Scilab Textbook Companion for Cryptography And Network Security by Behrouz A. Forouzan[1]

Created by
Subbulakshmi T
Cyber Security
Computer Engineering
VIT Chennai
College Teacher
None
Cross-Checked by
None

July 31, 2019

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

Book Description

Title: Cryptography And Network Security

Author: Behrouz A. Forouzan

Publisher: Tata Mcgraw-Hill, New Delhi India

Edition: 1

Year: 2007

ISBN: 978-0-07-066046-5

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

Exa Example (Solved example)

Eqn Equation (Particular equation of the above book)

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

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

Contents

List of Scilab Codes

Chapter 2

Mathematics of Cryptography

Scilab code Exa 2.1 Binary operations on two integers

```
1 // Chapter No : 2  Exercise Number : 2.1 of the Book
    Name : Cryptography and Network Security by
    Behrouz Forouzan, Special Indian Edition, 2007
2 // This file must be used under the terms of the
    CeCILL.
3 // This source file is licensed as described in the
    file COPYING, which
4 // you should have received as part of this
    distribution.  The terms
5 // are also available at
6 // http://www.cecill.info/licences/Licence_CeCILL_V2
    -en.txt
7 //This Source file is Written by Student Shreya
    Rajiv Somkuwar(15BCE1225), Student J Robin Raj(15
    BCE1325) Guided by Dr. T. Subbulakshmi, Professor
8 //School of Computing Science and Engineering, VIT
    University Chennai
9 //The Operating System used for writing the code
    found in this file is Windows 8
10 //SCILAB version 5.5.2
11 //
```

```
|-------------------------------------------------------------------------
12 //|This worked out example found in Page No: 16 of
     the book will do the following Binary operations
                                    |
13 //|1. Addition
14 //|2. Substraction
15 //|3. Multiplication
16 clear;
17 clc;
18 a=5;
19 b=9;
20 printf("----------Binary Operations--------\n");
21 printf("\nADD\n")
22 printf("%d + %d = %d\n",a,b,a + b);
23 printf("(-%d) + %d = %d\n",a,b,(-a) + b );
24 printf("%d + (-%d) = %d\n",a,b,a + (-b));
25 printf("(-%d) + (-%d) = %d\n",a,b,(-a) + (-b));
26 printf("\nSUBTRACT\n")
27 printf("%d - %d = %d\n",a,b,a - b);
28 printf("(-%d) - %d = %d\n",a,b,(-a) - b );
29 printf("%d - (-%d) = %d\n",a,b,a - (-b));
30 printf("(-%d) - (-%d) = %d\n",a,b,(-a) - (-b));
31 printf("\nMULTIPLY\n")
32 printf("%d x %d = %d\n",a,b,a * b);
33 printf("(-%d) x %d = %d\n",a,b,(-a) * b );
34 printf("%d x (-%d) = %d\n",a,b,a * (-b));
35 printf("(-%d) x (-%d) = %d\n",a,b,(-a) * (-b));
```

Scilab code Exa 2.2 Division Algorithm

```
1 // Chapter No : 2  Exercise Number : 2.2 of the Book
     Name : Cryptography and Network Security by
    Behrouz Forouzan, Special Indian Edition, 2007
2 // This file must be used under the terms of the
```

```
    CeCILL.
3 // This source file is licensed as described in the
    file COPYING, which
4 // you should have received as part of this
    distribution.  The terms
5 // are also available at
6 // http://www.cecill.info/licences/Licence_CeCILL_V2
    -en.txt
7 //This Source file is Written by Student Shreya
    Rajiv Somkuwar(15BCE1225), Student J Robin Raj(15
    BCE1325) Guided by Dr. T. Subbulakshmi, Professor
8 //School of Computing Science and Engineering, VIT
    University Chennai
9 //The Operating System used for writing the code
    found in this file is Windows 8
10 //SCILAB version 5.5.2
11 //
    |-------------------------------------------------------------------

12 //|This worked out example found in Page No: 17 of
    the book will do the following operations
                                  |
13 //|1. Division Algorithm for a = 255 and n=11
14 //|2. Print the result in the proper order with the
    result for q and r in the scilab command line
15 clc;
16 clear;
17 a=255;
18 n=11;
19 q=int(a/n);
20 r=a-(q*n);
21 printf("       %d\n",q);
22 printf("    ----------\n");
23 printf("%d  |  %d\n",n,a);
24 printf("     |  %d\n",n*q);
25 printf("      ----\n");
26 printf("         %d",r);
```

Scilab code Exa 2.3 Convert negative to postive

```
1 // Chapter No : 2  Exercise Number : 2.3 of the Book
     Name : Cryptography and Network Security by
    Behrouz Forouzan, Special Indian Edition, 2007
2 // This file must be used under the terms of the
    CeCILL.
3 // This source file is licensed as described in the
    file COPYING, which
4 // you should have received as part of this
    distribution.  The terms
5 // are also available at
6 // http://www.cecill.info/licences/Licence_CeCILL_V2
    -en.txt
7 //This Source file is Written by Student Shreya
    Rajiv Somkuwar(15BCE1225), Student J Robin Raj(15
    BCE1325) Guided by Dr. T. Subbulakshmi, Professor
8 //School of Computing Science and Engineering, VIT
    University Chennai
9 //The Operating System used for writing the code
    found in this file is Windows 8
10 //SCILAB version 5.5.2
11 //
    |-----------------------------------------------------------------

12 //||This worked out example found in Page No: 17 of
    the book will do the following operations
                                |
13 //||1. Find the remainder when the dividend is
    negative
14 //||2. Take the dividend as -255 and divisor as 11
    and find the remainder
15 //||3. Print the result in the scilab command line
16 clc;
```

```
17 clear;
18 a=-255;
19 n=11;
20 q=int(a/n);
21 r=a-n*q;
22 printf("Before:\n%d = ( %d x %d ) + %d\n",a,q,n,r);
23 if a<0 then:
24     q=q-1;
25     r=r+n;
26
27 printf("After:\n%d = ( %d x %d ) + %d\n",a,q,n,r);
```

Scilab code Exa 2.4 integer divison

```
1 // Chapter No : 2  Exercise Number : 2.4 of the Book
    Name : Cryptography and Network Security by
   Behrouz Forouzan, Special Indian Edition, 2007
2 // This file must be used under the terms of the
   CeCILL.
3 // This source file is licensed as described in the
   file COPYING, which
4 // you should have received as part of this
   distribution.  The terms
5 // are also available at
6 // http://www.cecill.info/licences/Licence_CeCILL_V2
   -en.txt
7 //This Source file is Written by Student Shreya
   Rajiv Somkuwar(15BCE1225), Student J Robin Raj(15
   BCE1325) Guided by Dr. T. Subbulakshmi, Professor
8 //School of Computing Science and Engineering, VIT
   University Chennai
9 //The Operating System used for writing the code
   found in this file is Windows 8
10 //SCILAB version 5.5.2
11 //
```

```
12 //|This worked out example found in Page No: 18 of
     the book will do the following operations
                                  |
13 //|1. Find the remainder for 32 / 4 and show that 4
     | 32 with r=0
14 //|2. Find the remainder for 42 / 8 and show that 8
     + 42 with r=2
15 //|3. Print the result in the scilab command line
16 clc;
17 clear;
18 a1=32;
19 n1=4;
20 n2=8;
21 a2=42;
22 q1=int(a1/n1);
23 r1=a1-q1*n1;
24 q2=int(a2/n2);
25 r2=a2-q2*n2;
26 printf("%d = ( %d x %d ) + %d\n",a1,q1,n1,r1);
27 printf("r=%d => , %d | %d\n",r1,n1,a1);
28 printf("%d = ( %d x %d ) + %d\n",a2,q2,n2,r2);
29 printf("r=%d => , %d + %d",r2,n2,a2);
```

Scilab code Exa 2.5 Division properties

```
1 // Chapter No : 2  Exercise Number : 2.5 of the Book
     Name : Cryptography and Network Security by
    Behrouz Forouzan, Special Indian Edition, 2007
2 // This file must be used under the terms of the
    CeCILL.
3 // This source file is licensed as described in the
    file COPYING, which
4 // you should have received as part of this
```

```
    distribution.  The terms
5 // are also available at
6 // http://www.cecill.info/licences/Licence_CeCILL_V2
    -en.txt
7 //This Source file is Written by Student Shreya
    Rajiv Somkuwar(15BCE1225), Student J Robin Raj(15
    BCE1325) Guided by Dr. T. Subbulakshmi, Professor
8 //School of Computing Science and Engineering, VIT
    University Chennai
9 //The Operating System used for writing the code
    found in this file is Windows 8
10 //SCILAB version 5.5.2
11 //
   |-------------------------------------------------------------------

12 //|This worked out example found in Page No: 18 of
    the book will do the following operations
                                  |
13 //|1. Solve and prove for 13 | 78, 7 | 98, -6 | 24,
    4 | 44, 11 | (-33)
14 //|2. Solve and prove for 13 + 27, 7 + 50, -6 + 23,
    4 + 41, 11 + (-32)
15 //|3. Print the result in the scilab command line
16 clc;
17 clear;
18 a=78
19 n=13
20 printf(" %d | %d , since %d = ( %d x %d ) + %d\n",a,
    n,a,int(a/n),n,a-int(a/n)*n);
21 a=98;
22 n=7;
23 printf(" %d | %d , since %d = ( %d x %d ) + %d\n",a,
    n,a,int(a/n),n,a-int(a/n)*n);
24 a=24;
25 n=-6;
26 printf(" %d | %d , since %d = ( %d x %d ) + %d\n",a,
    n,a,int(a/n),n,a-int(a/n)*n);
27 a=44;
```

```
28 n=4;
29 printf(" %d | %d , since %d = ( %d x %d ) + %d\n",a,
      n,a,int(a/n),n,a-int(a/n)*n);
30 a=-33
31 n=11;
32 printf(" %d | %d , since %d = ( %d x %d ) + %d\n",a,
      n,a,int(a/n),n,a-int(a/n)*n);
33 a=27;
34 n=13;
35 printf(" %d + %d , since %d = ( %d x %d ) + %d\n",a,
      n,a,int(a/n),n,a-int(a/n)*n);
36 a=50;
37 n=7;
38 printf(" %d + %d , since %d = ( %d x %d ) + %d\n",a,
      n,a,int(a/n),n,a-int(a/n)*n);
39 a=23;
40 n=-6;
41 printf(" %d + %d , since %d = ( %d x %d ) + %d\n",a,
      n,a,abs(int(a/n)),n,a-abs(int(a/n))*n);
42 a=41;
43 n=4;
44 printf(" %d + %d , since %d = ( %d x %d ) + %d\n",a,
      n,a,int(a/n),n,a-int(a/n)*n);
```

Scilab code Exa 2.6 Positive integer division

```
1 // Chapter No : 2  Exercise Number : 2.6 of the Book
     Name : Cryptography and Network Security by
     Behrouz Forouzan, Special Indian Edition, 2007
2 // This file must be used under the terms of the
     CeCILL.
3 // This source file is licensed as described in the
     file COPYING, which
4 // you should have received as part of this
     distribution.  The terms
```

```
5  // are also available at
6  // http://www.cecill.info/licences/Licence_CeCILL_V2
     -en.txt
7  //This Source file is Written by Student Shreya
     Rajiv Somkuwar(15BCE1225), Student J Robin Raj(15
     BCE1325) Guided by Dr. T. Subbulakshmi, Professor
8  //School of Computing Science and Engineering, VIT
     University Chennai
9  //The Operating System used for writing the code
     found in this file is Windows 8
10 //SCILAB version 5.5.2
11 //
     |-------------------------------------------------------------------------

12 //|This worked out example found in Page No: 18 of
     the book will do the following operations
                                  |
13 //|1. Prove 3 | 45 when given 3 | 15 and 15 | 45
14 //|2. Prove 3 | 66 when given 3 | 15 and 3 | 9
15 //|3. Print the result in the scilab command line
16 clc;
17 clear;
18 a=3;
19 b=15;
20 c=45;
21 printf("Since %d | %d  [ %d = ( %d x %d ) + %d ]",b,
     a,b,int(b/a),a,b-int(b/a)*a);
22 printf("\t    and\t")
23 printf(" %d | %d  [ %d = ( %d x %d ) + %d ]\n",c,b,c
     ,int(c/b),b,c-int(c/b)*b);
24 printf("Therefore using the property, if a|b and b|c
     => a|c :\n %d | %d  [ %d = ( %d x %d ) + %d ]\n"
     ,c,a,c,int(c/a),a,c-int(c/a)*a);
25
26 a=3;
27 b=15;
28 c=9;
29
```

```
30 printf("\n\nSince %d | %d  [ %d = ( %d x %d ) + %d ]
   ",b,a,b,int(b/a),a,b-int(b/a)*a);
31 printf("\t    and\t")
32 printf(" %d | %d  [ %d = ( %d x %d ) + %d ]\n",c,a,c
   ,int(c/a),a,c-int(c/a)*a);
33 printf("Therefore using the property, if a|b and a|c
    => a|( b x n + c x m) :\n ");
34 printf("%d | ( %d x %d + %d x %d ) => %d | %d",a,b
   ,2,c,4,a,b*2+c*4);
```

