at a time]
-->1.111
-->2.222
-->3.333
-->4.444
-->5.555
5.555000 4.444000 3.333000 2.222000 1.111000
Median is 3.333000

```

Figure 7.1: Case study 1 Median of list of numbers

```

22 end
23 end //sorting ends
24 //Calculation of median
25 if(n/2==0) then
26     median1=(a(n/2)+a(n/2+1))/2.0;
27 else
28     median1=a(n/2+1);
29 end
30 //Printing
31 for i=1:n
32     printf("%f  ",a(i));
33 end
34 printf("\nMedian is %f\n",median1);

```

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

```

1 //          Case Study: Chapter-7
2 //          2. Calculation of standard deviation
3
4 MAXSIZE=100;
5 sum1=0;n=0;sumsqr=0;
6 disp("Input values:input -1 to end");

```



```

-->exec('C:\Users\Devender Thakur\Desktop\scilab\scilab programs\7.Array\Casestudy7_2.sce', -1)

Input values:input -1 to end
-->65
-->9
-->27
-->78
-->12
-->20
-->33
-->49
-->-1
Number of items: 8
Mean: 36.625000
Standard deviation: 23.510304

```

Figure 7.2: Case study 2 Calculation of standard deviation

```

7 for i=1:MAXSIZE
8     value(i)=scanf("%f"); //Entering values in the
        array named value
9     if(value(i)==-1)
10        break;
11    end
12    sum1=sum1+value(i);
13    n=n+1;
14 end
15 mean1=sum1/n; //Computes mean
16 for i=1:n
17     deviation=value(i)-mean1;
18     sumsqr=sumsqr+deviation^2;
19 end
20 variance1=sumsqr/n;
21 stddeviation=sqrt(variance1); //Computes
        standard deviation
22 //Printing items,Mean and Standard deviation
23 printf("Number of items: %d\n",n);
24 printf("Mean: %f\n",mean1);
25 printf("Standard deviation: %f\n",stddeviation);

```

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

```
1 //          Case Study: Chapter-7
2 //          3.Evaluating a Test
3
4 STUDENTS=3;
5 ITEMS=5;
6 //Reading of correct answers
7 printf("Input key to the items\n");
8 for i=1:ITEMS
9     key(i)=read(%io(1),1,1,'(a)'); //Read data using
        read function
10    // key(i)=scanf("%c"); It can be used to read
        data
11 end
12 //Evaluation begins
13
14 for student=1:STUDENTS
15     //Reading students responses and counting
        correct ones
16     count=0;
17     printf("\nInput responses of student-%d",student
        );
18     for i=1:ITEMS
19         response(i)=scanf("%c");
20     end
21     correct=zeros(1,ITEMS)
22 //Commented code can be used to replace above line i
        .e. correct=zeros(1,ITEMS)
23     //for i=1:ITEMS
24     //     correct(i)=0;
25     //end
26     for i=1:ITEMS
27         if(response(i)==key(i)) then
```

```

28         count=count+1;
29         correct(i)=1;
30     end
31 end
32 //Printing of results
33 printf(" Student-%d\n",student);
34 printf(" Score is %d out of %d\n",count,ITEMS);
35 printf(" Response to the items below are wrong\n"
    );
36 n=0;
37 for i=1:ITEMS
38     if(correct(i)==0)
39         printf(" %d",i);
40         n=n+1;
41     end
42 end
43 if(n==0) then
44     printf(" NIL\n");
45 end
46 end

```

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

```

1 //           Case Study: Chapter-7
2 //           4. Production and sales analysis
3
4 //Input Data
5 disp(" Enter products manufactured week_wise");
6 disp(" M11, M12,--,M21, M22,-- etc");
7 for i=1:2
8     for j=1:5
9         M(i, j)=scanf("%d");
10    end

```

```

-->exec('C:\Users\Devender Thakur\Desktop\scilab\scilab programs\7.Array\casestudy7_3.sce', -1)
Input key to the items
-->a
-->b
-->c
-->d
-->a

Input responses of student-1
-->a
-->b
-->c
-->d
-->a
Student-1
Score is 5 out of 5
Response to the items below are wrong
NIL

Input responses of student-2
-->a
-->w
-->r
-->d
-->a
Student-2
Score is 3 out of 5
Response to the items below are wrong
  2  3
Input responses of student-3
-->a
-->a
-->a
-->a
-->a
Student-3
Score is 2 out of 5
Response to the items below are wrong
  2  3  4
-->

```

Figure 7.3: Case study 3 Evaluating a Test

```

11 end
12 disp("Enter products sold week_wise");
13 disp("S11 ,S12,--,S21 ,S22,--etc");
14 for i=1:2
15     for j=1:5
16         S(i,j)=scanf("%d");
17     end
18 end
19 disp("Enter cost of each product");
20 for j=1:5
21     C(j)=scanf("%d");
22 end
23 //Values matrices of production and sales
24 for i=1:2
25     for j=1:5
26         Mvalue(i,j)=M(i,j)*C(j);
27         Svalue(i,j)=S(i,j)*C(j);
28     end
29 end
30 //Total value of weekly production and sales
31 for i=1:2
32     Mweek(i)=0;
33     Sweek(i)=0;
34     for j=1:5
35         Mweek(i)=Mweek(i)+Mvalue(i,j);
36         Sweek(i)=Sweek(i)+Svalue(i,j);
37     end
38 end
39 //Monthly value of product_wise production and sales
40 for j=1:5
41     Mproduct(j)=0;
42     Sproduct(j)=0;
43     for i=1:2
44         Mproduct(j)=Mproduct(j)+Mvalue(i,j);
45         Sproduct(j)=Sproduct(j)+Svalue(i,j);
46     end
47 end
48 //Grand total of production and sales values

```

```

49 Mtotal=0;Stotal=0;
50 for i=1:2
51     Mtotal=Mtotal+Mweek(i);
52     Stotal=Stotal+Sweek(i);
53 end
54
55 //*****
56 //Selection and printing of information required
57 //*****
58 disp("Following is the list of things you request
    for");
59 disp("Enter appropriate number and press return key"
    );
60
61 disp("1.Value matrices of production and sales");
62 disp("2.Total value of weekly production and sales")
    ;
63 disp("3.Production-wise monthly value of production
    and sales");
64 disp("4.Grand total value of production and sales");
65 disp("5.Exit")
66
67 number=0;
68 while(1)
69     //Begining of while loop
70     number=input("ENTER YOUR CHOICE:");
71     if(number==5) then
72         disp("Good Bye");
73         break;
74     end
75     select number
76         //Value Matices
77     case 1 then
78         disp("VALUE MATRIX OF PRODUCTION");
79         for i=1:2
80             printf("Week(%d)\t",i);
81             for j=1:5
82                 printf("%7d",Mvalue(i,j));

```

```

83         end
84         printf("\n");
85     end
86     disp("VALUE MATRIX OF SALES");
87     for i=1:2
88         printf("Week(%d)\t",i);
89         for j=1:5
90             printf("%7d",Svalue(i,j));
91         end
92         printf("\n");
93     end
94     //Weekly Analysis
95     case 2 then
96         disp("TOTAL WEEKLY PRODUCTION AND SALES");
97         disp("          PRODUCTION          SALES");
98         disp("          -----          -----");
99         for i=1:2
100             printf("Week(%d)\t",i);
101             printf("%7d\t%9d\n",Mweek(i),Sweek(i));
102         end
103         //Product wise Analysis
104         case 3 then
105             disp("PRODUCTWISE TOTAL PRODUCTION AND SALES
106                 ");
107             disp("          PRODUCTION          SALES");
108             disp("          -----          -----");
109             for i=1:5
110                 printf("Product(%d)\t",i);
111                 printf("%7d\t%7d\n",Mproduct(i),Sproduct
112                     (i));
113             end
114             //Grand Totals
115             case 4 then
116                 disp("GRAND TOTAL OF PRODUCTION AND SALES");
117                 printf(" Total production = %d\n",Mtotal);
118                 printf(" Total sales = %d\n",Stotal);
119             //Default
120         else

```

```
119         printf("Wrong choicce ,select again\n");
120     end //End of select
121 end //End of while
122 disp("Exit from the program");
```

Scilab code Exa 7.1 Sum of squares of 10 numbers

```
1 //           Example:7.1
2 //Write a program using single-subscripted variable
  to evaluate:
3 // sum of squares of 10 numbers.The values x1,x2,...
  are read from the terminal.
4
5 //Reading values into array
6 disp("ENTER 10 REAL NUMBERS[Each in newline]");
7 total=0;
8 for i=1:10
9     x(i)=input(" ");
10    total=total+x(i)^2; //Computation of total
11 end
12 //Printing of x(i) values and total
13 for i=1:10
14     printf("x(%2d) =%5.2 f\n",i,x(i));
15 end
16 printf(" Total =%.2 f",total);
```

Scilab code Exa 7.2 Count the number of students

```
1 //           Example:7.2
```



```
-->exec('C:\Users\Devender Thakur\Desktop\scilab\scilab programs\7.Array\Casestudy7_4.sce', -1)
```

```
Enter products manufactured week_wise
```

```
M11,M12,--,M21,M22,--etc
```

```
-->11
```

```
-->15
```

```
-->12
```

```
-->14
```

```
-->13
```

```
-->13
```

```
-->13
```

```
-->13
```

```
-->15
```

```
-->12
```

```
Enter products sold week_wise
```

```
S11,S12,--,S21,S22,--etc
```

```
-->10
```

```
-->13
```

```
-->9
```

```
-->12
```

```
-->11
```

```
-->12
```

```
-->10
```

```
-->12
```

```
-->14
```

```
-->10
```

```
Enter cost of each product
```

```
-->10
```

```
-->20
```

```
-->30
```

```
-->15
```

```
-->25
```

```
Following is the list of things you request for
```

```
Enter appropriate number and press return key
```

```
1.Value matrices of production and sales
```

```
2.Total value of weekly production and sales
```

```
3.Production_wise monthly value of production and sales
```

```
4.Grand total value of production and sales
```

```
5.Exit
```

```
ENTER YOUR CHOICE:1
```

```
VALUE MATRIX OF PRODUCTION
```

```
Week(1) 110 300 360 210 325
```

```
Week(2) 130 260 390 225 300
```

```
VALUE MATRIX OF SALES
```

```
Week(1) 100 260 270 180 275
```

```
Week(2) 120 200 360 210 250
```

```
ENTER YOUR CHOICE:2
```

```
TOTAL WEEKLY PRODUCTION AND SALES
```

```
PRODUCTION SALES
```

```
-----
```

```
Week(1) 1305 1085
```

```
Week(2) 1305 1140
```

```
ENTER YOUR CHOICE:3
```

```
PRODUCTWISE TOTAL PRODUCTION AND SALES
```

```
-->exec('C:\Users\Devender Thakur\Desktop\scilab\scilab programs\7.Array\exm7_1.sce', -1)