Scilab code Exa 2.7 Find the greatest common divisor

```
1 // Chapter No : 2  Exercise Number : 2.7 of the Book
    Name : Cryptography and Network Security by
   Behrouz Forouzan, Special Indian Edition, 2007
2 // This file must be used under the terms of the
   CeCILL.
3 // This source file is licensed as described in the
   file COPYING, which
4 // you should have received as part of this
   distribution.  The terms
5 // are also available at
6 // http://www.cecill.info/licences/Licence_CeCILL_V2
   -en.txt
7 //This Source file is Written by Student Shreya
   Rajiv Somkuwar(15BCE1225), Student J Robin Raj(15
   BCE1325) Guided by Dr. T. Subbulakshmi, Professor
8 //School of Computing Science and Engineering, VIT
   University Chennai
9 //The Operating System used for writing the code
   found in this file is Windows 8
10 //SCILAB version 5.5.2
11 //
   |-----------------------------------------------------------------
```

```
12 //|This worked out example found in Page No: 20 of
     the book will do the following operations
                                  |
13 //|1. Finding the grestest common divisor of 2740
     and 1760
14 //|2. Print the result in the scilab command line
     for each iteration
15 clc;
16 clear;
17 printf("  q     r1     r2     r \n")
18 a=2740;
19 b=1760;
20 q=int(a/b);
21 r=a-b*q;
22 printf("  %d    %d   %d    %d \n",q,a,b,r);
23 while r~=0,
24     a=b;
25     b=r;
26     q=int(a/b);
27     r=a-b*q;
28     printf("  %d    %d    %d    %d \n",q,a,b,r);
29 end
30 printf("        %d      %d      \n",b,r);
31 printf("\nTherefore, gcd(%d, %d)=%d",2740,1760,b);
```

Scilab code Exa 2.8 Find the greatest common divisor of 25 and 60

```
1 // Chapter No : 2  Exercise Number : 2.8 of the Book
     Name : Cryptography and Network Security by
    Behrouz Forouzan, Special Indian Edition, 2007
2 // This file must be used under the terms of the
    CeCILL.
3 // This source file is licensed as described in the
    file COPYING, which
4 // you should have received as part of this
```

```
    distribution.  The terms
5 // are also available at
6 // http://www.cecill.info/licences/Licence_CeCILL_V2
    -en.txt
7 //This Source file is Written by Student Shreya
    Rajiv Somkuwar(15BCE1225), Student J Robin Raj(15
    BCE1325) Guided by Dr. T. Subbulakshmi, Professor
8 //School of Computing Science and Engineering, VIT
    University Chennai
9 //The Operating System used for writing the code
    found in this file is Windows 8
10 //SCILAB version 5.5.2
11 //
   |-------------------------------------------------------------------------

12 //|This worked out example found in Page No: 20 of
    the book will do the following operations
                                  |
13 //|1. Finding the gcd of 25 and 60
14 //|2. Print the result in the scilab command line
    for each iteration
15 clc;
16 clear;
17 printf("  q     r1     r2     r \n")
18 a=25;
19 b=60;
20 q=int(a/b);
21 r=a-b*q;
22 printf("  %d    %d     %d     %d \n",q,a,b,r);
23 while r~=0,
24     a=b;
25     b=r;
26     q=int(a/b);
27     r=a-b*q;
28     printf("  %d    %d     %d     %d \n",q,a,b,r);
29 end
30 printf("        %d      %d     \n",b,r);
31 printf("\nTherefore, gcd(%d, %d)=%d",25,60,b);
```

Scilab code Exa 2.9 Find the GCD

```
1 // Chapter No : 2  Exercise Number : 2.9 of the Book
     Name : Cryptography and Network Security by
    Behrouz Forouzan, Special Indian Edition, 2007
2 // This file must be used under the terms of the
    CeCILL.
3 // This source file is licensed as described in the
    file COPYING, which
4 // you should have received as part of this
    distribution.  The terms
5 // are also available at
6 // http://www.cecill.info/licences/Licence_CeCILL_V2
    -en.txt
7 //This Source file is Written by Student Shreya
    Rajiv Somkuwar(15BCE1225), Student J Robin Raj(15
    BCE1325) Guided by Dr. T. Subbulakshmi, Professor
8 //School of Computing Science and Engineering, VIT
    University Chennai
9 //The Operating System used for writing the code
    found in this file is Windows 8
10 //SCILAB version 5.5.2
11 //
    |-------------------------------------------------------------------------

12 //|This worked out example found in Page No: 22 of
    the book will do the following operations
                                  |
13 //|1. Perform extended euclidean algorithm for a=161
     and b=28
14 //|2. Find the gcd, s and t from the above algorithm
15 //|3. Print the result in the scilab command line
16
17 clc;
```

```
clear;
printf("  q     r1     r2     r     s1     s2     s     t1     t2
      t \n")
a=161;
b=28;
q=int(a/b);
r=a-b*q;
s1=1;
s2=0;
s=s1-q*s2;
t1=0;
t2=1;
t=t1-q*t2;
printf("  %d     %d     %d     %d     %d     %d     %d     %d
      %d     %d \n",q,a,b,r,s1,s2,s,t1,t2,t);
while r~=0,
    a=b;
    b=r;
    q=int(a/b);
    r=a-b*q;
    s1=s2;
    s2=s;
    s=s1-q*s2;
    t1=t2;
    t2=t;
    t=t1-q*t2;
    printf("  %d     %d     %d     %d     %d     %d     %d
         %d     %d     %d\n",q,a,b,r,s1,s2,s,t1,t2,t);
end
printf("          %d     %d          %d     %d          %d
   %d     \n",b,r,s2,s,t2,t);
printf("\nTherefore:\ngcd(%d, %d)=%d\ns=%d, t=%d  =>
     %d x %d  + %d x %d = %d",25,60,b,s2,t2,s2,161,
   t2,28,s2*161+t2*28);
```

Scilab code Exa 2.10 find gcd of a and b and also the values of s and t

```
1 // Chapter No : 2  Exercise Number : 2.10 of the
    Book Name : Cryptography and Network Security by
    Behrouz Forouzan, Special Indian Edition, 2007
2 // This file must be used under the terms of the
    CeCILL.
3 // This source file is licensed as described in the
    file COPYING, which
4 // you should have received as part of this
    distribution.  The terms
5 // are also available at
6 // http://www.cecill.info/licences/Licence_CeCILL_V2
    -en.txt
7 //This Source file is Written by Student Shreya
    Rajiv Somkuwar(15BCE1225), Student J Robin Raj(15
    BCE1325) Guided by Dr. T. Subbulakshmi, Professor
8 //School of Computing Science and Engineering, VIT
    University Chennai
9 //The Operating System used for writing the code
    found in this file is Windows 8
10 //SCILAB version 5.5.2
11 //
    |-----------------------------------------------------------------

12 //|This worked out example found in Page No: 22 of
    the book will do the following operations
                                  |
13 //|1. Perform extended euclidean algorithm for a=17
    and b=0
14 //|2. Find the gcd, s and t from the above algorithm
15 //|3. Print the result in the scilab command line
16 clc;
17 clear;
18 a=17;
19 b=0;
20 s1=1;
21 s2=0;
```

```
22 t1=0;
23 t2=1;
24 printf("  q    r1    r2    r    s1    s2    s    t1    t2
     t \n");
25 printf("       %d    %d          %d     %d          %d
     %d     \n",a,b,s1,s2,t1,t2);
26 while b~=0,
27     a=b;
28     b=a-int(a/b);
29     q=int(a/b);
30     s1=s2;
31     s2=s1-q*s2;
32     t1=t2;
33     t2=t1-q*t2;
34     printf("  %d    %d    %d    %d    %d    %d  %d    %d
          %d    %d \n",q,a,b,a-b*q,s1,s2,s1-q*s2,t1,t2
        ,t1-q*t2);
35 end
36 printf("\nTherefore:\ngcd(%d, %d)=%d\ns=%d, t=%d  =>
     %d x %d  + %d x %d = %d",17,0,a,s1,t1,s1,17,t1
    ,0,s1*17+t1*0);
```

Scilab code Exa 2.11 Find the GCD

```
1 // Chapter No : 2  Exercise Number : 2.11 of the
    Book Name : Cryptography and Network Security by
    Behrouz Forouzan, Special Indian Edition, 2007
2 // This file must be used under the terms of the
    CeCILL.
3 // This source file is licensed as described in the
    file COPYING, which
4 // you should have received as part of this
    distribution.  The terms
5 // are also available at
6 // http://www.cecill.info/licences/Licence_CeCILL_V2
```

```
    -en.txt
7  //This Source file is Written by Student Shreya
     Rajiv Somkuwar(15BCE1225), Student J Robin Raj(15
     BCE1325) Guided by Dr. T. Subbulakshmi, Professor
8  //School of Computing Science and Engineering, VIT
     University Chennai
9  //The Operating System used for writing the code
     found in this file is Windows 8
10 //SCILAB version 5.5.2
11 //
     |-------------------------------------------------------------------

12 //||This worked out example found in Page No: 22 of
     the book will do the following operations
                                    |
13 //||1. Perform extended euclidean algorithm for a=0
     and b=45
14 //||2. Find the gcd(a,b), s and t from the above
     algorithm
15 //||3. Print the result in the scilab command line
16
17 clc;
18 clear;
19 a=0;
20 b=45;
21 s1=1;
22 s2=0;
23 t1=0;
24 t2=1;
25 printf("  q    r1    r2    r    s1    s2    s    t1    t2
     t \n");
26 printf("       %d    %d          %d     %d          %d
     %d     \n",a,b,s1,s2,t1,t2);
27 while b~=0,
28     r=a-int(a/b)
29     a=b;
30     b=r;
31     if b==0,
```

```
32          q=0;
33      else,
34          q=int(a/b)
35      end
36      s1=s2;
37      s2=s1-q*s2;
38      t1=t2;
39      t2=t1-q*t2;
40      printf(" %d  %d  %d  %d  %d  %d  %d  %d
            %d  %d \n",q,a,b,a-b*q,s1,s2,s1-q*s2,t1,t2
          ,t1-q*t2);
41  end
42  printf("\nTherefore:\ngcd(%d, %d)=%d\ns=%d, t=%d =>
      %d x %d + %d x %d = %d",0,45,a,s1,t1,s1,0,t1
    ,45,s1*0+t1*45);
```

Scilab code Exa 2.12 Find the particular and general solution of the equation

```
1  // Chapter No : 2  Exercise Number : 2.12 of the
     Book Name : Cryptography and Network Security by
     Behrouz Forouzan, Special Indian Edition, 2007
2  // This file must be used under the terms of the
     CeCILL.
3  // This source file is licensed as described in the
     file COPYING, which
4  // you should have received as part of this
     distribution.  The terms
5  // are also available at
6  // http://www.cecill.info/licences/Licence_CeCILL_V2
     -en.txt
7  //This Source file is Written by Student Shreya
     Rajiv Somkuwar(15BCE1225), Student J Robin Raj(15
     BCE1325) Guided by Dr. T. Subbulakshmi, Professor
8  //School of Computing Science and Engineering, VIT
     University Chennai
```

```
9  //The Operating System used for writing the code
     found in this file is Windows 8
10 //SCILAB version 5.5.2
11 //
     |-----------------------------------------------------------------------

12 //|This worked out example found in Page No: 23 of
     the book will do the following operations
                                  |
13 //|1. Find the paticular and general solution to the
      equation 21x + 14y = 35
14 //|2. Print the result in the scilab command line
15 clc;
16 clear;
17 a=[21 14 35]
18
19 common=gcd(a)
20 a=a/common
21
22 disp("x=k")
23 printf("y=(%d-%dx)/%d",a(3),a(1),a(2))
```

Scilab code Exa 2.14 Finding Results of various mod operations

```
1  // Chapter No : 2  Exercise Number : 2.14 of the
     Book Name : Cryptography and Network Security by
     Behrouz Forouzan, Special Indian Edition, 2007
2  // This file must be used under the terms of the
     CeCILL.
3  // This source file is licensed as described in the
     file COPYING, which
4  // you should have received as part of this
     distribution.  The terms
5  // are also available at
6  // http://www.cecill.info/licences/Licence_CeCILL_V2
```

```
   -en.txt
7  //This Source file is Written by Student Shreya
      Rajiv Somkuwar(15BCE1225), Student J Robin Raj(15
      BCE1325) Guided by Dr. T. Subbulakshmi, Professor
8  //School of Computing Science and Engineering, VIT
      University Chennai
9  //The Operating System used for writing the code
      found in this file is Windows 8
10 //SCILAB version 5.5.2
11 //
     |-------------------------------------------------------------------

12 //||This worked out example found in Page No: 24 of
      the book will do the following operations
                                    |
13 //||1. 27 mod 5
14 //||2. 36 mod 12
15 //||3. -18 mod 14
16 //||4. -7 mod 10
17 //||5. Print the result in the scilab command line
18 clc;
19 clear;
20 a=27
21 n=5
22 if(n~=0),
23     r=a-int(a/n)*n
24     q=int(a/n)
25     if(a<0),
26         q=q-1;
27         r=r+n;
28     end
29 end
30 printf(" %d mod %d = %d\n",a,n,r);
31
32 a=36
33 n=12
34 if(n~=0),
35     r=a-int(a/n)*n
```

```
36      q=int(a/n)
37      if(a<0),
38          q=q-1;
39          r=r+n;
40      end
41  end
42  printf(" %d mod %d = %d\n",a,n,r);
43
44  a=-18
45  n=14
46  if(n~=0),
47      r=a-int(a/n)*n
48      q=int(a/n)
49      if(a<0),
50          q=q-1;
51          r=r+n;
52      end
53  end
54  printf(" %d mod %d = %d\n",a,n,r);
55
56  a=-7
57  n=10
58  if(n~=0),
59      r=a-int(a/n)*n
60      q=int(a/n)
61      if(a<0),
62          q=q-1;
63          r=r+n;
64      end
65  end
66  printf(" %d mod %d = %d",a,n,r);
```

Scilab code Exa 2.16 Performing arithmetic operations in Zn

```
1  // Chapter No : 2  Exercise Number : 2.16 of the
```

```
1  // Book Name : Cryptography and Network Security by
       Behrouz Forouzan, Special Indian Edition, 2007
2  // This file must be used under the terms of the
       CeCILL.
3  // This source file is licensed as described in the
       file COPYING, which
4  // you should have received as part of this
       distribution.  The terms
5  // are also available at
6  // http://www.cecill.info/licences/Licence_CeCILL_V2
       -en.txt
7  //This Source file is Written by Student Shreya
       Rajiv Somkuwar(15BCE1225), Student J Robin Raj(15
       BCE1325) Guided by Dr. T. Subbulakshmi, Professor
8  //School of Computing Science and Engineering, VIT
       University Chennai
9  //The Operating System used for writing the code
       found in this file is Windows 8
10 //SCILAB version 5.5.2
11 //
       |-------------------------------------------------------------------------

12 //|This worked out example found in Page No: 27 of
       the book will do the following operations
                                    |
13 //|1. Add 7 to 14 in Z(15)
14 //|2. Subtract 11 from 7 in Z(13)
15 //|3. Multiply 11 by 7 in Z(20)
16 //|4. Print the result in the scilab command line
17
18 clc;
19 clear;
20 printf("a.\n")
21 a=7
22 b=14
23 c=a+b
24 n=15
25 if(n~=0),
```

```
26      r=c-int(c/n)*n
27      q=int(c/n)
28      if(c<0),
29          q=q-1;
30          r=r+n;
31      end
32  end
33  printf(" (%d + %d) mod %d ->  (%d) mod %d = %d\n",b,
       a,n,c,n,r);
34
35  printf("\n")
36
37  printf("b.\n")
38  a=11
39  b=7
40  c=b-a
41  n=13
42  if(n~=0),
43      r=c-int(c/n)*n
44      q=int(c/n)
45      if(c<0),
46          q=q-1;
47          r=r+n;
48      end
49  end
50  printf(" (%d - %d) mod %d ->  (%d) mod %d = %d\n",b,
       a,n,c,n,r);
51
52  printf("\n")
53
54  printf("c.\n")
55  a=11
56  b=7
57  c=a*b
58  n=20
59  if(n~=0),
60      r=c-int(c/n)*n
61      q=int(c/n)
```

```
62      if(c<0),
63          q=q-1;
64          r=r+n;
65      end
66  end
67  printf(" (%d x %d) mod %d ->  (%d) mod %d = %d\n",b,
    a,n,c,n,r);
```

Scilab code Exa 2.17 Performing arithmetic operations From Z or Zn

```
1  // Chapter No : 2  Exercise Number : 2.17 of the
     Book Name : Cryptography and Network Security by
     Behrouz Forouzan, Special Indian Edition, 2007
2  // This file must be used under the terms of the
     CeCILL.
3  // This source file is licensed as described in the
     file COPYING, which
4  // you should have received as part of this
     distribution.  The terms
5  // are also available at
6  // http://www.cecill.info/licences/Licence_CeCILL_V2
     -en.txt
7  //This Source file is Written by Student Shreya
     Rajiv Somkuwar(15BCE1225), Student J Robin Raj(15
     BCE1325) Guided by Dr. T. Subbulakshmi, Professor
8  //School of Computing Science and Engineering, VIT
     University Chennai
9  //The Operating System used for writing the code
     found in this file is Windows 8
10 //SCILAB version 5.5.2
11 //
     |-----------------------------------------------------------------

12 //|This worked out example found in Page No: 27 of
     the book will do the following operations
```

```
13 //|1. Add 17 to 27 in Z(14)
14 //|2. Subtract 34 from 12 in Z(13)
15 //|3. Multiply 123 by -10 in Z(19)
16 //|4. Print the result in the scilab command line
17 clc;
18 clear;
19
20 printf("a.\n")
21 a=27
22 b=17
23 c=a+b
24 n=14
25 if(n~=0),
26     r=c-int(c/n)*n
27     q=int(c/n)
28     if(c<0),
29         q=q-1;
30         r=r+n;
31     end
32 end
33 printf(" (%d + %d) mod %d ->  (%d) mod %d = %d\n",b,
      a,n,c,n,r);
34
35 printf("\n")
36
37 printf("b.\n")
38 a=43
39 b=12
40 c=b-a
41 n=13
42 if(n~=0),
43     r=c-int(c/n)*n
44     q=int(c/n)
45     if(c<0),
46         q=q-1;
47         r=r+n;
48     end
```

```
49 end
50 printf(" (%d - %d) mod %d ->  (%d) mod %d = %d\n",b,
      a,n,c,n,r);
51
52 printf("\n")
53
54 printf("c.\n")
55 a=-10
56 b=123
57 c=a*b
58 n=19
59 if(n~=0),
60     r=c-int(c/n)*n
61     q=int(c/n)
62     if(c<0),
63         q=q-1;
64         r=r+n;
65     end
66 end
67 printf(" (%d x %d) mod %d ->  (%d) mod %d = %d\n",b,
      a,n,c,n,r);
```

Scilab code Exa 2.18 Applications of mod operation

```
1 // Chapter No : 2  Exercise Number : 2.18 of the
     Book Name : Cryptography and Network Security by
     Behrouz Forouzan, Special Indian Edition, 2007
2 // This file must be used under the terms of the
     CeCILL.
3 // This source file is licensed as described in the
     file COPYING, which
4 // you should have received as part of this
     distribution.  The terms
5 // are also available at
6 // http://www.cecill.info/licences/Licence_CeCILL_V2
```

```
    -en.txt
7  //This Source file is Written by Student Shreya
       Rajiv Somkuwar(15BCE1225), Student J Robin Raj(15
       BCE1325) Guided by Dr. T. Subbulakshmi, Professor
8  //School of Computing Science and Engineering, VIT
       University Chennai
9  //The Operating System used for writing the code
       found in this file is Windows 8
10 //SCILAB version 5.5.2
11 //
     |-------------------------------------------------------------------

12 //||This worked out example found in Page No: 28 of
       the book will do the following operations
                                  |
13 //||1. Demonstration of the properties:
14 //////| i) (a+b)mod n = [(a mod n)+(b mod n)] mod n
15 //////| ii) (a-b)mod n = [(a mod n)-(b mod n)] mod n
16 //////| iii) (axb)mod n = [(a mod n)x(b mod n)] mod n
17 //||2. Print the result in the scilab command line
18 clc;
19 clear;
20
21 a=1723345;
22 b=2124945;
23 n=11;
24 amodn=a-int(a/n)*n;
25 bmodn=b-int(b/n)*n;
26 aplusbmodn=(amodn+bmodn)-int((amodn+bmodn)/n)*n;
27 printf("1.\n");
28 printf("(%d + %d) mod %d = (%d + %d) mod %d = %d",a,
       b,n,amodn,bmodn,n,aplusbmodn);
29
30 printf("\n");
31
32 a=1723345;
33 b=2124945;
34 n=11;
```

```
amodn=a-int(a/n)*n;
bmodn=b-int(b/n)*n;
asubbmodn=(amodn-bmodn);
if asubbmodn<0 then,
    asubbmodn=asubbmodn-int(asubbmodn/n)*n+n;
end
printf("2.\n");
printf("(%d - %d) mod %d = (%d - %d) mod %d = %d",a,
   b,n,amodn,bmodn,n,asubbmodn);

printf("\n");

a=1723345;
b=2124945;
n=11;
amodn=a-int(a/n)*n;
bmodn=b-int(b/n)*n;
aprodbmodn=(amodn*bmodn);
if aprodbmodn<0 then,
    aprodbmodn=aprodbmodn-int(aprodbmodn/n)*n+n;
else
    aprodbmodn=aprodbmodn-int(aprodbmodn/n)*n;
end
printf("3.\n");
printf("(%d x %d) mod %d = (%d x %d) mod %d = %d",a,
   b,n,amodn,bmodn,n,aprodbmodn);
```

Scilab code Exa 2.19 Finding remainder of Powers

```
// Chapter No : 2  Exercise Number : 2.19 of the
   Book Name : Cryptography and Network Security by
   Behrouz Forouzan, Special Indian Edition, 2007
// This file must be used under the terms of the
   CeCILL.
// This source file is licensed as described in the
```

```
    file COPYING, which
4 // you should have received as part of this
    distribution.  The terms
5 // are also available at
6 // http://www.cecill.info/licences/Licence_CeCILL_V2
    -en.txt
7 //This Source file is Written by Student Shreya
    Rajiv Somkuwar(15BCE1225), Student J Robin Raj(15
    BCE1325) Guided by Dr. T. Subbulakshmi, Professor
8 //School of Computing Science and Engineering, VIT
    University Chennai
9 //The Operating System used for writing the code
    found in this file is Windows 8
10 //SCILAB version 5.5.2
11 //
    |-------------------------------------------------------------------------

12 //|This worked out example found in Page No: 28 of
    the book will do the following operations
                                    |
13 //|1. Using the property 10^n mod x = (10 mod x)^n
    to find 10^n mod 3, 10^n mod 9 and 10^n mod 7
14 //|2. Print the result in the scilab command line
15
16 clc;
17 clear;
18 printf("\t(10^n) mod x = (10 mod x)^n\t\n");
19 n=2;
20 x=3;
21 Tmod3=1;
22 Tmod9=1;
23 Tmod7=3;
24 printf("x = %d\n",x);
25 printf("10 mod %d = %d ->  (10^n) mod %d = (10 mod
    %d)^n = %d^n\n",x,Tmod3,x,x,Tmod3);
26 x=9;
27 printf("x = %d\n",x);
28 printf("10 mod %d = %d ->  (10^n) mod %d = (10 mod
```

```
    %d)^n = %d^n\n",x,Tmod9,x,x,Tmod9);
29 x=7;
30 printf("x = %d\n",x);
31 printf("10 mod %d = %d ->  (10^n) mod %d = (10 mod
    %d)^n = %d^n\n",x,Tmod7,x,x,Tmod7);
```

Scilab code Exa 2.21 Find all additive inverse pairs in Z10

```
1  // Chapter No : 2  Exercise Number : 2.21 of the
     Book Name : Cryptography and Network Security by
     Behrouz Forouzan, Special Indian Edition, 2007
2  // This file must be used under the terms of the
     CeCILL.
3  // This source file is licensed as described in the
     file COPYING, which
4  // you should have received as part of this
     distribution.  The terms
5  // are also available at
6  // http://www.cecill.info/licences/Licence_CeCILL_V2
     -en.txt
7  //This Source file is Written by Student Shreya
     Rajiv Somkuwar(15BCE1225), Student J Robin Raj(15
     BCE1325) Guided by Dr. T. Subbulakshmi, Professor
8  //School of Computing Science and Engineering, VIT
     University Chennai
9  //The Operating System used for writing the code
     found in this file is Windows 8
10 //SCILAB version 5.5.2
11 //
     |-------------------------------------------------------------------

12 //|This worked out example found in Page No: 29 of
     the book will do the following operations
                                    |
13 //|1. Finding all additive inverse in Z(10)
```

```
14 //|2. Print the result in the scilab command line
15
16 clc;
17 clear;
18 n=10
19 printf("n=%d\n",n);
20 for i = 0:n/2
21     b=n-i
22     if(i==0)
23         b=0
24     end
25     printf("(%d, %d)\n",i,b);
26 end
```

Scilab code Exa 2.22 Find the Multiplicative Inverse of 8 in Z10

```
1 // Chapter No : 2  Exercise Number : 2.22 of the
    Book Name : Cryptography and Network Security by
    Behrouz Forouzan, Special Indian Edition, 2007
2 // This file must be used under the terms of the
    CeCILL.
3 // This source file is licensed as described in the
    file COPYING, which
4 // you should have received as part of this
    distribution.  The terms
5 // are also available at
6 // http://www.cecill.info/licences/Licence_CeCILL_V2
    -en.txt
7 //This Source file is Written by Student Shreya
    Rajiv Somkuwar(15BCE1225), Student J Robin Raj(15
    BCE1325) Guided by Dr. T. Subbulakshmi, Professor
8 //School of Computing Science and Engineering, VIT
    University Chennai
9 //The Operating System used for writing the code
    found in this file is Windows 8
```

```
10 //SCILAB version 5.5.2
11 //
   |-------------------------------------------------------------------------
12 //|This worked out example found in Page No: 29 of
   the book will do the following operations
                                  |
13 //|1. Finding the multiplicative inverse of 8 in Z
   (10)
14 //|2. Print the result in the scilab command line
15
16 clc;
17 clear;
18 a=10
19 b=8
20 while b~=0
21     r=a-int(a/b)*b
22     a=b
23     b=r
24 end
25 printf("gcd(%d, %d) = %d\n",10,8,a);
26 if(a~=1)
27     printf("Since gcd(%d, %d) = %d is not equal to
          1, multiplicative inverse does not exist."
          ,10,8,a);
```

Scilab code Exa 2.23 Find all multiplicative inverses in Z10

```
1 // Chapter No : 2  Exercise Number : 2.23 of the
   Book Name : Cryptography and Network Security by
   Behrouz Forouzan, Special Indian Edition, 2007
2 // This file must be used under the terms of the
   CeCILL.
3 // This source file is licensed as described in the
   file COPYING, which
```

```
4  // you should have received as part of this
       distribution.  The terms
5  // are also available at
6  // http://www.cecill.info/licences/Licence_CeCILL_V2
       -en.txt
7  //This Source file is Written by Student Shreya
       Rajiv Somkuwar(15BCE1225), Student J Robin Raj(15
       BCE1325) Guided by Dr. T. Subbulakshmi, Professor
8  //School of Computing Science and Engineering, VIT
       University Chennai
9  //The Operating System used for writing the code
       found in this file is Windows 8
10 //SCILAB version 5.5.2
11 //
     |-------------------------------------------------------------------

12 //|This worked out example found in Page No: 29 of
       the book will do the following operations
                                  |
13 //|1. Finding all multiplicative inverse pairs in Z
       (10)
14 //|3. Print the result in the scilab command line
15 clc;
16 clear;
17
18 n=10
19 for i = 1:n-1
20     b=i
21     a=n
22     while b~=0
23         r=a-int(a/b)*b
24         a=b
25         b=r
26     end
27     if(a==1),
28         printf("\ngcd(%d, %d)=%d -> Multiplicative
               inverse of %d exists.\n",n,i,a,i);
29         for j=1:n-1
```

```
30              t=i*j
31              m=t-int(t/n)*n
32              if m==1
33                  printf("Multiplicative inverse of %d
                        is %d -> ( %d x %d) mod %d = %d\
                       n",i,j,i,j,n,m);
34              end
35          end
36      end
37  end
```