ENTER 10 REAL NUMBERS[Each in newline]
1.1
2.2
3.3
4.4
5.5
6.6
7.7
8.8
9.9
10.10
x( 1) = 1.10
x( 2) = 2.20
x( 3) = 3.30
x( 4) = 4.40
x( 5) = 5.50
x( 6) = 6.60
x( 7) = 7.70
x( 8) = 8.80
x( 9) = 9.90
x(10) =10.10
Total =446.86
-->
```

Figure 7.5: Sum of squares of 10 numbers

```

2 //Given below is the list of marks obtained by a
   class of 50 students in an
3 //annual examination. 43 65 51 27 79 11 56 61 82 09
   25 36 07 49 55 63 74 81 49
4 //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
5 //49 68 93 85 59
6 //Write a program to count the number of students
   belonging to each of
7 // following groups of marks
   :0-9,10-19,20-29,.....100.
8
9 //This program computes for 10 students. We could
   compute for 50 students by
10 //changing MAXVAL=50.
11
12 MAXVAL=10; COUNTER=11;
13 disp("Input Data[Marks of 10 students]");
14 group1=zeros(1,11);
15 //Reading and counting
16 for i=1:MAXVAL
17     //Reading of values
18     value(i)=input(" ");
19     //Counting frequency of groups
20     a=int16((value(i)/10));
21     if(a==0) then
22         group1(a+1)=group1(a+1)+1;
23     else
24         group1(a+1)=group1(a+1)+1;
25     end
26
27 end
28 //Printing of frequency table
29 printf("Group          Range          Frequency\n");
30 for i=0:COUNTER-1
31     if(i==0) ,
32         low=0;
33     else

```

```
-->exec('C:\Users\Devender Thakur\Desktop\scilab\scilab programs\7.Array\exm7_2.sce', -1)

Input Data[Marks of 10 students]
43
65
51
27
79
11
56
61
82
09
Group      Range      Frequency
1          0 to 9      1
2          10 to 19     1
3          20 to 29     1
4          30 to 39     0
5          40 to 49     1
6          50 to 59     2
7          60 to 69     2
8          70 to 79     1
9          80 to 89     1
10         90 to 99     0
11        100 to 100    0
```

Figure 7.6: Count the number of students

```
34         low=i*10;
35     end
36     if(i==10),
37         high=100;
38     else
39         high=low+9;
40     end
41     printf("%2d %8d to %3d %5d\n",i+1,low,high,
42         group1(i+1));
42 end
```

Scilab code Exa 7.3 Compute and print

```

1 //                               Example:7.3
2 //Write a program using two dimensional array to
   compute print following
3 //information from the table of data discussed:
4 //(a)Total value of sales by each girl.
5 //(b)Total value of each item sold
6 //(c)Grand total of all sales of all items by all
   girls.
7
8 MAXGIRLS=4;MAXITEMS=3;
9 frequency=zeros(1,5);
10 disp("Input data");
11 //Reading values and computing girl_total
12 disp("Enter values ,one at a time");
13
14 for i=1:MAXGIRLS
15     girl_total(i)=0;
16     for j=1:MAXITEMS
17         value(i,j)=scanf("%d");
18         girl_total(i)=girl_total(i)+value(i,j);
19     end
20 end
21 //Computing item total
22 for j=1:MAXITEMS
23     item_total(j)=0;
24     for i=1:MAXGIRLS
25         item_total(j)=item_total(j)+value(i,j);
26     end
27 end
28 //Computing grand total
29 grand_total=0;
30 for i=1:MAXGIRLS
31     grand_total=grand_total+girl_total(i);
32 end
33 //Printing of result
34 disp("GIRLS TOTALS");
35 for i=1:MAXGIRLS
36     printf("Salesgirl (%d)=%d\n",i,girl_total(i));

```

```

-->exec('C:\Users\Devender Thakur\Desktop\scilab\scilab programs\7.Array\exm7_3.sce', -1)

Input data

Enter values, one at a time
-->310
-->275
-->365
-->210
-->190
-->325
-->405
-->235
-->240
-->260
-->300
-->380

GIRLS TOTALS
Salesgirl(1)=950
Salesgirl(2)=725
Salesgirl(3)=880
Salesgirl(4)=940

ITEM TOTALS
Item(1)=1185
Item(2)=1000
Item(3)=1310
Grand Total=3495

```

Figure 7.7: Compute and print

```

37 end
38
39 disp("ITEM TOTALS");
40 for j=1:MAXITEMS
41     printf("Item (%d)=%d\n", j, item_total(j));
42 end
43 printf("Grand Total=%d\n", grand_total);

```

Scilab code Exa 7.4 Multiplication table

```
-->exec('C:\Users\Devender Thakur\Desktop\scilab\scilab programs\7.Array\exm7_4.sce', -1)
```

```
MULTIPLICATION TABLE
*|  1  2  3  4  5
-----
1|  1  2  3  4  5
2|  2  4  6  8 10
3|  3  6  9 12 15
4|  4  8 12 16 20
5|  5 10 15 20 25
```

Figure 7.8: Multiplication table

```
1 //                               Example:7.4
2 //Write a program to compute and print a
   multiplication table for numbers 1 to 5
3
4 ROWS=5; COLUMNS=5;
5 disp("MULTIPLICATION TABLE");
6 printf(" *|")
7 for j=1:COLUMNS
8     printf("%4d",j);
9 end
10 disp(" -----");
11 for i=1:ROWS
12     printf("%1d|",i);
13     for j=1:COLUMNS
14         product(i,j)=i*j;           //Calculate the
           product
15     printf("%4d",product(i,j));    //Print the product
16     end
17     printf("\n");
18 end
```

Scilab code Exa 7.5 Popularity of various cars

```

1 //                               Example:7.5
2 //A survey to know the popularity of four cars(
   Ambassador,fait ,Dolphin and
3 //Maruti) was conducted in four cities(Bombay,
   Calcutta ,Delhi and Madras).
4 //Each person surveyed was asked to give his city
   and type of car he was using.
5 //Write a program to prouce a table showing the
   popularity of various cars in
6 //four cities.
7
8 frequency=zeros(5,5);
9 printf("For each person ,enter the city code [B,C,D,M
   ]\n");
10 printf("followed by the car code [1,2,3,4].\n");
11 printf("Enter the letter X 0(zero)to indicate end.\n
   ");
12
13 //Tabulation begins
14 for i=1:99
15     [n,city,car]=mscanf("%c %d");
16     if(city=='X') then
17         break;
18     end
19     select city
20     case 'B' then frequency(1,car)=frequency(1,car)
        +1;
21     case 'C' then frequency(2,car)=frequency(2,car)
        +1;
22     case 'D' then frequency(3,car)=frequency(3,car)
        +1;
23     case 'M' then frequency(4,car)=frequency(4,car)
        +1;
24     end
25
26 end
27
28 //Tabulation completed and Printing begins

```



```

29 disp("                POPULATORY TABLE");
30 printf("
    ----- \n"
    );
31 printf(" City          Ambassadeor   fait   Dolphin   Maruti
    \n");
32 printf("
    ----- \n"
    );
33
34 for i=1:4
35     select i
36     case 1 then printf(" Bombay  ");
37     case 2 then printf(" Calcutta");
38     case 3 then printf(" Delhi   ");
39     case 4 then printf(" Madras  ");
40     end
41     for j=1:4
42     printf("%8d",frequency(i,j));
43     end
44     printf("\n");
45 end
46 printf("
    ----- ");
47 //                Printing ends

```

```

-->exec('C:\Users\Devender Thakur\Desktop\scilab\scilab programs\7.Array\exm7_5.sce', -1)
For each person,enter the city code[B,C,D,M]
followed by the car code[1,2,3,4].
Enter the letter X 0(zero)to indicate end.
-->M 1
-->C 2
-->B 1
-->D 3
-->M 2
-->B 4
-->C 1
-->D 3
-->M 4
-->B 2
-->D 1
-->C 3
-->D 3
-->M 4
-->C 1
-->X 0

          POPULATORY TABLE

-----
City      Ambasseador  fait  Dolphin  Maruti
-----
Bombay    1      1      0      1
Calcutta  2      1      1      0
Delhi     1      0      3      0
Madras    1      1      0      2
-----
-->

```

Figure 7.9: Popularity of various cars

Chapter 8

Character Arrays and Strings

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

```
1 //                               Csaе study: Chapter-8, Page No
   //                               :253
2 //                               1.Counting words in a text
3
4 characters=0;words=0;lines1=0;
5 printf("          KEY IN THE TEXT.\n");
6 printf("GIVE ONE SPACE AFTER EACH WORD.\n");
7 printf("WHEN COMPLETED,ENTER end\n");
8
9 l=' ';
10 while(l~='end')
11     l=read(%io(1),1,1,'(a)'); //Reading a line of
   //text
12     if(l=='end') then
13         break;
14     end
15     line=[ascii(l)];           //Array of ascii
   //values of line l
16     len=length(line);         //compute length of
   //line
17     for i=1:len
```

```

-->exec('C:\Users\Devender Thakur\Desktop\scilab\scilab programs\8.String\Casestudy8_1.sce', -1)
      KEY IN THE TEXT.
GIVE ONE SPACE AFTER EACH WORD.
WHEN COMPLETED,ENTER end
-->Admiration is a very short-lived passion.
-->Admiration involves a glorious obliquity of vision.
-->Always we like those who admire us but we do not
-->like those whom we admire.
-->Fools admire, but men of sense approve.
-->end
Number of lines = 5
Number of words = 36
Number of characters = 205