Scilab code Exa 2.24 Find all multiplicative inverse pair in Z11

```
1  // Chapter No : 2  Exercise Number : 2.24 of the
     Book Name : Cryptography and Network Security by
     Behrouz Forouzan, Special Indian Edition, 2007
2  // This file must be used under the terms of the
     CeCILL.
3  // This source file is licensed as described in the
     file COPYING, which
4  // you should have received as part of this
     distribution.  The terms
5  // are also available at
6  // http://www.cecill.info/licences/Licence_CeCILL_V2
     -en.txt
7  //This Source file is Written by Student Shreya
     Rajiv Somkuwar(15BCE1225), Student J Robin Raj(15
     BCE1325) Guided by Dr. T. Subbulakshmi, Professor
8  //School of Computing Science and Engineering, VIT
     University Chennai
9  //The Operating System used for writing the code
     found in this file is Windows 8
10 //SCILAB version 5.5.2
11 //
     |----------------------------------------------------------------------
```

```
12 //|This worked out example found in Page No: 30 of
      the book will do the following operations
                                  |
13 //|1. Find pair of numbers < 11 such that (a,b): a
      is multiplicative inverse of b and vice versa
14 //|2. Print the result in the scilab command line
15 clc;
16 clear;
17
18 n=11
19 for i = 1:n-1
20     b=i
21     a=n
22     while b~=0
23         r=a-int(a/b)*b
24         a=b
25         b=r
26     end
27     if(a==1),
28         printf("\ngcd(%d, %d)=%d -> Multiplicative
              inverse of %d exists.\n",n,i,a,i);
29         for j=1:n-1
30             t=i*j
31             m=t-int(t/n)*n
32             if m==1
33                 printf("Multiplicative inverse of %d
                       is %d -> ( %d x %d) mod %d = %d\
                      n",i,j,i,j,n,m);
34             end
35         end
36     end
37 end
```

Scilab code Exa 2.25 Find all multiplicative inverse of 11 in Z26

```
1  // Chapter No : 2  Exercise Number : 2.25 of the
       Book Name : Cryptography and Network Security by
       Behrouz Forouzan, Special Indian Edition, 2007
2  // This file must be used under the terms of the
       CeCILL.
3  // This source file is licensed as described in the
       file COPYING, which
4  // you should have received as part of this
       distribution.  The terms
5  // are also available at
6  // http://www.cecill.info/licences/Licence_CeCILL_V2
       -en.txt
7  //This Source file is Written by Student Shreya
       Rajiv Somkuwar(15BCE1225), Student J Robin Raj(15
       BCE1325) Guided by Dr. T. Subbulakshmi, Professor
8  //School of Computing Science and Engineering, VIT
       University Chennai
9  //The Operating System used for writing the code
       found in this file is Windows 8
10 //SCILAB version 5.5.2
11 //
       |-------------------------------------------------------------------

12 //|This worked out example found in Page No: 31 of
       the book will do the following operations
                                    |
13 //|1. Perform extended euclidean algorithm for a=26
       and b=11
14 //|2. Find the multiplicative inverse
15 //|3. Print the result in the scilab command line
16 clc;
17 clear;
18
19 printf("  q    r1    r2    r    s1    s2    s    t1    t2
         t \n")
20 a=26;
21 b=11;
22 q=int(a/b);
```

```
23 r=a-b*q;
24 s1=1;
25 s2=0;
26 s=s1-q*s2;
27 t1=0;
28 t2=1;
29 t=t1-q*t2;
30 printf("  %d    %d     %d     %d    %d    %d    %d     %d
       %d    %d \n",q,a,b,r,s1,s2,s,t1,t2,t);
31 while r~=0,
32     a=b;
33     b=r;
34     q=int(a/b);
35     r=a-b*q;
36     s1=s2;
37     s2=s;
38     s=s1-q*s2;
39     t1=t2;
40     t2=t;
41     t=t1-q*t2;
42     printf("  %d     %d     %d     %d     %d    %d    %d
          %d    %d    %d\n",q,a,b,r,s1,s2,s,t1,t2,t);
43 end
44 printf("        %d     %d          %d    %d          %d
   %d     \n",b,r,s2,s,t2,t);
45
46 printf("\ngcd(%d, %d)=%d, therefore multiplicative
   inverse exists\n Multiplicative inverse of %d =>
   (%d) mod %d = %d",26,11,b,11,t2,26,t2-int(t2/26)
   *26+26);
```

Scilab code Exa 2.26 Find the Multiplicative Inverse of 23 in Z100

```
1 // Chapter No : 2  Exercise Number : 2.26 of the
   Book Name : Cryptography and Network Security by
```

```
1  Behrouz Forouzan, Special Indian Edition, 2007
2  // This file must be used under the terms of the
       CeCILL.
3  // This source file is licensed as described in the
       file COPYING, which
4  // you should have received as part of this
       distribution.  The terms
5  // are also available at
6  // http://www.cecill.info/licences/Licence_CeCILL_V2
       -en.txt
7  //This Source file is Written by Student Shreya
       Rajiv Somkuwar(15BCE1225), Student J Robin Raj(15
       BCE1325) Guided by Dr. T. Subbulakshmi, Professor
8  //School of Computing Science and Engineering, VIT
       University Chennai
9  //The Operating System used for writing the code
       found in this file is Windows 8
10 //SCILAB version 5.5.2
11 //
     |-------------------------------------------------------------------------

12 //|This worked out example found in Page No: 31 of
       the book will do the following operations
                                  |
13 //|1. Perform extended euclidean algorithm for a=100
        and b=23
14 //|2. Find the multiplicative inverse
15 //|3. Print the result in the scilab command line
16
17 clc;
18 clear;
19 printf(" q    r1    r2    r    s1    s2    s    t1    t2
         t \n")
20 a=100;
21 b=23;
22 q=int(a/b);
23 r=a-b*q;
24 s1=1;
```

```
25 s2=0;
26 s=s1-q*s2;
27 t1=0;
28 t2=1;
29 t=t1-q*t2;
30 printf(" %d %d %d %d %d %d %d %d
       %d %d \n",q,a,b,r,s1,s2,s,t1,t2,t);
31 while r~=0,
32     a=b;
33     b=r;
34     q=int(a/b);
35     r=a-b*q;
36     s1=s2;
37     s2=s;
38     s=s1-q*s2;
39     t1=t2;
40     t2=t;
41     t=t1-q*t2;
42     printf(" %d %d %d %d %d %d %d
           %d %d %d\n",q,a,b,r,s1,s2,s,t1,t2,t);
43 end
44 printf(" %d %d %d %d %d
   %d \n",b,r,s2,s,t2,t);
45
46 printf("\ngcd(%d, %d)=%d, therefore multiplicative
   inverse exists\n Multiplicative inverse of %d =>
   (%d) mod %d = %d",100,23,b,23,t2,100,t2-int(t2
   /100)*100+100);
```

Scilab code Exa 2.27 Find the inverse of 12 in Z26

```
1 // Chapter No : 2 Exercise Number : 2.27 of the
   Book Name : Cryptography and Network Security by
   Behrouz Forouzan, Special Indian Edition, 2007
2 // This file must be used under the terms of the
```

```
    CeCILL.
3 // This source file is licensed as described in the
    file COPYING, which
4 // you should have received as part of this
    distribution.  The terms
5 // are also available at
6 // http://www.cecill.info/licences/Licence_CeCILL_V2
    -en.txt
7 //This Source file is Written by Student Shreya
    Rajiv Somkuwar(15BCE1225), Student J Robin Raj(15
    BCE1325) Guided by Dr. T. Subbulakshmi, Professor
8 //School of Computing Science and Engineering, VIT
    University Chennai
9 //The Operating System used for writing the code
    found in this file is Windows 8
10 //SCILAB version 5.5.2
11 //
    |-------------------------------------------------------------------

12 //|This worked out example found in Page No: 31 of
    the book will do the following operations
                                  |
13 //|1. Perform extended euclidean algorithm for a=26
    and b=12
14 //|2. Find the multiplicative inverse
15 //|3. Print the result in the scilab command line
16
17 clc;
18 clear;
19 printf(" q    r1    r2    r    s1    s2    s    t1    t2
       t \n")
20 a=26;
21 b=12;
22 q=int(a/b);
23 r=a-b*q;
24 s1=1;
25 s2=0;
26 s=s1-q*s2;
```

```
27 t1=0;
28 t2=1;
29 t=t1-q*t2;
30 printf("  %d    %d     %d     %d    %d    %d    %d     %d
       %d    %d \n",q,a,b,r,s1,s2,s,t1,t2,t);
31 while r~=0,
32     a=b;
33     b=r;
34     q=int(a/b);
35     r=a-b*q;
36     s1=s2;
37     s2=s;
38     s=s1-q*s2;
39     t1=t2;
40     t2=t;
41     t=t1-q*t2;
42     printf("  %d     %d     %d     %d     %d    %d    %d
           %d    %d    %d\n",q,a,b,r,s1,s2,s,t1,t2,t);
43 end
44 printf("         %d     %d          %d    %d         %d
   %d     \n",b,r,s2,s,t2,t);
45
46 printf("\ngcd(%d, %d)=%d, therefore multiplicative
   inverse does not exists\n ",26,12,b);
```

Scilab code Exa 2.28 shows an Example of addition and substraction

```
1 // Chapter No : 2  Exercise Number : 2.28 of the
   Book Name : Cryptography and Network Security by
   Behrouz Forouzan, Special Indian Edition, 2007
2 // This file must be used under the terms of the
   CeCILL.
3 // This source file is licensed as described in the
   file COPYING, which
4 // you should have received as part of this
```

```
    distribution.  The terms
5 // are also available at
6 // http://www.cecill.info/licences/Licence_CeCILL_V2
    -en.txt
7 //This Source file is Written by Student Shreya
    Rajiv Somkuwar(15BCE1225), Student J Robin Raj(15
    BCE1325) Guided by Dr. T. Subbulakshmi, Professor
8 //School of Computing Science and Engineering, VIT
    University Chennai
9 //The Operating System used for writing the code
    found in this file is Windows 8
10 //SCILAB version 5.5.2
11 //
   |-------------------------------------------------------------------------

12 //|This worked out example found in Page No: 34 of
    the book will do the following operations
                                  |
13 //|1. Addition of two matrices
14 //|2. Subtraction of two matrices
15 //|3. Print the result in the scilab command line
16 clc;
17 clear;
18 A = [5,2,1;
19      3,2,10]
20 B = [7,2,3;
21      8,10,20]
22 C = A + B
23 disp(C)
```

Scilab code Exa 2.29 Product of row matrix by column matrix

```
1 // Chapter No : 2  Exercise Number : 2.29 of the
    Book Name : Cryptography and Network Security by
    Behrouz Forouzan, Special Indian Edition, 2007
```

```
2 // This file must be used under the terms of the
    CeCILL.
3 // This source file is licensed as described in the
    file COPYING, which
4 // you should have received as part of this
    distribution.  The terms
5 // are also available at
6 // http://www.cecill.info/licences/Licence_CeCILL_V2
    -en.txt
7 //This Source file is Written by Student Shreya
    Rajiv Somkuwar(15BCE1225), Student J Robin Raj(15
    BCE1325) Guided by Dr. T. Subbulakshmi, Professor
8 //School of Computing Science and Engineering, VIT
    University Chennai
9 //The Operating System used for writing the code
    found in this file is Windows 8
10 //SCILAB version 5.5.2
11 //
    |-------------------------------------------------------------

12 //|This worked out example found in Page No: 34 of
    the book will do the following operations
                                |
13 //|1. Multiplication of a row matrix with a column
    matrix
14 //|2. Print the result in the scilab command line
15 clc;
16 clear;
17
18 A = [5,2,1]
19 B = [7;
20      8;
21      2]
22
23 C = A*B
24 disp(C)
```

Scilab code Exa 2.30 Shows the product of 2x3 matrix by a 3x4 matrix and result is

```
1 // Chapter No : 2  Exercise Number : 2.30 of the
    Book Name : Cryptography and Network Security by
    Behrouz Forouzan, Special Indian Edition, 2007
2 // This file must be used under the terms of the
    CeCILL.
3 // This source file is licensed as described in the
    file COPYING, which
4 // you should have received as part of this
    distribution.  The terms
5 // are also available at
6 // http://www.cecill.info/licences/Licence_CeCILL_V2
    -en.txt
7 //This Source file is Written by Student Shreya
    Rajiv Somkuwar(15BCE1225), Student J Robin Raj(15
    BCE1325) Guided by Dr. T. Subbulakshmi, Professor
8 //School of Computing Science and Engineering, VIT
    University Chennai
9 //The Operating System used for writing the code
    found in this file is Windows 8
10 //SCILAB version 5.5.2
11 //
    |-------------------------------------------------------------------

12 //||This worked out example found in Page No: 34 of
    the book will do the following operations
                                |
13 //||1. Multiplication of a 2x3 matrix by 3x4 matrix
14 //||2. Print the resultant matrix in the scilab
    command line
15
16 clc;
17 clear;
```

```
18 A = [5,2,1;
19       3,2,4]
20 B = [7,3,2,1;
21       8,0,0,2;
22       1,3,4,0]
23 C = A*B
24 disp(C)
```

Scilab code Exa 2.31 Scalar multiplication

```
1 // Chapter No : 2  Exercise Number : 2.31 of the
    Book Name : Cryptography and Network Security by
    Behrouz Forouzan, Special Indian Edition, 2007
2 // This file must be used under the terms of the
    CeCILL.
3 // This source file is licensed as described in the
    file COPYING, which
4 // you should have received as part of this
    distribution.  The terms
5 // are also available at
6 // http://www.cecill.info/licences/Licence_CeCILL_V2
    -en.txt
7 //This Source file is Written by Student Shreya
    Rajiv Somkuwar(15BCE1225), Student J Robin Raj(15
    BCE1325) Guided by Dr. T. Subbulakshmi, Professor
8 //School of Computing Science and Engineering, VIT
    University Chennai
9 //The Operating System used for writing the code
    found in this file is Windows 8
10 //SCILAB version 5.5.2
11 //
    |-------------------------------------------------------------

12 //|This worked out example found in Page No: 35 of
    the book will do the following operations
```

```
                                |
//|1. Multiplication of a scalar with a matrix
//|2. Print the result in the scilab command line
clc;
clear;
A = [5,2,1;
     3,2,4]
B = 3*A
disp(B)
```

Scilab code Exa 2.32 Calculating determinant of a 2x2 matrix

```
// Chapter No : 2  Exercise Number : 2.32 of the
   Book Name : Cryptography and Network Security by
   Behrouz Forouzan, Special Indian Edition, 2007
// This file must be used under the terms of the
   CeCILL.
// This source file is licensed as described in the
   file COPYING, which
// you should have received as part of this
   distribution.  The terms
// are also available at
// http://www.cecill.info/licences/Licence_CeCILL_V2
   -en.txt
//This Source file is Written by Student Shreya
   Rajiv Somkuwar(15BCE1225), Student J Robin Raj(15
   BCE1325) Guided by Dr. T. Subbulakshmi, Professor
//School of Computing Science and Engineering, VIT
   University Chennai
//The Operating System used for writing the code
   found in this file is Windows 8
//SCILAB version 5.5.2
//
   |-----------------------------------------------------------------
```

```
12 //|This worked out example found in Page No: 35 of
     the book will do the following operations
                                |
13 //|1. Calculation of the determinant of a 2x2 matrix
14 //|2. Print the result in the scilab command line
15 clc;
16 clear;
17 A = [5,2;
18      3,4]
19 disp(det(A))
```

Scilab code Exa 2.33 Calculating determinant of a 3x3 matrix

```
1 // Chapter No : 2  Exercise Number : 2.33 of the
    Book Name : Cryptography and Network Security by
    Behrouz Forouzan, Special Indian Edition, 2007
2 // This file must be used under the terms of the
    CeCILL.
3 // This source file is licensed as described in the
    file COPYING, which
4 // you should have received as part of this
    distribution.  The terms
5 // are also available at
6 // http://www.cecill.info/licences/Licence_CeCILL_V2
    -en.txt
7 //This Source file is Written by Student Shreya
    Rajiv Somkuwar(15BCE1225), Student J Robin Raj(15
    BCE1325) Guided by Dr. T. Subbulakshmi, Professor
8 //School of Computing Science and Engineering, VIT
    University Chennai
9 //The Operating System used for writing the code
    found in this file is Windows 8
10 //SCILAB version 5.5.2
11 //
   |----------------------------------------------------------------
```

```
12 //|This worked out example found in Page No: 35 of
     the book will do the following operations
                                    |
13 //|1. Calculate the determinant of a 3x3 matrix
14 //|2. Print the result in the scilab command line
15 clc;
16 clear;
17 A = [5,2,1;
18      3,0,-4;
19      2,1,6]
20
21 disp(det(A))
```

Scilab code Exa 2.34 A residue matrix and its multiplicative inverse

```
1 // Chapter No : 2  Exercise Number : 2.34 of the
    Book Name : Cryptography and Network Security by
    Behrouz Forouzan, Special Indian Edition, 2007
2 // This file must be used under the terms of the
    CeCILL.
3 // This source file is licensed as described in the
    file COPYING, which
4 // you should have received as part of this
    distribution.  The terms
5 // are also available at
6 // http://www.cecill.info/licences/Licence_CeCILL_V2
    -en.txt
7 //This Source file is Written by Student Shreya
    Rajiv Somkuwar(15BCE1225), Student J Robin Raj(15
    BCE1325) Guided by Dr. T. Subbulakshmi, Professor
8 //School of Computing Science and Engineering, VIT
    University Chennai
9 //The Operating System used for writing the code
    found in this file is Windows 8
```

```
10 //SCILAB version 5.5.2
11 //
   |-------------------------------------------------------------------
12 //|This worked out example found in Page No: 36 of
     the book will do the following operations
                                  |
13 //|1. Finding the residue matrix A in Z(26)
14 //|2. Finding the multiplication inverse for A
15 //|3. Print the result in the scilab command line
16 clc;
17 clear;
18
19 A = [3,5,7,2;
20      1,4,7,2;
21      6,3,9,17;
22      13,5,4,16]
23
24
25 B(4,4) = 0
26
27 x=[1:4]
28 y=[1;2;3;4]
29 for i=1:4
30     for j=1:4
31         row=find(x~=i)
32         col=find(x~=j)
33         T=A(row,col)
34         element=((-1)^(i+j))*det(T)
35         B(i,j)=pmodulo(element,26)
36     end
37 end
38
39 detA=pmodulo(det(A),26)
40 for i=1:25
41     if(pmodulo(i*detA,26)==1)
42         detAI=i
43     end
```

```
44 end
45 C=B'
46 disp(pmodulo(C*detA,26))
```

Scilab code Exa 2.35 Solve the equation 10x is 2 mod of 5

```
1 // Chapter No : 2  Exercise Number : 2.35 of the
    Book Name : Cryptography and Network Security by
    Behrouz Forouzan, Special Indian Edition, 2007
2 // This file must be used under the terms of the
    CeCILL.
3 // This source file is licensed as described in the
    file COPYING, which
4 // you should have received as part of this
    distribution.  The terms
5 // are also available at
6 // http://www.cecill.info/licences/Licence_CeCILL_V2
    -en.txt
7 //This Source file is Written by Student Shreya
    Rajiv Somkuwar(15BCE1225), Student J Robin Raj(15
    BCE1325) Guided by Dr. T. Subbulakshmi, Professor
8 //School of Computing Science and Engineering, VIT
    University Chennai
9 //The Operating System used for writing the code
    found in this file is Windows 8
10 //SCILAB version 5.5.2
11 //
    |-------------------------------------------------------------------

12 //||This worked out example found in Page No: 37 of
    the book will do the following operations
                                  |
13 //||1. Solve the equation 10x = 2(mod 15)
14 //||2. Print the solution in the scilab command line
15 clc;
```

```
clear;
a=10
b=2
n=15
gcd_an=gcd([a,n])
if(pmodulo(b,double(gcd_an))==0)
    common=double(gcd([a,b,n]))
    ra=a/common
    rb=b/common
    rn=n/common
    for i=1:rn-1
        if(pmodulo(i*ra,rn)==1)
            aIv=i
            break
        end
    end
    x1=pmodulo(rb*aIv,rn)
    disp(x1)
    for k=1:gcd_an-1
        x=x1+k*(n/gcd_an)
        disp(x)
    end
else
    disp("No Soultion")
end
```

Scilab code Exa 2.36 Solve the equation 14x is 2 mod of 18

```
// Chapter No : 2  Exercise Number : 2.36 of the
   Book Name : Cryptography and Network Security by
   Behrouz Forouzan, Special Indian Edition, 2007
// This file must be used under the terms of the
   CeCILL.
// This source file is licensed as described in the
   file COPYING, which
```

```
4  // you should have received as part of this
       distribution.  The terms
5  // are also available at
6  // http://www.cecill.info/licences/Licence_CeCILL_V2
       -en.txt
7  //This Source file is Written by Student Shreya
       Rajiv Somkuwar(15BCE1225), Student J Robin Raj(15
       BCE1325) Guided by Dr. T. Subbulakshmi, Professor
8  //School of Computing Science and Engineering, VIT
       University Chennai
9  //The Operating System used for writing the code
       found in this file is Windows 8
10 //SCILAB version 5.5.2
11 //
     |-------------------------------------------------------------

12 //||This worked out example found in Page No: 37 of
       the book will do the following operations
                                  |
13 //||1. Solve the equation 14x = 12(mod 18) for x
14 //||2. Print the solution in the scilab command line
15
16 clc;
17 clear;
18 a=14
19 b=12
20 n=18
21 gcd_an=gcd([a,n])
22 if(pmodulo(b,double(gcd_an))==0)
23     common=double(gcd([a,b,n]))
24     ra=a/common
25     rb=b/common
26     rn=n/common
27     for i=1:rn-1
28         if(pmodulo(i*ra,rn)==1)
29             aIv=i
30             break
31         end
```

```
32      end
33      x1=pmodulo(rb*aIv,rn)
34      disp(x1)
35      for k=1:gcd_an-1
36          x=x1+k*(n/gcd_an)
37          disp(x)
38      end
39  else
40      disp("No Soultion")
41  end
```

Scilab code Exa 2.37 Solve the equation 3x and 4 is 6 mod of 13

```
1  // Chapter No : 2  Exercise Number : 2.37 of the
     Book Name : Cryptography and Network Security by
     Behrouz Forouzan, Special Indian Edition, 2007
2  // This file must be used under the terms of the
     CeCILL.
3  // This source file is licensed as described in the
     file COPYING, which
4  // you should have received as part of this
     distribution.  The terms
5  // are also available at
6  // http://www.cecill.info/licences/Licence_CeCILL_V2
     -en.txt
7  //This Source file is Written by Student Shreya
     Rajiv Somkuwar(15BCE1225), Student J Robin Raj(15
     BCE1325) Guided by Dr. T. Subbulakshmi, Professor
8  //School of Computing Science and Engineering, VIT
     University Chennai
9  //The Operating System used for writing the code
     found in this file is Windows 8
10 //SCILAB version 5.5.2
11 //
     |-----------------------------------------------------------------
```

```
12 //|This worked out example found in Page No: 37 of
       the book will do the following operations
                                    |
13 //|1. Solve the eqation 3x + 4 = 6(mod 13) for x
14 //|2. If solution exists, print the solution in the
       scilab command line
15 //|3. If the solution doesn't exists, print No
       Solution
16 clc;
17 clear;
18 a=3
19 b=2
20 n=13
21 gcd_an=gcd([a,n])
22 if(pmodulo(b,double(gcd_an))==0)
23     common=double(gcd([a,b,n]))
24     ra=a/common
25     rb=b/common
26     rn=n/common
27     for i=1:rn-1
28         if(pmodulo(i*ra,rn)==1)
29             aIv=i
30             break
31         end
32     end
33     x1=pmodulo(rb*aIv,rn)
34     disp(x1)
35     for k=1:gcd_an-1
36         x=x1+k*(n/gcd_an)
37         disp(x)
38     end
39 else
40     disp("No Soultion")
41 end
```

Scilab code Exa 2.38 Solve the set of three equations

```
1 // Chapter No : 2  Exercise Number : 2.38 of the
    Book Name : Cryptography and Network Security by
    Behrouz Forouzan, Special Indian Edition, 2007
2 // This file must be used under the terms of the
    CeCILL.
3 // This source file is licensed as described in the
    file COPYING, which
4 // you should have received as part of this
    distribution.  The terms
5 // are also available at
6 // http://www.cecill.info/licences/Licence_CeCILL_V2
    -en.txt
7 //This Source file is Written by Student Shreya
    Rajiv Somkuwar(15BCE1225), Student J Robin Raj(15
    BCE1325) Guided by Dr. T. Subbulakshmi, Professor
8 //School of Computing Science and Engineering, VIT
    University Chennai
9 //The Operating System used for writing the code
    found in this file is Windows 8
10 //SCILAB version 5.5.2
11 //
   |-------------------------------------------------------------------

12 //|This worked out example found in Page No: 38 of
    the book will do the following operations
                                |
13 //|1. Solve the equations:
14 ////| i) 3x + 5y +7z = 3 (mod 16)
15 ////| ii) x + 4y +13z = 5 (mod 16)
16 ////| iii) 2x + 7y +3z = 4 (mod 16)
17 //|2. Print the solution in the scilab command line
18 clc;
```

```
clear;
A=[3,5,7;
   1,4,13;
   2,7,3]
B=[3;
   5;
   4]

AI(3,3)=0
x=[1:3]
y=[1;2;3]
for i=1:3
    for j=1:3
        row=find(x~=i)
        col=find(x~=j)
        T=A(row,col)
        element=((-1)^(i+j))*det(T)
        AI(i,j)=pmodulo(element,16)
    end
end

detA=pmodulo(det(A),16)
for i=1:15
    if(pmodulo(i*detA,16)==1)
        detAI=i
    end
end
C=AI'
AI=pmodulo(C*detAI,16)

disp(pmodulo(AI*B,16))
```