-->

```

Figure 8.1: Case study 1 Counting words in a text

```

18      //ascii value of ' '(i.e.space) is 32
19      if(line(i)==32) then
20          words=words+1;          //Count the number
                                   of words
21      end
22  end
23  lines1=lines1+1;                //Count the number
                                   of lines
24  characters=characters+len; //Count the number
                                   of characters
25  end
26  //Printing results
27  printf("Number of lines = %d\n",lines1);
28  //Number of lines are added to words because last
    word of each line remains-
29  //uncounted as words are incremented at the
    occurence of space.
30  printf("Number of words = %d\n",words+lines1);
31  printf("Number of characters = %d\n",characters);

```

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

```
1 //          Csaе study: Chapter-8, Page No
   :253
2 //          2.Processing of a customer list
3
4
5 CUSTOMERS=10;
6 printf("      Input names and telephone numbers\n");
7 printf(" [Names must have First, Second and Last_name
   ]\n");
8 for i=1:CUSTOMERS
9     //Read data
10    [first_name(i),second_name(i),surname(i),
      telephone(i)]=scanf("%s %s %s %s");
11    //Converting full name to surname with initials
12    l1=length(surname(i));          //Compute
      length of surname at i
13    name(i)=strncpy(surname(i),l1) ;
14    name(i)=strcat([name(i),', ']);
15    dummy(1)=part(first_name(i,1),1);
16    name(i)=strcat([name(i),dummy]);
17    name(i)=strcat([name(i),'. ']);
18    dummy(1)=part(second_name(i,1),1);
19    name(i)=strcat([name(i),dummy]);
20 end
21 //Alphabetical odering of surnames
22 for i=1:CUSTOMERS          //Outer loop
    begins
23     for j=2:CUSTOMERS-i+1    //Inner loop
        begins
24         k=strcmp(name(j-1),name(j));
25         if(k>0) then
26
27             //Swaping names
28             l1=length(name(j-1));
29             l2=length(name(j));
30             dummy=strncpy(name(j-1),l1);
```

```

31         name(j-1)=strncpy(name(j),12);
32         l3=length(dummy);
33         name(j)=strncpy(dummy,13);
34
35         //Swapping telephone numbers
36         l3=length(telephone(j-1));
37         l4=length(telephone(j));
38         dummy=strncpy(telephone(j-1),l3);
39         telephone(j-1)=strncpy(telephone(j),l4);
40         telephone(j)=strncpy(dummy,13);
41     end
42 end //Inner loop ends
43 end //Outer loop ends
44 //Printing alphabetical list
45 disp("CUSTOMER LIST IN ALPHABETICAL ORDER");
46 for i=1:CUSTOMERS
47     printf("%-20s\t %-10s\n",name(i),telephone(i));
48 end

```

Scilab code Exa 8.1 Read a series of words

```

1 //           Exampple 8.1
2 //Write a program to read a series of words from
   terminal using scanf function.
3
4 //Read data using scanf function
5 disp("Enter text:")
6 [word1,word2,word3,word4]=scanf("%s %s %s %s");
7 //Printing the results
8 printf("word1 = %s\nword2 = %s\n",word1,word2);
9 printf("word3 = %s\nword4 = %s\n",word3,word4);

```

```

-->exec('C:\Users\Devender Thakur\Desktop\scilab\scilab programs\8.String\Casestudy8_2.sce', -1)
      Input names and telephone numbers
      [Names must have First,Second and Last_name]
-->Gottfried Wilhelm Leibniz 711518
-->Joseph Louis Lagrange 869245
-->Jean Robert Argand 900823
-->Carl Freidrich Gauss 806788
-->Simon Denis Poisson 853240
-->Friedrich Wilhelm Bessel 719731
-->Charles Francois Strum 222031
-->George Gabriel Stokes 545454
-->Mohandas Karamchand Gandhi 362718
-->Josian Willard Gibbs 123145

      CUSTOMER LIST IN ALPHABETICAL ORDER
      Argand,J.R          900823
      Bessel,F.W         719731
      Gandhi,M.K         362718
      Gauss,C.F          806788
      Gibbs,J.W          123145
      Lagrange,J.L       869245
      Leibniz,G.W        711518
      Poisson,S.D        853240
      Stokes,G.G         545454
      Strum,C.F          222031

-->

```

Figure 8.2: Case study 2 Processing of a customer list

```

-->exec('C:\Users\Devender Thakur\Desktop\scilab\scilab programs\8.String\exm8_1.sce', -1)

Enter text:
-->Oxford Road, London M17ED
word1 = Oxford
word2 = Road,
word3 = London
word4 = M17ED

-->exec('C:\Users\Devender Thakur\Desktop\scilab\scilab programs\8.String\exm8_1.sce', -1)

Enter text:
-->Oxford-Road, London-M17ED United Kingdom
word1 = Oxford-Road,
word2 = London-M17ED
word3 = United
word4 = Kingdom

```

Figure 8.3: Read a series of words

Scilab code Exa 8.2 Read a line of text

```

1 //           Example 8.2
2 //Write a program to read a line of text containing
  a series of-
3 //words from the terminal.
4
5 disp("Enter text. Press <Return> at end");
6 line=read(%io(1),1,1,'(a)'); //Read a line
7 disp(line); //Display line

```

Scilab code Exa 8.3 Copy one string into another


```

-->exec('C:\Users\Devender Thakur\Desktop\scilab\scilab programs\8.String\exm8_2.sce', -1)

Enter text. Press <Return> at end
-->Programming in c is interesting.

Programming in c is interesting.

-->

```

Figure 8.4: Read a line of text

```

1 //          Example 8.3
2 //Write a program to copy one string into another
   and count the number
3 //of characters copied.
4
5 //Read data using scanf function
6 disp("Enter a string:")
7 [string2]=scanf("%s"); //Read string
8 l=length(string2); //Compute the length
9 string1=' '; //string1 is empty
10 for i=1:l
11     string1=string1+ part(string2,i);
12 end
13 //Printing the results
14 printf(" %s\n",string1);
15 printf(" Number of characters = %d\n",l);

```

Scilab code Exa 8.4 Display the string under various format specifications

```

1 //          Exampplle 8.4
2 //Write a program to store the string "United
   Kingdom" in the array country-
3 //and display the string under various format
   specifications.
4

```

```

-->exec('C:\Users\Devender Thakur\Desktop\scilab\scilab programs\8.String\exm8_3.sce', -1)

Enter a string:
-->Manchester
Manchester
Number of characters = 10

-->exec('C:\Users\Devender Thakur\Desktop\scilab\scilab programs\8.String\exm8_3.sce', -1)

Enter a string:
-->Westminster
Westminster
Number of characters = 11

-->

```

Figure 8.5: Copy one string into another

```

5
6 country= 'United Kingdom';
7 printf("\n");
8 printf(" *123456789012345*\n");
9 printf("-----\n");
10 printf("%15s\n", country);
11 printf("%5s\n", country);
12 printf("%15.6s\n", country);
13 printf("%-15.6s\n", country);
14 printf("%15.0s\n", country);
15 printf("%.3s\n", country);
16 printf("%s\n", country);
17 printf("-----\n");

```

Scilab code Exa 8.5 Program using for loop

```

1 //                               Example 8.5
2 //Write a program using for loop to print the
  following output:
3 //   C

```

```

-->exec('C:\Users\Devender Thakur\Desktop\scilab\scilab programs\8.String\exm8_4.sce', -1)

*123456789012345*
-----
  United Kingdom
United Kingdom
      United
United

Uni
United Kingdom
-----

-->

```

Figure 8.6: Display the string under various format specifications

```

4 // CP
5 // .....
6 // .....
7 // CProgramming
8 // CProgramming
9 // .....
10 // .....
11 // CPr
12 // CP
13 // C
14 string1= 'CProgramming ' ;
15 printf(" -----\n");
16 f= ' ';
17 for i=1:12
18     f=f+part(string1,i);
19
20     printf("|%-13s |\n",f);
21 end
22 printf("|-----|\n");
23 for j=0:11
24     s= ' ';
25     for i=1:12-j
26         s=s+part(string1,i);
27     end
28     printf(" |%-13s |\n",s);

```

```

29 end
30 printf(" -----");
31 //for c=0:11
32 //    d=c+1;
33 //    mprintf(" |%-12.*s |\n",d, string1);
34 //end
35 //disp("-----");
36 //for c=11:c-1:0
37 //    d=c+1;
38 //    printf(" |%-12.*s |\n",d, string1);
39 //end
40 //disp("-----");

```

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

```

1 //          Example 8.6
2 //Write a program which would print the alphabet set
3 //a to z A to Z in decimal-
4 //character form.
5 for c=65:122
6     if(c>90&c<97) then
7         continue;          //Terminate current
8         iteration
9     end
10    c1=ascii(c);           //Convert ascii value to
11    character
12    printf(" |%4d  -  %c\ ",c ,c1);
13 end
14 printf(" |\n")

```

```

-->exec('C:\Users\Devender Thakur\Desktop\scilab\scilab programs\8.String\exm8_5.sce', -1)
-----
| C      |
| CP     |
| CPr    |
| CPro   |
| CProg  |
| CProgr |
| CProgra|
| CProgram|
| CProgramm|
| CProgrammi|
| CProgrammin|
| CProgramming|
|-----|
| CProgramming|
| CProgrammin|
| CProgrammi|
| CProgramm|
| CProgram|
| CProgra|
| CProgr|
| CProg|
| CPro|
| CPr|
| CP|
| C|
-----
-->

```

Figure 8.7: Program using for loop