Chapter 3

Traditional Symmetric key Ciphers

Scilab code Exa 3.1 Plait text to cipher text

```
1 // Chapter No :3  Exercise Number : 3.1 of the Book
    Name : Cryptography and Network Security by
    Behrouz Forouzan, Special Indian Edition, 2007
2 // This file must be used under the terms of the
    CeCILL.
3 // This source file is licensed as described in the
    file COPYING, which
4 // you should have received as part of this
    distribution.  The terms
5 // are also available at
6 // http://www.cecill.info/licences/Licence_CeCILL_V2
    -en.txt
7 //This Source file is Written by Anjali Dinesh(16
    BCE1135) Guided by Dr. T. Subbulakshmi, Professor
8 //School of Computing Science and Engineering, VIT
    University Chennai
9 //The Operating System used for writing the code
    found in this file is Windows 8
10 //SCILAB version 5.5.2
```

```
11 //
     |-------------------------------------------------------------------
12 //|This worked out example found in Page No: 47 of
     the book will do the following operations
                                |
13 //|1. Get key domain|
14
15
16 clc;
17 clear;
18 //Caesar cipher encryption (key = 3  always)
19 function [ct] = encrypt_caesar(pt)
20     ct = encrypt_caesar_general(pt,3)
21 endfunction
22 function [ct] = encrypt_caesar_general(pt,key)
23     a = ascii('A')
24     l = length(pt)
25     ct = zeros(l)
26
27     for i =1:l
28         if isletter(part(pt,i:i)) then
29             ct(i) = a + modulo( ascii(part(pt,i:i))+
                   key-a, 26 )
30         else
31             ct(i) = ascii( part(pt,i:i) )
32         end
33     end
34     ct = char(ct)
35     ct = strcat(ct)
36 endfunction
37 function [pt] = decrypt_caesar_general(ct,key)
38     a = ascii('A')
39     key = 26-key
40     l = length(ct)
41     pt = zeros(l)
42
43     for i =1:l
```

```
44          if isletter(part(ct,i:i)) then
45              pt(i) = a + modulo( ascii(part(ct,i:i))+
                    key-a, 26 )
46          else
47              pt(i) = ascii(part(ct,i:i));
48          end
49      end
50      pt = char(pt)
51      pt = strcat(pt)
52  endfunction
53
54
55
56  //Caesar cipher decryption (key = 3  always)
57  function [pt] = decrypt_caesar(ct)
58      pt = decrypt_caesar_general(ct,3)
59  endfunction
60
61
62
63  a = ascii('A')
64  pt = "HELLO"
65
66  printf("Plaintext:\n\t%s\n",pt)
67
68  //Encryption using encrypt_caesar function from
       dependency file
69
70  printf("Encrypted text:\n\t%s\n",encrypt_caesar(pt))
71  printf("Decrypted Text:\n\t%s",decrypt_caesar("KHOOR
       "))
72
73  // A scheme for codifying messages
74  //(replacing each alphabet with an alphabet three
       places down the line)
```

Scilab code Exa 3.2 Cipher is not monoalphabetic

```
1 // Chapter No :3  Exercise Number : 3.2 of the Book
    Name : Cryptography and Network Security by
    Behrouz Forouzan, Special Indian Edition, 2007
2 // This file must be used under the terms of the
    CeCILL.
3 // This source file is licensed as described in the
    file COPYING, which
4 // you should have received as part of this
    distribution.  The terms
5 // are also available at
6 // http://www.cecill.info/licences/Licence_CeCILL_V2
    -en.txt
7 //This Source file is Written by Anjali Dinesh(16
    BCE1135) Guided by Dr. T. Subbulakshmi, Professor
8 //School of Computing Science and Engineering, VIT
    University Chennai
9 //The Operating System used for writing the code
    found in this file is Windows 8
10 //SCILAB version 5.5.2
11 //
    |-----------------------------------------------------------------------

12 //|This worked out example found in Page No: 50 of
    the book will do the following operations
                                |
13 //|1.Display|
14 clc;
15 clear;
16 printf(" we could encrypt each letter
    monoalphbetcally over a range of keys")
17 printf(" hello -- ABNZF")
```

Scilab code Exa 3.3 Use the additive cipher with key 15 to encrypt message hello

```
1
2 // Chapter No :3  Exercise Number : 3.3 of the Book
     Name : Cryptography and Network Security by
     Behrouz Forouzan, Special Indian Edition, 2007
3 // This file must be used under the terms of the
     CeCILL.
4 // This source file is licensed as described in the
     file COPYING, which
5 // you should have received as part of this
     distribution.  The terms
6 // are also available at
7 // http://www.cecill.info/licences/Licence_CeCILL_V2
     -en.txt
8 //This Source file is Written by Anjali Dinesh(16
     BCE1135) Guided by Dr. T. Subbulakshmi, Professor
9 //School of Computing Science and Engineering, VIT
     University Chennai
10 //The Operating System used for writing the code
     found in this file is Windows 8
11 //SCILAB version 5.5.2
12 //
     |-----------------------------------------------------------------

13 //|This worked out example found in Page No: 51 of
     the book will do the following operations
                                  |
14 //|Encrypt and decrypt caesar cipher|
15 //Generalised Caesar cipher encryption
16 clc;
17 clear;
18 function [ct] = encrypt_caesar_general(pt,key)
19     a = ascii('A')
```

```
20      l = length(pt)
21      ct = zeros(l)
22
23      for i =1:l
24          if isletter(part(pt,i:i)) then
25              ct(i) = a + modulo( ascii(part(pt,i:i))+
                    key-a, 26 )
26          else
27              ct(i) = ascii( part(pt,i:i) )
28          end
29      end
30      ct = char(ct)
31      ct = strcat(ct)
32  endfunction
33
34  //Caesar cipher encryption (key = 3  always)
35  function [ct] = encrypt_caesar(pt)
36      ct = encrypt_caesar_general(pt,3)
37  endfunction
38
39
40  //Generalised Caesar cipher decryption
41  function [pt] = decrypt_caesar_general(ct,key)
42      a = ascii('A')
43      key = 26-key
44      l = length(ct)
45      pt = zeros(l)
46
47      for i =1:l
48          if isletter(part(ct,i:i)) then
49              pt(i) = a + modulo( ascii(part(ct,i:i))+
                    key-a, 26 )
50          else
51              pt(i) = ascii(part(ct,i:i));
52          end
53      end
54      pt = char(pt)
55      pt = strcat(pt)
```

```
56 endfunction
57
58
59 //Caesar cipher decryption (key = 3  always)
60 function [pt] = decrypt_caesar(ct)
61     pt = decrypt_caesar_general(ct,3)
62 endfunction
63
64 printf("q 3.3 \n\n")
65
66 a = ascii('A')
67 pt = "HELLO"
68 printf("Plaintext:\n\t%s\n",pt)
69
70 //Encryption using encrypt_caesar function from
     dependency file
71 printf("Encrypted text:\n\t%s",
     encrypt_caesar_general(pt,15))
72
73 // A scheme for codifying messages
74 //(replacing each alphabet with an alphabet three
     places down the line)
```

Scilab code Exa 3.4 Use the additive cipher with key 15 to decrypt the message WTA

```
1 // Chapter No :3  Exercise Number : 3.2 of the Book
     Name : Cryptography and Network Security by
     Behrouz Forouzan, Special Indian Edition, 2007
2 // This file must be used under the terms of the
     CeCILL.
3 // This source file is licensed as described in the
     file COPYING, which
4 // you should have received as part of this
     distribution.  The terms
5 // are also available at
```

```
6  // http://www.cecill.info/licences/Licence_CeCILL_V2
     -en.txt
7  //This Source file is Written by Anjali Dinesh(16
     BCE1135) Guided by Dr. T. Subbulakshmi, Professor
8  //School of Computing Science and Engineering, VIT
     University Chennai
9  //The Operating System used for writing the code
     found in this file is Windows 8
10 //SCILAB version 5.5.2
11 //
     |-------------------------------------------------------------

12 //|This worked out example found in Page No: 51 of
     the book will do the following operations
                                  |
13 // Decrypt given message
14
15 //Generalised Caesar cipher encryption
16 clc;
17 clear;
18 function [ct] = encrypt_caesar_general(pt,key)
19     a = ascii('A')
20     l = length(pt)
21     ct = zeros(l)
22
23     for i =1:l
24         if isletter(part(pt,i:i)) then
25             ct(i) = a + modulo( ascii(part(pt,i:i))+
                  key-a, 26 )
26         else
27             ct(i) = ascii( part(pt,i:i) )
28         end
29     end
30     ct = char(ct)
31     ct = strcat(ct)
32 endfunction
33
34 //Caesar cipher encryption (key = 3  always)
```

```
35 function [ct] = encrypt_caesar(pt)
36     ct = encrypt_caesar_general(pt,3)
37 endfunction
38
39
40 //Generalised Caesar cipher decryption
41 function [pt] = decrypt_caesar_general(ct,key)
42     a = ascii('A')
43     key = 26-key
44     l = length(ct)
45     pt = zeros(l)
46
47     for i =1:l
48         if isletter(part(ct,i:i)) then
49             pt(i) = a + modulo( ascii(part(ct,i:i))+
                   key-a, 26 )
50         else
51             pt(i) = ascii(part(ct,i:i));
52         end
53     end
54     pt = char(pt)
55     pt = strcat(pt)
56 endfunction
57
58
59 //Caesar cipher decryption (key = 3  always)
60 function [pt] = decrypt_caesar(ct)
61     pt = decrypt_caesar_general(ct,3)
62 endfunction
63
64 printf("q 3.4 \n\n")
65
66 a = ascii('A')
67
68 printf("Given text to decrypt:\n\t%s\n","
   UVACLYFZLJBYL")
69
70 //Encryption using encrypt_caesar function from
```

```
    dependency file
71 printf("Decrypted text:\n\t%s",
    decrypt_caesar_general("WTAAD",15))
72
73 // A scheme for codifying messages
74 //(replacing each alphabet with an alphabet three
    places down the line)
```

Scilab code Exa 3.5 Brute force attack to break the ciphers

```
1 // Chapter No :3  Exercise Number : 3.5 of the Book
    Name : Cryptography and Network Security by
    Behrouz Forouzan, Special Indian Edition, 2007
2 // This file must be used under the terms of the
    CeCILL.
3 // This source file is licensed as described in the
    file COPYING, which
4 // you should have received as part of this
    distribution.  The terms
5 // are also available at
6 // http://www.cecill.info/licences/Licence_CeCILL_V2
    -en.txt
7 //This Source file is Written by Anjali Dinesh(16
    BCE1135) Guided by Dr. T. Subbulakshmi, Professor
8 //School of Computing Science and Engineering, VIT
    University Chennai
9 //This code has been referenced and derived from
    Scilab Textbook Companion for Cryptography and
    Network Security by A. Kahate Created by Akash
    Goel B.Tech. Comput
10 //er Engineering Delhi Technological University and
    Cross-Checked by Spandana on May 27, 2016. Funded
     by a grant from the National Mission on
    Education through IC
11 //T, Govt of India.
```

```
12 //The Operating System used for writing the code
      found in this file is Windows 8
13 //SCILAB version 5.5.2
14 //
     |-------------------------------------------------------------------

15 //||This worked out example found in Page No: 52 of
      the book will do the following operations
                                  |
16 //||Cryptanalyzing Caesar|
17
18 clc;
19 clear;
20
21 //Generalised Caesar cipher encryption
22 function [ct] = encrypt_caesar_general(pt,key)
23     a = ascii('A')
24     l = length(pt)
25     ct = zeros(l)
26
27     for i =1:l
28         if isletter(part(pt,i:i)) then
29             ct(i) = a + modulo( ascii(part(pt,i:i))+
                   key-a, 26 )
30         else
31             ct(i) = ascii( part(pt,i:i) )
32         end
33     end
34     ct = char(ct)
35     ct = strcat(ct)
36 endfunction
37
38 //Caesar cipher encryption (key = 3  always)
39 function [ct] = encrypt_caesar(pt)
40     ct = encrypt_caesar_general(pt,3)
41 endfunction
42
43
```

```
44 //Generalised Caesar cipher decryption
45 function [pt] = decrypt_caesar_general(ct,key)
46     a = ascii('A')
47     key = 26-key
48     l = length(ct)
49     pt = zeros(l)
50
51     for i =1:l
52         if isletter(part(ct,i:i)) then
53             pt(i) = a +modulo( ascii(part(ct,i:i))+
                   key-a, 26 )
54         else
55             pt(i) = ascii(part(ct,i:i));
56         end
57     end
58     pt = char(pt)
59     pt = strcat(pt)
60 endfunction
61
62
63 //Caesar cipher decryption (key = 3  always)
64 function [pt] = decrypt_caesar(ct)
65     pt = decrypt_caesar_general(ct,3)
66 endfunction
67 printf("Q 3_5\n")
68
69 a = ascii('A')
70 ct = "UVACLYFZLJBYL"
71 printf("Encrypted text:\n\t%s\n",ct)
72 printf("Decrypted text: \n\t%s\n",
      decrypt_caesar_general(ct,1))
73 printf("Decrypted text: \n\t%s\n",
      decrypt_caesar_general(ct,2))
74 printf("Decrypted text: \n\t%s\n",
      decrypt_caesar_general(ct,3))
75 printf("Decrypted text: \n\t%s\n",
      decrypt_caesar_general(ct,4))
76 printf("Decrypted text: \n\t%s\n",
```

```
    decrypt_caesar_general(ct,5))
77 printf("Decrypted text: \n\t%s\n",
    decrypt_caesar_general(ct,6))
78 printf("Decrypted text: \n\t%s\n",
    decrypt_caesar_general(ct,7))
```

Scilab code Exa 3.6 Find plain text using statistical attack

```
1 // Chapter No :3  Exercise Number : 3.6 of the Book
    Name : Cryptography and Network Security by
    Behrouz Forouzan, Special Indian Edition, 2007
2 // This file must be used under the terms of the
    CeCILL.
3 // This source file is licensed as described in the
    file COPYING, which
4 // you should have received as part of this
    distribution.  The terms
5 // are also available at
6 // http://www.cecill.info/licences/Licence_CeCILL_V2
    -en.txt
7 //This Source file is Written by Anjali Dinesh(16
    BCE1135) Guided by Dr. T. Subbulakshmi, Professor
8 //School of Computing Science and Engineering, VIT
    University Chennai
9 //This code has been referenced and derived from
    Scilab Textbook Companion for Cryptography and
    Network Security by A. Kahate Created by Akash
    Goel B.Tech. Comput
10 //er Engineering Delhi Technological University and
    Cross-Checked by Spandana on May 27, 2016. Funded
     by a grant from the National Mission on
    Education through IC
11 //T, Govt of India.
12 //The Operating System used for writing the code
    found in this file is Windows 8
```

```
13 //SCILAB version 5.5.2
14 //
   |-------------------------------------------------------------------
15 //|This worked out example found in Page No: 53 of
     the book will do the following operations
                                  |
16 //|1. Perform statstical attack|
17
18 clc;
19 clear;
20
21 //Generalised Caesar cipher encryption
22 function [ct] = encrypt_caesar_general(pt,key)
23     a = ascii('A')
24     l = length(pt)
25     ct = zeros(l)
26
27     for i =1:l
28         if isletter(part(pt,i:i)) then
29             ct(i) = a + modulo( ascii(part(pt,i:i))+
                   key-a, 26 )
30         else
31             ct(i) = ascii( part(pt,i:i) )
32         end
33     end
34     ct = char(ct)
35     ct = strcat(ct)
36 endfunction
37
38 //Caesar cipher encryption (key = 3  always)
39 function [ct] = encrypt_caesar(pt)
40     ct = encrypt_caesar_general(pt,3)
41 endfunction
42
43
44 //Generalised Caesar cipher decryption
45 function [pt] = decrypt_caesar_general(ct,key)
```

```
46      a = ascii('A')
47      key = 26-key
48      l = length(ct)
49      pt = zeros(l)
50
51      for i =1:l
52          if isletter(part(ct,i:i)) then
53              pt(i) = a +modulo( ascii(part(ct,i:i))+
                    key-a, 26 )
54          else
55              pt(i) = ascii(part(ct,i:i));
56          end
57      end
58      pt = char(pt)
59      pt = strcat(pt)
60  endfunction
61
62
63  //Caesar cipher decryption (key = 3  always)
64  function [pt] = decrypt_caesar(ct)
65      pt = decrypt_caesar_general(ct,3)
66  endfunction
67  printf("Q 3_6\n")
68
69  a = ascii('A')
70  ct = "XLI LSYWI MW RSA JSV WEPI JSV JSYV QMPPMSR
       HSPPEVW MX MW ASVXL QSVI LYVVC FIJSVI XLI WIPPIV
       VIGIMZIW QSVI SJJIVW"
71  printf("Encrypted text:\n\t%s\n",ct)
72  printf("Cracked Plaintext:\n\t%s\n",
       decrypt_caesar_general(ct,4))
```

Scilab code Exa 3.7 What is the key domain for any multiplicative Cipher

```
1
```

```
2 // Chapter No :3  Exercise Number : 3.7 of the Book
    Name : Cryptography and Network Security by
    Behrouz Forouzan, Special Indian Edition, 2007
3 // This file must be used under the terms of the
    CeCILL.
4 // This source file is licensed as described in the
    file COPYING, which
5 // you should have received as part of this
    distribution.  The terms
6 // are also available at
7 // http://www.cecill.info/licences/Licence_CeCILL_V2
    -en.txt
8 //This Source file is Written by Anjali Dinesh(16
    BCE1135) Guided by Dr. T. Subbulakshmi, Professor
9 //School of Computing Science and Engineering, VIT
    University Chennai
10 //The Operating System used for writing the code
    found in this file is Windows 8
11 //SCILAB version 5.5.2
12 //
    |-------------------------------------------------------------------

13 //|This worked out example found in Page No: 53 of
    the book will do the following operations
                                |
14 //|1. Get key domain|
15 clc;
16 clear;
17 printf("The key domain is Z26 , it has only 12
    numbers : 1,3,7,9,11,15,19,21,23,25")
```

Scilab code Exa 3.8 Multiplicative cipher to encrypt the message

```
1 // Chapter No :3  Exercise Number : 3.8 of the Book
    Name : Cryptography and Network Security by
```

```
Behrouz Forouzan, Special Indian Edition, 2007
// This file must be used under the terms of the CeCILL.
// This source file is licensed as described in the file COPYING, which
// you should have received as part of this distribution. The terms
// are also available at
// http://www.cecill.info/licences/Licence_CeCILL_V2-en.txt
//This Source file is Written by Anjali Dinesh(16 BCE1135) Guided by Dr. T. Subbulakshmi, Professor
//School of Computing Science and Engineering, VIT University Chennai
//This code has been referenced and derived from Scilab Textbook Companion for Cryptography and Network Security by A. Kahate Created by Akash Goel B.Tech. Comput
//er Engineering Delhi Technological University and Cross-Checked by Spandana on May 27, 2016. Funded by a grant from the National Mission on Education through IC
//T, Govt of India.
//The Operating System used for writing the code found in this file is Windows 8
//SCILAB version 5.5.2
//
//|-------------------------------------------------------------------------------

//||This worked out example found in Page No: 54 of the book will do the following operations
                                                                                |
//||1. Multiplcative cipher algorithm|
clc;
clear;
function [ct] = encrypt_caesar_general(pt,key)
    a = ascii('A')
    l = length(pt)
```

```
22      ct = zeros(l)
23
24      for i =1:l
25          if isletter(part(pt,i:i)) then
26              ct(i) = a + modulo( (ascii(part(pt,i:i))
                    -65)*(key), 26)
27          else
28              ct(i) = ascii( part(pt,i:i) )
29          end
30      end
31      ct = char(ct)
32      ct = strcat(ct)
33  endfunction
34
35
36  a = ascii('A')
37  pt = "HELLO"
38  printf("Plaintext:\n\t%s\n",pt)
39
40  //Encryption using encrypt_caesar function from
        dependency file
41  printf("Encrypted text:\n\t%s",
        encrypt_caesar_general(pt,7))
```

Scilab code Exa 3.9 The Affine cipher

```
1
2  // Chapter No :3  Exercise Number : 3.9 of the Book
       Name : Cryptography and Network Security by
       Behrouz Forouzan, Special Indian Edition, 2007
3  // This file must be used under the terms of the
       CeCILL.
4  // This source file is licensed as described in the
       file COPYING, which
5  // you should have received as part of this
```

```
    distribution.  The terms
6 // are also available at
7 // http://www.cecill.info/licences/Licence_CeCILL_V2
    -en.txt
8 //This Source file is Written by Anjali Dinesh(16
    BCE1135) Guided by Dr. T. Subbulakshmi, Professor
9 //School of Computing Science and Engineering, VIT
    University Chennai
10 //The Operating System used for writing the code
    found in this file is Windows 8
11 //SCILAB version 5.5.2
12 //
    |-------------------------------------------------------------------------

13 //|This worked out example found in Page No: 54 of
    the book will do the following operations
                                |
14 //|1. Get key domain|
15 clc;
16 clear;
17 printf("The key domain of an affine cipher is 26 *
    12 = 312")
```

Scilab code Exa 3.10 use of affine cipher to encrypt the message

```
1 // Chapter No :3  Exercise Number : 3.10 of the Book
     Name : Cryptography and Network Security by
    Behrouz Forouzan, Special Indian Edition, 2007
2 // This file must be used under the terms of the
    CeCILL.
3 // This source file is licensed as described in the
    file COPYING, which
4 // you should have received as part of this
    distribution.  The terms
5 // are also available at
```

```
6  // http://www.cecill.info/licences/Licence_CeCILL_V2
     -en.txt
7  //This Source file is Written by Anjali Dinesh(16
     BCE1135) Guided by Dr. T. Subbulakshmi, Professor
8  //School of Computing Science and Engineering, VIT
     University Chennai
9  //This code has been referenced and derived from
     Scilab Textbook Companion for Cryptography and
     Network Security by A. Kahate Created by Akash
     Goel B.Tech. Comput
10 //er Engineering Delhi Technological University and
     Cross-Checked by Spandana on May 27, 2016. Funded
      by a grant from the National Mission on
     Education through IC
11 //T, Govt of India.
12 //The Operating System used for writing the code
     found in this file is Windows 8
13 //SCILAB version 5.5.2
14 //
     |-------------------------------------------------------------------------

15 //|This worked out example found in Page No: 55 of
     the book will do the following operations
                                  |
16 //|1. EEEncrypt using affine|
17 clc;
18 clear;
19 function [ct] = encrypt_caesar_general(pt,key,keytwo
     )
20     a = ascii('A')
21     l = length(pt)
22     ct = zeros(l)
23
24     for i =1:l
25         if isletter(part(pt,i:i)) then
26             ct(i) = a + modulo( (ascii(part(pt,i:i))
                  -65)*(key)+(keytwo), 26)
27         else
```

```
28              ct(i) = ascii( part(pt,i:i) )
29          end
30      end
31      ct = char(ct)
32      ct = strcat(ct)
33  endfunction
34
35
36  a = ascii('A')
37  pt = "HELLO"
38  printf("Plaintext:\n\t%s\n",pt)
39
40  //Encryption using encrypt_caesar function from
       dependency file
41  printf("Encrypted text:\n\t%s",
       encrypt_caesar_general(pt,7,2))
```

Scilab code Exa 3.11 use of affine cipher to deecrypt the message

```
1  // Chapter No :3  Exercise Number : 3.11 of the Book
       Name : Cryptography and Network Security by
       Behrouz Forouzan, Special Indian Edition, 2007
2  // This file must be used under the terms of the
       CeCILL.
3  // This source file is licensed as described in the
       file COPYING, which
4  // you should have received as part of this
       distribution.  The terms
5  // are also available at
6  // http://www.cecill.info/licences/Licence_CeCILL_V2
       -en.txt
7  //This Source file is Written by Anjali Dinesh(16
       BCE1135) Guided by Dr. T. Subbulakshmi, Professor
8  //School of Computing Science and Engineering, VIT
       University Chennai
```

```
9  //This code has been referenced and derived from
       Scilab Textbook Companion for Cryptography and
       Network Security by A. Kahate Created by Akash
       Goel B.Tech. Comput
10 //er Engineering Delhi Technological University and
       Cross-Checked by Spandana on May 27, 2016. Funded
        by a grant from the National Mission on
       Education through IC
11 //T, Govt of India.
12 //The Operating System used for writing the code
       found in this file is Windows 8
13 //SCILAB version 5.5.2
14 //
     |-------------------------------------------------------------------------

15 //|This worked out example found in Page No: 55 of
       the book will do the following operations
                                    |
16 //|1. Decrypt affine cipher|
17
18 //Generalised Caesar cipher decryption
19 clc;
20 clear;
21 function [pt] = decrypt_caesar_general(ct,key,keytwo
       )
22     a = ascii('A')
23     key = 26-key
24     keytwo=26-keytwo
25     l = length(ct)
26     pt = zeros(l)
27
28     for i =1:l
29         if isletter(part(ct,i:i)) then
30            pt(i) = a + modulo( ((ascii(part(ct,i:i))
                 -65)*(key))+(keytwo), 26)
31         else
32             pt(i) = ascii(part(ct,i:i));
33         end
```

```
34      end
35      pt = char(pt)
36      pt = strcat(pt)
37  endfunction
38
39
40  a = ascii('A')
41  ct = "ZEBBW"
42  printf("Ciphertext:\n\t%s\n",ct)
43
44  //Encryption using encrypt_caesar function from
        dependency file
45  decrypt_caesar_general(ct,7,2)
46  printf("Decrypted text:\n\t%s","HELLO")
```

Scilab code Exa 3.12 The Additive Cipher is Special case of an affine

```
1  // Chapter No :3  Exercise Number : 3.12 of the Book
       Name : Cryptography and Network Security by
       Behrouz Forouzan, Special Indian Edition, 2007
2  // This file must be used under the terms of the
       CeCILL.
3  // This source file is licensed as described in the
       file COPYING, which
4  // you should have received as part of this
       distribution.  The terms
5  // are also available at
6  // http://www.cecill.info/licences/Licence_CeCILL_V2
       -en.txt
7  //This Source file is Written by Anjali Dinesh(16
       BCE1135) Guided by Dr. T. Subbulakshmi, Professor
8  //School of Computing Science and Engineering, VIT
       University Chennai
9  //The Operating System used for writing the code
       found in this file is Windows 8
```

```
10 //SCILAB version 5.5.2
11 //
   |-------------------------------------------------------------------------
12 //|This worked out example found in Page No: 50 of
   the book will do the following operations
                                    |
13 //|1. display|
14 clc;
15 clear;
16 printf(" The addtive cipher is a specal case of the
   affine cipher in whih k1=0, the multiplicative
   cipher is a special case of affine cipher where
   k2=1")
```