```

-->exec('C:\Users\Devender Thakur\Desktop\scilab\scilab programs\8.String\exm8_6.sce', -1)
| 65 - A| 66 - B| 67 - C| 68 - D| 69 - E| 70 - F| 71 - G| 72 - H| 73 - I| 74 - J| 75 - K| 76 - L| 77 - M| 78 - N|
| 79 - O| 80 - P| 81 - Q| 82 - R| 83 - S| 84 - T| 85 - U| 86 - V| 87 - W| 88 - X| 89 - Y| 90 - Z| 91 - a| 92 - b| 93
| 94 - c| 95 - d| 96 - e| 97 - f| 98 - g| 99 - h| 100 - i| 101 - j| 102 - k| 103 - l| 104 - m| 105 - n| 106 - o| 107 - p| 108
| 109 - q| 110 - r| 111 - s| 112 - t| 113 - u| 114 - v| 115 - w| 116 - x| 117 - y| 118 - z|
-->

```

Figure 8.8: Print the alphabet set a to z and A to Z

```
-->exec('C:\Users\Devender Thakur\Desktop\scilab\scilab programs\8.String\exm8_7.sce', -1)
VISWANATH PRATAP SINGH
-->|
```

Figure 8.9: Concatenate the three parts into one string

Scilab code Exa 8.7 Concatenate the three parts into one string

```
1 //           Example 8.7
2 //The name of employees of an organization are
  stored in three arrays namely–
3 //first_name ,second_name and last_name .
4 //Write a program to concatenate the three parts
  into one string called name.
5
6
7 //Store the name in the three arrays
8 first_name=[ 'VISWANATH' ];
9 second_name=[ 'PRATAP' ];
10 last_name=[ 'SINGH' ];
11
12 //Concatenate three parts into one
13 name=[first_name second_name last_name];
14 // Print the result
15 for i=1:3
16     printf("%s ",name(i));
17 end
18 //Statement below can also be used to print the
  result
19 //disp(name);
```

Scilab code Exa 8.8 Compare whether strings are equal

```
1 //                               Example 8.8
2 //s1,s2 and s3 are three string variables. Write a
  program to read two string-
3 //constants in to s1 and s2 and compare whether they
  are equal or not,join-
4 //them together. Then copy contents of s1 to
  variable s3. At the end program-
5 // should print all three variables and their
  lengths
6
7
8 //Read data
9 printf("Enter two string constants\n");
10 [s1 s2]=scanf("%s %s");
11
12 //Comparing two strings
13 x=strcmp(s1,s2);
14 if x~=0 then
15     printf("String are not equal\n");
16     //Concatenate two strings s1 and s2
17     s1=strcat([s1,s2]);
18 else
19     printf("String are equal\n");
20 end
21
22 l1=length(s1);
23 //Coping s1 to s3
24 s3=strncpy(s1,l1);
25 //finding length of strings
26 l2=length(s2);
27 l3=length(s3);
28 //Output
29 printf("s1 = %s\t length = %d characters\n",s1,l1);
30 printf("s2= %s\t length = %d characters\n",s2,l2);
31 printf("s3= %s\t length = %d characters\n",s3,l3);
```

```

-->exec('C:\Users\Devender Thakur\Desktop\scilab\scilab programs\8.String\exm8_8.sce', -1)
Enter two string constants
-->New York
String are not equal
s1 = NewYork    length = 7 characters
s2=  York      length = 4 characters
s3=  NewYork   length = 7 characters

-->exec('C:\Users\Devender Thakur\Desktop\scilab\scilab programs\8.String\exm8_8.sce', -1)
Enter two string constants
-->London London
String are equal
s1 = London    length = 6 characters
s2=  London    length = 6 characters
s3=  London    length = 6 characters

-->

```

Figure 8.10: Compare whether strings are equal

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

```

1 //                               Example 8.9
2 //Write a program that would sort a list of names in
   alphabetical order.
3
4 ITEMS=5;
5 //Reading the list
6 printf("Enter names of %d items\n",ITEMS);
7 i=1;
8 while(i<=ITEMS)
9     string1(i)=scanf("%s");
10    i=i+1;
11 end
12 //Sorting begins
13 for i=1:ITEMS                               //Outer loop begins
14     for j=2:ITEMS-i+1                       //Inner loop begins

```



```

-->exec('C:\Users\Devender Thakur\Desktop\scilab\scilab programs\8.String\exm8_9.sce', -1)
Enter names of 5 items
-->London
-->Manchester
-->Delhi
-->Paris
-->Moscow

Alphabetical list
Delhi
London
Manchester
Moscow
Paris

-->

```

Figure 8.11: Sort a list of names in alphabetical order

```

15         k=strcmp(string1(j-1),string1(j))
16         if(k>0) then
17             //Compute length and Exchange of
                contents
18             l1=length(string1(j-1));
19             l2=length(string1(j));
20             dummy=strncpy(string1(j-1),l1);
21             string1(j-1)=strncpy(string1(j),l2);
22             l3=length(dummy);
23             string1(j)=strncpy(dummy,l3);
24         end
25     end //Inner loop ends
26 end //Outer loop ends
27 //Sorting completed
28 disp(" Alphabetical list");
29 for i=1:ITEMS
30     printf("%s\n",string1(i));
31 end

```

Chapter 9

User Defined Functions

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

```
1 //          Case Study:Chapter-9[page no:310]
2 //          1. Calculation of Area under a Curve
3 funcprot(0);
4 //global variables
5 global start_point
6 global end_point;
7 global total_area;
8 global numtraps;
9 function []=input1()
10     global start_point;
11     global end_point;
12     global total_area;
13     global numtraps;
14     total_area=0;
15     start_point=input("Enter lower limit:");
16     end_point=input("Enter upper limit:");
17     numtraps=input("Enter number of trapezoids:");
18 endfunction
19 function [total_area]=find_area(a,b,n)
20     global total_area;
21     base=(b-1)/n;           //base is local
```

```

        variable
22     lower=a;           //lower is local
        variable
23     for lower=a:(lower+base):(b-base)
24         h1=function_x(lower);
25         h2=function_x(lower+base);
26         total_area=total_area+trap_area(h1,h2,base);
           //total area is calculated
27     end
28 endfunction
29 function [area]=trap_area(height_1,height_2,base)
30     area =0.5*(height_1+height_2)*base;    //area
           is local variable
31 endfunction
32 function [x] =function_x(x)
33     x=(x^2)+1;
34 endfunction
35
36 //calling functions
37 disp("AREA UNDER CURVE");
38 input1();           //calling input1
           () function
39 total_area=find_area(start_point,end_point,numtraps)
           ;//calling find_area() function
40 printf("TOTAL AREA = %f",total_area);

```

Scilab code Exa 9.1 Multiple functions

```

1 //           Example 9.1
2 //Write a program with multiple functions that do
   not communicate-
3 //data between them.
4

```

```

-->exec('C:\Users\Devender Thakur\Desktop\scilab\scilab programs\9.function\Casestudy1.sci', -1)

AREA UNDER CURVE
Enter lower limit:0
Enter upper limit:3
Enter number of trapezoids:30
TOTAL AREA = 12.002222
-->exec('C:\Users\Devender Thakur\Desktop\scilab\scilab programs\9.function\Casestudy1.sci', -1)

AREA UNDER CURVE
Enter lower limit:0
Enter upper limit:3
Enter number of trapezoids:100
TOTAL AREA = 12.000200
-->

```

Figure 9.1: Case study 1 Calculation of Area under a Curve

```

5 //Function1:prntline()
6 funcprot(0);
7 function []=prntline() //contains no argument
8     for i=1:35
9         printf("%c", '-');
10    end
11    printf("\n");
12 endfunction
13 //Function2: value()
14 function []=value() //contains no argument
15    principal=input("Principal amount?");
16    inrate=input("Interest rate?");
17    period=input("Period?");
18    sum1=principal;
19    year=1;
20    //Computation using while loop
21    while(year<=period)
22        sum1=sum1*(1+inrate); //calculates
23        principal amount after certain years
24        year=year+1;
25    end
26    printf("%8.2f %5.2f %5d %12.2f\n",principal,
27        inrate,period,sum1);
28 endfunction
29 //Calling functions

```

```

-->exec('C:\Users\Devender Thakur\Desktop\scilab\scilab programs\9.function\exm9_1.sci', -1)
-----
Principal amount? 5000
Interest rate?    0.12
Period?          5
5000.00  0.12    5    8811.71
-----

```

Figure 9.2: Multiple functions

```

28 printline();
29 value();
30 printline();

```

Scilab code Exa 9.2 Include arguments in function calls

```

1 //           Example 9.2
2 //Modify Example 9.1 to include arguments in
  function calls.
3
4 funcprot(0);
5 function []=printline(ch)           //function with
  argument ch
6     for i=1:52
7         printf("%c",ch);
8     end
9     printf("\n");
10 endfunction
11 function []=value(p,r,n)           //function with
  argument p,r,n
12     sum1=p;
13     year=1;
14     while(year<=n)
15         sum1=sum1*(1+r);
16         year=year+1;

```



```

-->exec('C:\Users\Devender Thakur\Desktop\scilab\scilab programs\9.function\exm9_3.sci', -1)
Enter principal amount,interest rate,and period
[Enter in single line seperated by space]
-->5000 0.12 5
*****
5000.000000    0.120000    5    8811.708416
=====
-->

```

Figure 9.4: Return result

```

8     printf("\n");
9 endfunction
10 function [amount]=value(p,r,n) //returns amount
11     sum1=p;
12     year=1;
13     while(year<=n)
14         sum1=sum1*(1+r);
15         year=year+1;
16     end
17     amount=sum1;
18 endfunction
19 printf("Enter principal amount,interest rate ,and
    period\n[Enter in single line seperated by space]
    ");
20 [principal,inrate,period]=scanf("%f %f %d");
21 //Calling functions
22 printline('* ',52);
23 amount=value(principal,inrate,period);
24 printf("%f\t%f\t%d\t%f\n",principal,inrate,period,
    amount);
25 printline('= ',52);

```

Scilab code Exa 9.4 Computes x raised to the power y

```

1 //          Example 9.4
2 //Write a program with function power that computes
   x raised to the power y.
3 funcprot(0);
4 function p = power(x,y)
5     p=1.0;                // x to power 0
6     if((y>=0)) then
7         while(y)        //computes positive powers
8             p=p*x;
9             y=y-1;
10        end
11    else
12        while(y)        //computes negative powers
13            p=p/x;
14            y=y+1;
15        end
16    end
17 endfunction
18 disp("Enter x,y:");
19 disp("[Enter in single line seperated by space] ");
20 [x,y]=scanf("%d %d");    //input using scanf
   function
21
22 //calling power() function and printing its output
23 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

```

1 //          Example 9.5
2 //Write a program to calculate standard deviation of
   an array values.
3 //Array elements are read from terminal.Use
   functions to calculate-

```



```

-->exec('C:\Users\Devender Thakur\Desktop\scilab\scilab programs\9.function\exm9_4.sci', -1)

Enter x,y:

[Enter in single line seperated by space]
-->16 2
16 to the power 2 is 256.000000
-->exec('C:\Users\Devender Thakur\Desktop\scilab\scilab programs\9.function\exm9_4.sci', -1)

Enter x,y:

[Enter in single line seperated by space]
-->16 -2
16 to the power -2 is 0.003906
-->

```

Figure 9.5: Computes x raised to the power y

```

4 //standard deviation and mean
5 funcprot(0);
6 //passing array named 'value' to function std_dev at
   'a'
7 function [std]=std_dev(a,n)
8     sum1=0;
9     x=mean1(a,n); //calling
   mean1() function
10    for i=1:n
11        sum1=sum1+(x-a(i))^2;
12        std=sqrt(sum1/double(n)); //computes
   standard deviation
13    end
14 endfunction
15 function [x]=mean1(a,n)
16     sum1=0;
17     for i=1:n
18         sum1=sum1+a(i);
19         x=sum1/double(n); //x contain
   mean value
20    end
21 endfunction
22 SIZE=int8(5); //size of
   array

```

```

-->exec('C:\Users\Devender Thakur\Desktop\scilab\scilab programs\9.function\exm9_5.sci', -1)
Enter 5 float values
35.0
67.0
79.5
14.20
55.75
Std.deviation is 23.231582
-->

```

Figure 9.6: Calculate standard deviation of an array values

```

23 printf("Enter %d float values",SIZE);
24 for i=1:SIZE
25     value(i)=input(" ");           //entering
        values in the array
26 end
27 printf("Std.deviation is %f",std_dev(value,SIZE));
        //calling std_dev() function

```

Scilab code Exa 9.6 Sort an array

```

1 //           Example 9.6
2 //Write a program that uses a function to sort an
    array of integers.
3 funcprot(0);
4 function [x]=sort(m,x)           //Passing an array i.
    e. marks to function sort()
5     for i=1:m                     // i represents number
        of passes
6         for j=2:m-i+1             // j represents
            number of comperision in each pass
7             if(x(j-1)>=x(j)) then
8                 t=x(j-1);
9                 x(j-1)=x(j);
10                x(j)=t;

```

```

-->exec('C:\Users\Devender Thakur\Desktop\scilab\scilab programs\9.function\exm9_6.sci', -1)

Marks before sorting

40 90 73 81 35

Marks after sorting

35 40 73 81 90

```

Figure 9.7: Sort an array

```

11         end
12     end
13 end
14 endfunction
15 marks=int16([40,90,73,81,35]); //creating an array
    named marks of 5 integers
16 disp("Marks before sorting");
17 disp(marks);
18 x=sort(5,marks); //calling sort()
    function
19 disp("Marks after sorting");
20 disp(x);

```

Scilab code Exa 9.7 Automatic variables

```

1 //           Example 9.7
2 //Write a multifunction how automatic variables work
3
4 funcprot(0);
5 function []=function1()
6     m=int32(10); //Local Variable
7     disp(m); //First Output
8 endfunction
9 function []=function2()

```

```

-->exec('C:\Users\Devender Thakur\Desktop\scilab\scilab programs\9.function\exm9_7.sci', -1)

10

100

1000

-->

```

Figure 9.8: Automatic variables

```

9      m= int32(100);           //Local Variable
10     function1();           //Calling function1()
11     disp(m);                //Second Output
12 endfunction
13 function []=funcmain()
14 m=int32(1000);
15 function2();               //calling function2()
16 disp(m);                   //Third output
17 endfunction
18 funcmain()                 //calling funcmain() function

```

Scilab code Exa 9.8 Global variables

```

1 //           Example 9.8
2 //Write a multifunction to illustrate the properties
  of global variables.
3 funcprot(0);
4 function [x]=fun1()
5     global x;
6     x=x+10;                 //global x
7 endfunction
8 function [x]=fun2()
9     x=1                      //Local x
10 endfunction

```

```

-->exec('C:\Users\Devender Thakur\Desktop\scilab\scilab programs\9.function\exm9_8.sci', -1)
x=10
x=20
x=1
x=30

-->

```

Figure 9.9: Global variables

```

11 function [x]=fun3()
12     global x;
13     x=x+10;           //global x
14 endfunction
15     global x;
16     x=10;
17     printf("x=%d\n",x)
18     //calling fun1(),fun2(),fun3() functions
19     printf("x=%d\n",fun1());
20     printf("x=%d\n",fun2());
21     printf("x=%d\n",fun3());

```

Scilab code Exa 9.16 Factorial of a number using recursion

```

1 //           Topic 9.16 RECURSION
2 //           Page no. 288
3 //Write a program to calculate factorial of a number
  using recursion
4 function [fact1]=factorial1(n)
5     fact1=-1
6     if(n<0) then
7         disp("Please enter positive value[i.e. 0 or
           greater than 0] ");
8         return;           //Quits the
           current function
9     end

```

```
-->exec('C:\Users\Devender Thakur\Desktop\scilab\scilab programs\9.function\topic9_16.sci', -1)
Enter number:6
Factorial of 6 = 720
-->
```

Figure 9.10: Factorial of a number using recursion

```
10     if((n==0)|(n==1)) then
11         fact1=1;
12     else
13         fact1=n*factorial1(n-1);    //recursive call
14         to factorial1()
15     end
16 endfunction
17 n=input("Enter number:");
18 //calling factorial1() function inside printf()
19 printf("Factorial of %d = %d",n,factorial1(n));
```

Chapter 10

Structures and Unions

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

```
1 //                               Case study: Chapter-10, Page No
   //                               :341
2 //                               Book Shop Inventory
3
4 funcprot(0);
5 //Defining functions
6 function [string1]=get1()
7     string1=read(%io(1),1,1,'(a)');
8 endfunction
9 function [i] =look_up(table,s1,s2,m)
10     for i=1:m
11         x=strcmp(s1,table(i).title);
12         y=strcmp(s2,table(i).author);
13         if x==0 & y==0 then
14             return i; //Book found
15         end
16     end
17     i=-1; //Book not found
18 endfunction
19
20 //Creates array of structures
```

```

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

```



```

44     quantity=input("Enter number of copies :");
45     if quantity<book(index).quantity then
46         printf("Cost of %d copies = %.2f\n",quantity,
                book(index).price*quantity);
47     else
48         printf("Required copies not in stock\n");
49     end
50 else
51     printf("Book not in list\n");
52 end
53 printf("\nDo you want any other book?(YES/NO):")
    ;
54 response=get1();a=2;
55 end
56 printf(" Thank you.      Good Bye");

```

Scilab code Exa 10.1 Define a structure type

```

1 //          Example 10.1
2 //Define a structure type,struct personal that would
   contain person name,-
3 // date of joining and salary. Write a program to
   read this information from
4 // keyboard and print same on the screen.
5
6 funcprot(0);
7 function [ ]=struc(n,d,m,y,s)
8     //Defining structure members
9     personal=struct('name',n,'day',d,'month',m,'year
   ',y,'salary',s);
10    person=personal;
11    //Accessing structure members
12 printf(" %s %d %s %d %.2f",person.name,person.day,

```

```

-->exec('C:\Users\Devender Thakur\Desktop\scilab\scilab programs\Structure\Casestudy.sci', -1)
Enter title and author name as per the list:
Title:
-->BASIC
Author:
-->Balagurusamy
Balagurusamy BASIC 30.00 January 1984 TMH
Enter number of copies :5
Required copies not in stock

Do you want any other book?(YES/NO):
-->y
Enter title and author name as per the list:
Title:
-->COBOL
Author:
-->Balagurusamy
Balagurusamy COBOL 60.00 December 1988 Macmillan
Enter number of copies :7
Cost of 7 copies = 420.00

Do you want any other book?(YES/NO):
-->y
Enter title and author name as per the list:
Title:
-->C Programming
Author:
-->Ritche
Book not in list

Do you want any other book?(YES/NO):
-->n
    Thank you.    Good Bye
-->

```

Figure 10.1: Case study 1 Book Shop Inventory

```

-->exec('C:\Users\Devender Thakur\Desktop\scilab\scilab programs\Structure\exm10_1.sce', -1)

Input values[Name day month year and salary]
-->M.L.Goel 10 January 1945 4500
M.L.Goel 10 January 1945 4500.00

```

Figure 10.2: Define a structure type

```

    person.month, person.year, person.salary);
13 endfunction
14 disp("Input values [Name day month year and salary]")
    ;
15 //Reading data
16 [name, day, month, year, salary]=scanf("%s %d %s %d %f")
    ;
17 //Calling function struc()
18 struc(name, day, month, year, salary);

```

Scilab code Exa 10.2 Comparison of structure variables

```

1 //          Example 10.2
2 // Write a program to illustrate the comparison of
  structure variables.
3
4 function []=class()
5     //Defining structures
6     student1=struct('number', 111, 'name', 'Rao', 'marks',
7     '72.50');
8     student2=struct('number', 222, 'name', 'Raddy', '
9     marks', 67.00);
10    student3=struct('number', [], 'name', [], 'marks',
11    [], []);
12    student3=student2;
13    if(student3==student2) , //Logical operation
14        on structures

```

```

-->exec('C:\Users\Devender Thakur\Desktop\scilab\scilab programs\Structure\exm10_2.sci', -1)

Student2 and student 3 are same
222 Raddy 67.000000
-->|

```

Figure 10.3: Comparison of structure variables

```

11         disp(" Student2 and student 3 are same");
12         printf(" %d %s %f", student3.number, student3.
                name, student3.marks);
13     else
14         disp(" Student2 and student 3 are not same");
15     end
16 endfunction
17 //calling function class
18 class()

```

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

```

1 //           Example 10.3
2 // Write a program to calculate the subject-wise and
  student-wise totals
3 //and store them as a part of the structue.
4
5 //Defining array of structures
6 student=[struct('sub1',45,'sub2',67,'sub3',81,'total
  ',0)
7         struct('sub1',75,'sub2',53,'sub3',69,'total
  ',0)
8         struct('sub1',57,'sub2',36,'sub3',71,'total
  ',0)
9         ];
10 total=struct('sub1',0,'sub2',0,'sub3',0,'total',0);
11

```

```

-->exec('C:\Users\Devender Thakur\Desktop\scilab\scilab programs\Structure\exm10_3.sce', -1)
STUDENT      TOTAL
student(1)    193
student(2)    197
student(3)    164
SUBJECT      TOTAL
Subject 1     177
Subject 2     156
Subject 3     221
Grand Total = 554
-->

```

Figure 10.4: Calculate the subject wise and student wise totals

```

12 //Calculate the student-wise and subject-wise totals
13 for i=1:3
14     student(i).total=student(i).sub1+student(i).sub2
        +student(i).sub3;
15     total.sub1=total.sub1+student(i).sub1;
16     total.sub2=total.sub2+student(i).sub2;
17     total.sub3=total.sub3+student(i).sub3;
18     total.total=total.total+student(i).total;
19 end
20 //Printing student-wise totals
21 printf("STUDENT      TOTAL\n");
22 for i=1:3
23     printf("student(%d)      %d\n",i,student(i).
        total);
24 end
25 //Printing subject-wise totals
26 printf("SUBJECT      TOTAL\n");
27 printf(" %s      %d\n", "Subject 1",total.sub1);
28 printf(" %s      %d\n", "Subject 2",total.sub2);
29 printf(" %s      %d\n", "Subject 3",total.sub3);
30 //Printing grand total
31 printf("Grand Total = %d",total.total);

```

Scilab code Exa 10.4 Array member to represent the three subjects

```
1 //                               Example 10.4
2 //Rewrite the program of Example 10.3 to using an
   array member to represent
3 //the three subjects.
4
5 //Defining array of structures and array with in
   structure
6 student(1)=[struct('sub',[45 67 81],'total',0)];
7 student(2)=[struct('sub',[75 53 69],'total',0)];
8 student(3)=[struct('sub',[57 36 71],'total',0)];
9 total=student;
10 for i=1:3
11     total.sub(i)=0;
12 end
13 total.total=0;
14 //Calculate the student-wise and subject-wise totals
15 for i=1:3
16     for j=1:3
17         student(i).total=student(i).total+student(i)
           .sub(j);
18         total.sub(j)=total.sub(j)+student(i).sub(j);
19     end
20     total.total=total.total+student(i).total; //
           Grand total
21 end
22 //Printing student-wise totals
23 printf("STUDENT          TOTAL\n");
24 for i=1:3
25     printf("student(%d)          %d\n",i,student(i).
           total);
26 end
27 //Printing subject-wise totals
28 printf("SUBJECT          TOTAL\n");
29 for j=1:3
30     printf("subject -(%d)          %d\n",j,total.sub(j)
           );
```

```

-->exec('C:\Users\Devender Thakur\Desktop\scilab\scilab programs\Structure\exm10_4.sce', -1)
STUDENT      TOTAL
student(1)   193
student(2)   197
student(3)   164
SUBJECT      TOTAL
subject-(1)  177
subject-(2)  156
subject-(3)  221
Grand Total = 554
-->

```

Figure 10.5: Array member to represent the three subjects

```

31 end
32 //Printing grand total
33 printf("Grand Total = %d",total.total);

```

Scilab code Exa 10.5 structure as a parameter to a function

```

1 //          Example 10.5
2 //Write a simple program to illustrate the method of
  sending an entire
3 //structure as a parameter to a function.
4
5 funcprot(0);
6 //Defining functions
7 function [item]=update(product,p,q)
8     product.price=product.price+p;
9     product.quantity=product.quantity+q;
10    item=product;
11 endfunction
12 function [value] =mul(stock)
13     value=stock.price*stock.quantity;
14 endfunction
15
16 //Creates structure item

```

```

-->exec('C:\Users\Devender Thakur\Desktop\scilab\scilab programs\Structure\exml0_5.sce', -1)
Input increment values: price increment and quantity increment
-->10 12
Updated values of items
Name      :XYZ
Price     :35.750000
Quantity  :24
Value of item = 858.000000

-->

```

Figure 10.6: structure as a parameter to a function

```

17 item=struct('name','XYZ','price',25.75,'quantity',
18           ,12);
18 //Read data
19 printf("Input increment values:");
20 printf(" price increment and quantity increment\n");
21 ;
21 [p_increment,q_increment]=scanf("%f %d");
22
23 //Calling update() and mul() functions
24 //Passing structure item to functions update() and
25 //mul()
26 item=update(item,p_increment,q_increment);
27 value=mul(item);
28 //-----
29 //Printing Results
30 printf("Updated values of items\n");
31 printf("Name      :%s\n",item.name);
32 printf("Price     :%f\n",item.price);
33 printf("Quantity  :%d\n",item.quantity);
34 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

```
1 //                               Example 12.1
2 // Write a program to read data from keyboard ,write
   it to a file called INPUT,
3 // again read the same data from the INPUT file and
   display it on the screen.
4
5 warning('off');
6 disp("Data Input");
7
8 //Open the file INPUT
9 f1=mopen('INPUT.txt','w');
10 // Get character from keyboard
11 c=read(%io(1),1,1,'(a)');
12 mfprintf(f1,'%s',c);
13
14 //close the file input
15 mclose(f1);
16
17
18 disp("Data Output");
19 //Reopen the file INPUT
```

```

-->exec('C:\Users\Devender Thakur\Desktop\scilab\scilab programs\Files\exp12_1.sce', -1)

Data Input
-->This is a program to test the file handling features on this system

Data Output
This is a program to test the file handling features on this system
-->

```

Figure 12.1: Read data from keyboard and write it to a file

```

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

```

Scilab code Exa 12.2 Read and write odd and even numbers

```

1 //                               Example12.2
2 //A file named DATA contains a series of integer
   numbers. Code a program
3 //to read these numbers and then write all 'odd'
   numbers to a file to be
4 //called ODD and all 'even' numbers to a file to be
   called EVEN.
5
6 warning('off');
7 //Input numbers in the DATA.txt file
8 printf("Contents of DATA file\n");
9 f1=mopen('DATA.txt', 'wt');
10 for i=1:30
11     number(i)=scanf("%d");
12     if(number(i) == -1)
13         break;

```

```

14     end
15     fprintf(f1, '%d\n', number(i));
16 end
17 fclose(f1);
18
19 f2=fopen('ODD.txt', 'wt');
20 f3=fopen('EVEN.txt', 'wt');
21 f1=fopen('DATA.txt', 'rt');
22 //Read numbers from DATA file
23 EOF=length(number);
24 i=1;
25 even=0;
26 odd=0;
27 while (i<EOF)
28     [n, number]=mfscanf(f1, "%d")
29     if(modulo(number, 2)==0)
30         fprintf(f3, '%d\n', number);
31         even=even+1;
32     else
33         fprintf(f2, '%d\n', number);
34         odd=odd+1;
35     end
36     i=i+1;
37 end
38 fclose(f1);
39 fclose(f2);
40 fclose(f3);
41 //Write odd numbers in the ODD.txt file
42 f2=fopen('ODD.txt', 'rt');
43 printf("\nContents of ODD file\n");
44 i=1;
45 while (i<=odd)
46     [n, number]=mfscanf(f2, "%d")
47     printf("%4d", number);
48     i=i+1;
49 end
50 //Write even numbers in the EVEN.txt file
51 f3=fopen('EVEN.txt', 'rt');

```

```

-->exec('C:\Users\Devender Thakur\Desktop\scilab\scilab programs\Files\exml2_2.sce', -1)
Contents of DATA file
-->111
-->222
-->333
-->444
-->555
-->666
-->777
-->888
-->999
-->000
-->121
-->232
-->343
-->454
-->565
-->-1

Contents of ODD file
111 333 555 777 999 121 343 565
Contents of EVEN file
222 444 666 888 0 232 454
-->

```

Figure 12.2: Read and write odd and even numbers

```

52 printf("\nContents of EVEN file\n");
53 i=1;
54 while (i<=even)
55     [n,number]=mfscanf(f3,"%d")
56     printf("%4d",number);
57     i=i+1;
58 end
59 //close the files
60 mclose(f2);
61 mclose(f3);

```

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

```

1 //                                     Example12.3
2 //A program to open a file named INVENTORY and
   store in it the following
3 // data: Item name      Number      Price      Quantity
4 //           AAA-1       111        17.50      115
5 //           BBB-2       125        36.00      75
6 //           C-3        247        31.75     104
7 //Extend the program to read this data from the file
   INVENTORY and display
8 //inventory table with the value of each item.
9
10 disp("Input file name");
11 filename=scanf("%s"); //Read file name that is '
   INVENTORY'
12 fp=fopen(filename,'w'); //Open file in write mode,
   fp is file descriptor
13 disp("Input inventory data");
14 disp("Item name      Number      Price      Quantity");
15 for i=1:3
16     //read data from terminal
17     [n,item(i),number(i),price(i),quantity(i)]=mscanf(
        "%s %d %f %d");
18     //write data to the file
19     fprintf(fp, '%s\t%d\t%.2f\t%d\n',item(i),number(i)
        ),price(i),quantity(i));
20 end
21 fclose(fp); //close the file
22 fp=fopen(filename,'r'); //open file in read mode
23 disp("Item name      Number      Price      Quantity      Value"
   );
24 for i=1:3
25     //Read data from the file 'INVENTORY'
26     [n,item,number,price,quantity]=mfsanf(fp,"%s %d
        %f %d");
27     value=price*quantity; //Computes value
28     //Printing of the data
29     printf(' %s %7d %8.2f %8d %11.2f\n',item,
        number,price,quantity,value);

```

```

-->exec('C:\Users\Devender Thakur\Desktop\scilab\scilab programs\Files\exm12_3.sce', -1)

Input file name
-->INVENTORY

Input inventory data

Item name      Number      Price      Quantity
-->AAA-1 111 17.50 115
-->BBB-2 125 36.00 75
-->C-3 247 31.75 104

Item name      Number      Price      Quantity      Value
AAA-1          111      17.50        115      2012.50
BBB-2          125      36.00         75      2700.00
C-3            247      31.75        104      3302.00

-->

```

Figure 12.3: Read and write data to and from the file INVENTORY

```

30 end
31 mclose(fp);

```

Scilab code Exa 12.4 Error handling in file operations

```

1 //                               Example12.4
2 //Write a program to illustatre error handling in
  file operations.
3
4 warning('off');
5 fp1=mopen('TEST','w'); //Open file in write mode,
  fp1 is file descriptor
6 for i=10:10:100
7     //write data to the file
8     mfprintf(fp1,'%d\n',i);
9 end
10 mclose(fp1);

```

```

11 disp("Input file name");
12 filename='a';
13 while(filename~= ' ')
14     filename=scanf("%s");
15     //Error handling
16     try
17         fp2=fopen(filename, 'r');
18         if(fp2>0) ,
19             break;           //Terminates the loop if
                               file exist or opened
20     end
21
22     catch
23         //Messages to be displayed when error
                occurred
24         printf("Can not open file.\n");
25         printf("Type file name again.\n");
26     end
27 end
28 //Code below runs while there is no error
29 for i=1:20
30     number = fscanf(fp2, "%d"); //Read data from
                file 'TEST'
31     if meof(fp2) then           //Test for end of
                file
32         printf("Ran out of data");
33         break;
34     else
35         printf("%d\n", number); //prints the data
36     end
37 end
38 fclose(fp2);

```

```

-->exec('C:\Users\Devender Thakur\Desktop\scilab\scilab programs\Files\exm12_4.sce', -1)

Input file name
-->TETS
Can not open file.
Type file name again.
-->TEST
10
20
30
40
50
60
70
80
90
100
Ran out of data
-->

```

Figure 12.4: Error handling in file operations

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

```

1 // Example12.5
2 //Write a program that uses function ftell(mtell)
  and fseek(mseek).
3
4 warning('off');
5 //Open file 'RANDOM' in write mode,fp is file
  descriptor
6 fp=mopen('RANDOM','w');
7 c=read(%io(1),1,1,'(a)');
8 mfprintf(fp,'%s',c); //write data to the file
9 printf("Number of characters entered = %d\n",mtell(
  fp));
10 mclose(fp);
11
12 //Open file 'RANDOM' in read mode
13 fp=mopen('RANDOM','r');
14 n=0;
15 while(meof(fp)==0)
16 //n is the offset from origin in number of bytes.

```



```

-->exec('C:\Users\Devender Thakur\Desktop\scilab\scilab programs\Files\exm12_5.sce', -1)
-->ABCDEFGHIJKLMNPOQRSTUVWXYZ
Number of characters entered = 26
Position of A is 1
Position of F is 6
Position of K is 11
Position of P is 16
Position of U is 21
Position of Z is 26
Position of is 30
ZYXWVUTSRQPONMLKJIHGFEDCBA
-->

```

Figure 12.5: use of function ftell or mtell and fseek or mseek

```

17 //The new position is at the signed distance given
    by n bytes from the beginning
18     mseek(n,fp, 'set ');
19     //Print the chracter and its postion
20     printf("Position of %c is %d\n",ascii(mget(1, 'c'
        ,fp)),mtell(fp));
21     n=n+5;
22 end
23 n=0; //Initial offset
24 while(mtell(fp)>1)
25 //New position is at the signed distance given by n
    bytes from the end
26     mseek(n,fp, 'end ');
27     //Print the characters from the end
28     printf("%c", (ascii(mget(1, 'c', fp))));
29     n=n-1;
30 end
31 mclose(fp);

```

Scilab code Exa 12.6 Append additional items to the file INVENTORY

```

1 //                                     Example 12.6
2 //Write a program to append additional items to the
   file INVENTORY
3 //created in Example 12.3 and print the total
   contents of the file.
4 funcprot(0);
5 warning('off');
6 function[item] =append(product,fp)
7     printf("Item name:\n");
8     product.name=scanf("%s");
9     printf("Item number:\n");
10    product.number=scanf("%d");
11    printf("Item price\n");
12    product.price=scanf("%f");
13    printf("Quantity:\n");
14    product.quantity=scanf("%d");
15    //Write data in the file
16    fprintf(fp, '%s %d %.2f %d\n',product.name,
           product.number,product.price,product.
           quantity);
17    item=product;
18 endfunction
19 //Creating structure
20 item=struct('name','0','number','0','price','0',
           'quantity','0');
21 //Read file name that is 'INVENTORY'
22 disp("Type file name");
23 filename=scanf("%s");
24 //Open file in append mode,fp is file descriptor
25 fp=fopen(filename,'a+');
26 b=0;response=-1;
27 //Read data
28 while(response==1|b==0)
29     item=append(item,fp); //calling append()
           function
30     printf("Item %s appended.\n",item.name);
31     printf("Do you want to add another item\ (1 for
           YES/0 for NO)?");

```

```
32     response=scanf("%d");
33     b=1;
34 end
35 n=mtell(fp);      //position of last character
36 mclose(fp);
37
38 //Opening file in the read mode
39 fp=mopen(filename, 'r');
40 while (mtell(fp) < n-2)
41     //read data from terminal
42     [g,item.name,item.number,item.price,item.quantity
43      ]=mfscanf(fp,"%s %d %f %d");
44     //Print Data to screen
45     printf('%s %7d %8.2f %8d\n',item.name,item.number
46            ,item.price,item.quantity);
47 end
48 mclose(fp);
```

```

-->exec('C:\Users\Devender Thakur\Desktop\scilab\scilab programs\Files\exm12_6.sce', -1)

Type file name
-->INVENTORY
Item name:
-->XXX
Item number:.
-->444
Item price
-->40.50
Quantity:
-->34
Item XXX appended.
Do you want to add another item(1 for YES/0 for NO)?
-->1
Item name:
-->YYY
Item number:.
-->555
Item price
-->50.50
Quantity:
-->45
Item YYY appended.
Do you want to add another item(1 for YES/0 for NO)?
-->0
AAA-1    111    17.50    115
BBB-2    222    36.00     75
C-3     247    31.75    104
XXX     444    40.50     34
YYY     555    50.50     45

```

Figure 12.6: Append additional items to the file INVENTORY

Chapter 13

Dynamic Memory Allocation and linked Lists

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

```
1 //                               Case Study: Chapter:13 ,Page No
   //                               .:434
2 //                               1.Insertion in a sorted list
3
4 funcprot(0);
5 //Create the list
6 function [List]=create(list1)
7     global List;
8     // Create the current node
9     list1.number=input("Input a number(Type -999 to
   end); ");
10    if list1.number==-999 then
11        list1.next=NULL;
12        list1.add=NULL;
13    else
14        list1.add=list1.add+1;
15        list1.next=NULL;
16        List(i)=list1;
17        if(i==1) then
```

```

18
19         else
20             List(i-1).next=List(i).add
21         end
22         i=i+1;
23         create(list1); // Create the next node
24     end
25     return;
26 endfunction
27 function []=print1(list1)
28     if list1(i)(1).next~=NULL then
29         printf("%d——>",list1(i)(1).number); //Print
30             current item
31         i=i+1;
32         if list1(i)(1).next==NULL then
33             printf("%d",list1(i)(1).number);
34         end
35         print1(list1); //Move to next item
36     end
37     return;
38 endfunction
39 function [List]=insert(list1)
40     global List;
41     x=input("Input number to be inserted: "); //Read
42     the number
43     //find the location so that number could be
44     placed in sorted order
45     while (list1(i)(1).next~=NULL)
46         if(list1(i)(1).number>=x) then
47             break;
48         end
49         i=i+1;
50     end
51     key=i;
52     //Insetion at end
53     if(list1(i)(1).next==NULL & list1(i)(1).number <
54         x) then
55         list1(i+1)(1).number=x;

```

```

52         list1(i+1)(1).add=i+1;
53         list1(i+1)(1).next=NULL;
54         list1(i)(1).next=list1(i+1)(1).add;
55         List=list1;
56         return;
57     end
58     i=1;
59     while (list1(i)(1).next~=NULL)
60         i=i+1;
61     end
62     j=i+1;
63     //Key node found and insert new node or item
64     while(list1(i)(1).add~=key)
65         list1(i+1)(1).number=list1(i)(1).number;
66         i=i-1;
67     end
68     list1(i+1)(1).number=list1(i)(1).number
69     list1(i)(1).number=x;
70     list1(j)(1).add=j;
71     list1(j)(1).next=NULL;
72     list1(j-1)(1).next=list1(j)(1).add;
73     List=list1;
74 endfunction
75
76 global List;
77 NULL=0;i=1;
78 //Create the structure i.e. node
79 node=struct('number',0,'add',0,'next',0);
80 head=node;
81 //Calling the functions
82 printf("Input a sorted(ascending) list");
83 List=create(head);
84 printf("\nOriginal List: ");
85 print1(List);
86 List=insert(List);
87 printf("\nNew List: ");
88 print1(List);

```

```

-->exec('C:\Users\Devender Thakur\Desktop\scilab\scilab programs\Dynamic memory\Casestudy1.sci', -1)
Input a sorted(ascending) list
Input a number(Type -999 to end); 10
Input a number(Type -999 to end); 20
Input a number(Type -999 to end); 30
Input a number(Type -999 to end); 40
Input a number(Type -999 to end); -999

Original List: 10--->20--->30--->40
Input number to be inserted: 25

New List: 10--->20--->25--->30--->40
-->

```

Figure 13.1: Case study 1 Insertion in a sorted list

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

```

1 // Case Study: Chapter:13 ,Page No
  .:438
2 // 2.Building a Sorted List
3
4 funcprot(0);
5 //Create the list
6 function [List]=create(list1)
7     global List;
8     // Create the current node
9     list1.number=input("Input a number(Type -999 to
    end); ");
10    if list1.number==-999 then
11        list1.next=NULL;
12        list1.add=NULL;
13    else
14        list1.add=list1.add+1;
15        list1.next=NULL;
16        List(i)=list1;
17        if(i==1) then

```



```

18
19         else
20             List(i-1).next=List(i).add
21         end
22         i=i+1;
23         create(list1); // Create the next node
24     end
25     return;
26 endfunction
27 function []=print1(list1)
28     if list1(i)(1).next~=NULL then
29         printf("%d——>",list1(i)(1).number); //Print
30             current item
31         i=i+1;
32         if list1(i)(1).next==NULL then
33             printf("%d",list1(i)(1).number);
34         end
35         print1(list1); //Move to next item
36     end
37     return;
38 endfunction
39 //Sorting of the numbers in the list
40 function [List]=insert_sort(list1)
41     global List;
42     j=1;
43     while (list1(j)(1).next~=NULL)
44         i=1;
45         while (list1(i)(1).next~=NULL)
46             if(list1(i)(1).number >list1(i+1)(1).
47                 number) then
48                 temp=list1(i)(1).number;
49                 list1(i)(1).number=list1(i+1)(1).
50                     number;
51                 list1(i+1)(1).number=temp;
52             end
53             i=i+1;
54         end
55     end
56     j=j+1;

```

```

-->exec('C:\Users\Devender Thakur\Desktop\scilab\scilab programs\Dynamic memory\Casestudy2.sci', -1)
Input a number(Type -999 to end); 80
Input a number(Type -999 to end); 70
Input a number(Type -999 to end); 50
Input a number(Type -999 to end); 40
Input a number(Type -999 to end); 60
Input a number(Type -999 to end); -999

Original List: 80--->70--->50--->40--->60
After sorting: 40--->50--->60--->70--->80
-->

```

Figure 13.2: Case study 2 Building a Sorted List

```

53     end
54     List=list1;
55 endfunction
56
57 global List;
58 NULL=0;i=1;
59 //Create the structure i.e. node
60 node=struct('number',0,'add',0,'next',0);
61 head=node;
62 //Calling the functions
63 List=create(head);
64 printf("\nOriginal List: ");
65 print1(List);
66 List=insert_sort(List); //Sort the list
67 printf("\nAfter sorting: ");
68 print1(List);

```

Scilab code Exa 13.3 Create a linear linked list

```

1 //                               Example 13.3
2 //Write a program to create a linear linked list
  interactively

```

```

3 //and print the list and total number of items in
  the list.
4
5 funcprot(0);
6 NULL=0;i=1;
7 //Create the list
8 function [List]=create(list1)
9     global List;
10    //Create the current node in the list
11    list1.number=input("Input a number(Type -999 to
    end); ")
12    if list1.number==-999 then
13        list1.next=NULL;
14        list1.add=NULL;
15    else
16        //Create the next node in the list
17        list1.add=list1.add+1;
18        list1.next=NULL;
19        List(i)=list1;
20        if(i==1) then
21
22            else
23                List(i-1).next=List(i).add
24            end
25            i=i+1;
26            create(list1); //Call create() function
27        end
28    return;
29 endfunction
30 //Function to print the numbers of list
31 function []=print1(list1)
32     if list1(i)(1).next~=NULL then
33         printf("%d——>",list1(i)(1).number); //Print
            current item
34         i=i+1;
35         if list1(i)(1).next==NULL then
36             printf("%d",list1(i)(1).number);
37         end

```

```

38         print1(list1); //Move to next item
39     end
40     return;
41 endfunction
42 //Function to count the number of items in the list
43 function []=count(list1)
44     global c;
45     if list1(i)(1).next==NULL then
46         return;
47     else
48         i=i+1;
49         c=i;
50         count(list1);
51     end
52     return;
53 endfunction
54 //Create the structure i.e. node
55 node=struct('number',0,'add',0,'next',0);
56 head=node;
57 global List;
58 //Calling the functions
59 List=create(head);
60 print1(List);
61 global c;
62 c=1;
63 count(List);
64 //Print the total number of items
65 printf("\nNumber of items = %d",c);

```

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

```

1 //                                     Example 13.4
2 //Write a function to insert a given item before a

```

```

-->exec('C:\Users\Devender Thakur\Desktop\scilab\scilab programs\Dynamic memory\exml3_3.sci', -1)
Input a number(Type -999 to end); 60
Input a number(Type -999 to end); 20
Input a number(Type -999 to end); 10
Input a number(Type -999 to end); 40
Input a number(Type -999 to end); 30
Input a number(Type -999 to end); 50
Input a number(Type -999 to end); -999
60--->20--->10--->40--->30--->50
Number of items = 6
-->

```

Figure 13.3: Create a linear linked list

```

    specified node known as
3 //key node.
4
5 funcprot(0);
6 //Create the list
7 function [List]=create(list1)
8     global List;
9     // Create the current node
10    list1.number=input("Input a number(Type -999 to
    end); ");
11    if list1.number==-999 then
12        list1.next=NULL;
13        list1.add=NULL;
14    else
15        list1.add=list1.add+1;
16        list1.next=NULL;
17        List(i)=list1;
18        if(i==1) then
19
20        else
21            List(i-1).next=List(i).add
22        end
23        i=i+1;
24        create(list1);// Creates the next node
25    end
26    return;
27 endfunction

```

```

28 //Function to insert the item before the specified
    key node
29 function [List]=insert(list1)
30     x=input("Value of new item?");
31     printf("Value of key item?(Before which you want
        to insert?)");
32     key=scanf("%d");
33     while list1(i)(1).next~=NULL
34         i=i+1;
35     end
36     j=i+1;
37     //Find the key node and insert the new node
38     while(list1(i)(1).number~=key)
39         list1(i+1)(1).number=list1(i)(1).number;
40         i=i-1;
41         if(i==0) then
42             printf("Item not Found");
43             return;
44         end
45     end
46     list1(i+1)(1).number=list1(i)(1).number
47     list1(i)(1).number=x; //Inset the new node
        before the key node
48     list1(j)(1).add=j;
49     list1(j)(1).next=NULL;
50     list1(j-1)(1).next=list1(j)(1).add;
51     List=list1;
52 endfunction
53 //Function to print the numbers of list
54 function []=print1(list1)
55     if list1(i)(1).next~=NULL then
56         printf("%d——>",list1(i)(1).number); //Print
            current item
57         i=i+1;
58         if list1(i)(1).next==NULL then
59             printf("%d",list1(i)(1).number);
60         end
61         print1(list1); //Move to next item

```

```

-->exec('C:\Users\Devender Thakur\Desktop\scilab\scilab programs\Dynamic memory\exml3_4.sci', -1)
Input a number(Type -999 to end); 60
Input a number(Type -999 to end); 20
Input a number(Type -999 to end); 10
Input a number(Type -999 to end); 40
Input a number(Type -999 to end); 50
Input a number(Type -999 to end); -999

Original List: 60--->20--->10--->40--->50
Value of new item?30
Value of key item?(Before which you want to insert?)
-->10

New List: 60--->20--->30--->10--->40--->50
-->

```

Figure 13.4: Insert the item before the specified key node

```

62     end
63     return;
64 endfunction
65 global List;
66 NULL=0;i=1;
67 //Create the structure i.e. node
68 node=struct('number',0,'add',0,'next',0);
69 head=node;
70 //Calling the functions
71 List=create(head);
72 printf("\nOriginal List: ");
73 print1(List);
74 List=insert(List);
75 printf("\nNew List: ");
76 print1(List);

```

Scilab code Exa 13.5 Delete a specified node in the list

```

1 //                               Example 13.5
2 //Write a program/function to delete a specified

```

```

node.
3
4 funcprot(0);
5 //Create the list
6 function [List]=create(list1)
7     global List;
8     // Create the current node
9     list1.number=input("Input a number(Type -999 to
    end); ")//scanf("%d");
10    if list1.number==-999 then
11        list1.next=NULL;
12        list1.add=NULL;
13    else
14        list1.add=list1.add+1;
15        list1.next=NULL;
16        List(i)=list1;
17        if(i==1) then
18
19            else
20                List(i-1).next=List(i).add
21            end
22            i=i+1;
23            create(list1);// Create the next node
24        end
25        return;
26    endfunction
27 //Function to print the numbers of list
28 function []=print1(list1)
29     if list1(i)(1).next~=NULL then
30         printf("%d--->",list1(i)(1).number);//Print
    current item
31         i=i+1;
32         if list1(i)(1).next==NULL then
33             printf("%d",list1(i)(1).number);
34         end
35         print1(list1);//Move to next item
36     end
37     return;

```



```

38 endfunction
39 //Function to delete the specified node
40 function [List]=delet(list1)
41     key=input(" Value of item number to be deleted?")
42     ;//Read value of key
43     //Find and delete the key node
44     while(list1(i)(1).number~=key) then
45         if list1(i)(1).next==NULL then
46             printf("Item not found in the list");
47             return;
48         end
49         i=i+1;
50     end
51     while(list1(i).next~=NULL)
52         list1(i)(1).number=list1(i+1)(1).number;
53         i=i+1;
54     end
55     list1(i-1)(1).next=NULL;
56     List=list1;
57 endfunction
58 global List;
59 NULL=0;i=1;
60 //Create the structure i.e. node
61 node=struct('number',0,'add',0,'next',0);
62 head=node;
63 //Calling the functions
64 List=create(head);
65 printf("\nOriginal List: ");
66 print1(List);
67 List=delet(List);
68 printf("\nAfter deletion List is: ");
69 print1(List)

```

```
-->exec('C:\Users\Devender Thakur\Desktop\scilab\scilab programs\Dynamic memory\exml3_5.sci', -1)
Input a number(Type -999 to end); 10
Input a number(Type -999 to end); 20
Input a number(Type -999 to end); 33
Input a number(Type -999 to end); 12
Input a number(Type -999 to end); 11
Input a number(Type -999 to end); -999

Original List: 10--->20--->33--->12--->11
Value of item number to be deleted?33

After deletion List is: 10--->20--->12--->11
-->
```

Figure 13.5: Delete a specified node in the list