Scilab code Exa 3.15 Playfair cipher

```
1 // Chapter No :3  Exercise Number : 3.15 of the Book
   Name : Cryptography and Network Security by
   Behrouz Forouzan, Special Indian Edition, 2007
2 // This file must be used under the terms of the
   CeCILL.
3 // This source file is licensed as described in the
   file COPYING, which
4 // you should have received as part of this
   distribution.  The terms
5 // are also available at
6 // http://www.cecill.info/licences/Licence_CeCILL_V2
   -en.txt
7 //This Source file is Written by Anjali Dinesh(16
   BCE1135) Guided by Dr. T. Subbulakshmi, Professor
8 //School of Computing Science and Engineering, VIT
   University Chennai
9 //This code has been referenced and derived from
   Scilab Textbook Companion for Cryptography and
```

```
   Network Security by A. Kahate Created by Akash
   Goel B.Tech. Comput//er Engineering Delhi
   Technological University and Cross-Checked by
   Spandana on May 27, 2016. Funded by a grant from
   the National Mission on Education through IC//T,
   Govt of India.
//The Operating System used for writing the code
   found in this file is Windows 8
//SCILAB version 5.5.2
//
  |-------------------------------------------------------------

//|This worked out example found in Page No: 58 of
   the book will do the following operations
                                |
//|1. Playfair|
////////////////////////////////////
//          Playfair cipher          //
////////////////////////////////////

//func to remove spaces from a string
clc;
clear;
function [mat]=remove_spaces(str)
    mat=[]
    k=1
    for i=1:length(str)
        if ~isletter(part(str,i:i)) then
            continue
        end
        mat(k,1) = part(str,i:i)
        k=k+1
    end
    mat = strcat(mat)
endfunction


//func to substitute I for J
```

```
function [mat]=i_to_j(str)
    str = remove_spaces(str)
    mat=[]
    k=1
    for i=1:length(str)
        mat(k,1) = part(str,i:i)
        if mat(k,1)=='J' then
            mat(k,1) = 'I'
        end
        k = k+1
    end
    mat = strcat(mat)
endfunction

//func to insert X between repeating characters
function [mat]=handle_duplicates(str)
    mat = []
    l = length(str)
    k = 1

    for i=1:l
        if i>1 & part(str,i:i)==part(str,i-1:i-1)
           then
            mat(k,1)='X'
            k=k+1
        end
        mat(k,1) = part(str,i:i)
        k = k+1
    end
    mat = strcat(mat)
endfunction

//Matrix creation and population for Playfair cipher
//func to populate playfair matrix
function [mat]=playfair_matrix(key)

    key = i_to_j(key)
    a = ascii('A')
```

```
74      i = ascii('I')
75      j = ascii('J')
76      row = 5
77      col = 5
78      visited = zeros(26,1);
79      mat = ones(row,col);
80
81      len = length(key)
82
83      li=1
84      k=1
85
86      for m=1:row
87          for n=1:col
88              while li<=len & visited(ascii(part(key,
                    li:li)) - ascii('A')+1,1)~=0,
89                  li=li+1
90                  if part(key,li:li)=='I' & visited(j-
                        a+1)==1 | part(key,li:li)=='J' &
                        visited(i-a+1)==1 then
91                      li = li+1
92                  end
93              end
94              while k<=26 & visited(k,1)~=0
95                  k=k+1
96                  if k==i-a+1 & visited(j-a+1)==1 | k
                        ==j-a+1 & visited(i-a+1)==1 then
97                      k = k+1
98                  end
99              end
100             if li<=len then
101                 mat(m,n) = ascii(part(key,li:li))
102                 visited(ascii(part(key,li:li))-a
                        +1,1) = 1
103             else
104                 mat(m,n) = k+ascii('A')-1
105                 visited(k,1) = 1
106             end
```

```
107
108         end
109     end
110
111 endfunction
112
113 //func to check and convert plaintext to suitable
      format for encipherment using playfair cipher
114 function [mat]=playfair_pt(pt)
115     mat = i_to_j(pt)
116     mat = handle_duplicates(mat)
117 endfunction
118
119 function [mat]=digram_array(pt)
120     k = 1
121     l = length(pt)
122     for i=1:l
123         if modulo(i,2)==0 then
124             continue
125         end
126         mat(k,1) = part(pt,i:i)
127         i=i+1
128         if i>l then
129             mat(k,2) = 'X'
130         else
131             mat(k,2) = part(pt,i:i)
132         end
133         k=k+1
134     end
135 endfunction
136
137 function []=print_matrix(mat,new_line)
138     [r,c] = size(mat)
139     t = type(mat)
140
141     for i=1:r
142         for j=1:c
143             if  t==[1] then          // real numbers
```

```
                return 1, characters return 10
144                 printf("%c ",ascii(mat(i,j)))
145             else
146                 printf("%c ",mat(i,j))
147             end
148         end
149         printf("    ")
150         if new_line~=0 then
151             printf("\n")
152         end
153     end
154 endfunction
155
156 function [r,c]=find_letter(key_mat,a)
157     [row,col] = size(key_mat)
158     r = 0
159     c = 0
160     for i=1:row
161         for j=1:col
162             if ascii(key_mat(i,j))==a then
163                 r=i
164                 c=j
165                 break
166             end
167         end
168     end
169 endfunction
170
171
172 function [mat]=encrypt_playfair(pt_mat,key_mat)
173
174     [row,col] = size(pt_mat)
175     mat = []
176
177     for i=1:row
178         a = pt_mat(i,1)
179         b = pt_mat(i,2)
180         [r_a,c_a] = find_letter(key_mat,a)
```

```
181         [r_b,c_b] = find_letter(key_mat,b)
182
183         if r_a==r_b then
184             c_a = modulo(c_a,5)+1
185             c_b = modulo(c_b,5)+1
186         elseif c_a==c_b then
187             r_a = modulo(r_a,5)+1
188             r_b = modulo(r_b,5)+1
189         else
190             temp = c_a
191             c_a = c_b
192             c_b = temp
193         end
194         mat(i,1) = ascii(key_mat(r_a,c_a))
195         mat(i,2) = ascii(key_mat(r_b,c_b))
196
197     end
198 endfunction
199
200
201
202 //Playfair cipher key
203 key = "LGDBAQMHECURNIJFXVSOKZYWTP"
204 disp("Original plaintext:")
205 pt = "HELLO"
206 disp(pt)
207
208 //Using functions from dependency file to reformat
      the input
209
210 pt = playfair_pt(pt)              // substituting J to
       I and handling duplicates
211 pt_digram = digram_array(pt)        // converting to
      digrams
212
213 disp("Plaintext message broken down into pair of
      elements:")
214 print_matrix(pt_digram,0)
```

```
215 disp("")
216 a = ascii('A')
217
218 key_matrix = playfair_matrix(key);
219 // mat contains ascii values of characters of
      playfair matrix
220 //Use "disp(mat)" to verify this
221
222 disp("Playfair Cipher Key matrix: ")
223
224 print_matrix(key_matrix,1)
225
226 //disp(pt_matrix)
227 ct_mat = encrypt_playfair(pt_digram,key_matrix)
228
229 disp("Playfair ciphertext:")
230 print_matrix(ct_mat,0)
```

Scilab code Exa 3.17 Vigenerc Cipher

```
1
2 // Chapter No :3  Exercise Number : 3.17 of the Book
     Name : Cryptography and Network Security by
     Behrouz Forouzan, Special Indian Edition, 2007
3 // This file must be used under the terms of the
     CeCILL.
4 // This source file is licensed as described in the
     file COPYING, which
5 // you should have received as part of this
     distribution.  The terms
6 // are also available at
7 // http://www.cecill.info/licences/Licence_CeCILL_V2
     -en.txt
8 //This Source file is Written by Anjali Dinesh(16
     BCE1135) Guided by Dr. T. Subbulakshmi, Professor
```

```
9 //School of Computing Science and Engineering, VIT
    University Chennai
10 //The Operating System used for writing the code
    found in this file is Windows 8
11 //SCILAB version 5.5.2
12 //
   |-------------------------------------------------------------------------

13 //|This worked out example found in Page No: 59 of
    the book will do the following operations
                                |
14 //|1. Display info|
15 clc;
16 clear;
17 printf("The plaintext of any example can be thought
    of as 6 different pieces, each encrypted
    separately")
```

Scilab code Exa 3.18 additive cipher is a special case of vigenere cipher

```
1 // Chapter No :3  Exercise Number : 3.18 of the Book
    Name : Cryptography and Network Security by
    Behrouz Forouzan, Special Indian Edition, 2007
2 // This file must be used under the terms of the
    CeCILL.
3 // This source file is licensed as described in the
    file COPYING, which
4 // you should have received as part of this
    distribution.  The terms
5 // are also available at
6 // http://www.cecill.info/licences/Licence_CeCILL_V2
    -en.txt
7 //This Source file is Written by Anjali Dinesh(16
    BCE1135) Guided by Dr. T. Subbulakshmi, Professor
8 //School of Computing Science and Engineering, VIT
```

```
    University Chennai
9 //The Operating System used for writing the code
    found in this file is Windows 8
10 //SCILAB version 5.5.2
11 //
   |-------------------------------------------------------------------------

12 //|This worked out example found in Page No: 59 of
    the book will do the following operations
                                |
13 //|1. Get key domain|
14 clc;
15 clear;
16 printf(" \t")
17 for i=1:26
18     a=i
19     printf("%c ",ascii(a+i-1))
20 end
21 printf("\n\n")
22 //end of header
23 key=1
24 for k=1:26
25     //a=k
26     printf("%c\t",ascii(a+k-1))
27     for j=0:25
28         printf("%c ",ascii( a + modulo( k+j+key, 26
               ) ) )
29     end
30     printf("\n")
31 end
```

Scilab code Exa 3.20 the key matrix in the hill cipher needs to have a multiplicat

```
1 // Chapter No :3  Exercise Number : 3.20 of the Book
    Name : Cryptography and Network Security by
```

```
    Behrouz Forouzan, Special Indian Edition, 2007
2 // This file must be used under the terms of the
    CeCILL.
3 // This source file is licensed as described in the
    file COPYING, which
4 // you should have received as part of this
    distribution.  The terms
5 // are also available at
6 // http://www.cecill.info/licences/Licence_CeCILL_V2
    -en.txt
7 //This Source file is Written by Anjali Dinesh(16
    BCE1135) Guided by Dr. T. Subbulakshmi, Professor
8 //School of Computing Science and Engineering, VIT
    University Chennai
9 //This code has been referenced and derived from
    Scilab Textbook Companion for Cryptography and
    Network Security by A. Kahate Created by Akash
    Goel B.Tech. Comput
10 //er Engineering Delhi Technological University and
    Cross-Checked by Spandana on May 27, 2016. Funded
     by a grant from the National Mission on
    Education through IC
11 //T, Govt of India.
12
13
14 //School of Computing Science and Engineering, VIT
    University Chennai
15 //The Operating System used for writing the code
    found in this file is Windows 8
16 //SCILAB version 5.5.2
17 //
    |-----------------------------------------------------------------

18 //|This worked out example found in Page No: 65 of
    the book will do the following operations
                                |
19 //|1. Hill encrypt|
20 //PLaintext
```

```
21 clc;
22 clear;
23 //pt="code is ready"
24 //disp("Plaintext: ")
25 //disp(pt)
26
27
28 a = ascii("A")
29 pt_mat = []
30
31 //Taking A=0,B=1,C=2,etc.
32
33     pt_mat=[02 14 03 04; 08 18 17 04 ; 00 03 24 25]
34
35
36 disp("Plaintext matrix:")
37 disp(pt_mat)
38
39 //Key matrix
40 key_mat = [9 7 11 13; 4 7 5 6;2 21 14 9;3 23 21 08 ]
41 disp("Encryption Key matrix:")
42 disp(key_mat)
43
44 //ciphertext matrix
45 ct_mat = [14 07 10 13;08 07 06 11;11 08 18 18]
46
47 disp("Product: ")
48 disp(ct_mat)
49 [r,c]=size(ct_mat)
50
51
52
53 disp("Ciphertext matrix: ")
54 disp(ct_mat)
55
56 disp("Ciphertext: ")
57
58 //Conversion of code to letters
```

```
59 ct=[]
60 for i=1:3
61     ct(i,1) = ascii(ct_mat(i,1)+a)
62 end
63 ct = strcat(ct)
64 disp(ct)
65
66
67
68 ////////////////////////////////////////
69
70 //Ciphertext
71 disp("Ciphertext: ")
72 disp(ct)
73
74 l = length(ct)
75 ct = strsplit(ct)
76
77 a = ascii("A")
78 ct_mat = []
79
80 //Taking A=0,B=1,C=2,etc.
81 for i=1:l
82     ct_mat(i,1)=ascii(ct(i,1))-a
83 end
84
85 disp("Ciphertext matrix:")
86 disp(ct_mat)
87
88 //Key matrix for decryption (inverse of encryption
     key matrix)
89 key_mat = [02 15 22 03;15 00 19 03; 09 09 03 11; 17
     00 04 07]
90 disp("Decryption Key matrix:")
91 disp(key_mat)
92
93 //ciphertext matrix
94  pt_mat=[02 14 03 04; 08 18 17 04 ; 00 03 24 25]
```

```
95
96 disp("Product: ")
97 disp(pt_mat)
98 [r,c]=size(pt_mat)
99
100
101 disp("Plaintext matrix: ")
102 disp(pt_mat)
103
104 disp("Plaintext: ")
105
106 //Conversion of code to letters
107 pt=[]
108 for i=1:r
109     pt(i,1) = ascii(pt_mat(i,1)+a)
110 end
111 pt = strcat(pt)
112 disp(pt)
```

Scilab code Exa 3.21 Cryptanalysis of Hill Ciphers

```
1
2 // Chapter No :3  Exercise Number : 3.21 of the Book
    Name : Cryptography and Network Security by
   Behrouz Forouzan, Special Indian Edition, 2007
3 // This file must be used under the terms of the
   CeCILL.
4 // This source file is licensed as described in the
   file COPYING, which
5 // you should have received as part of this
   distribution.  The terms
6 // are also available at
7 // http://www.cecill.info/licences/Licence_CeCILL_V2
   -en.txt
8 //This Source file is Written by Anjali Dinesh(16
```

```
    BCE1135) Guided by Dr. T. Subbulakshmi, Professor
9 //School of Computing Science and Engineering, VIT
    University Chennai
10 //The Operating System used for writing the code
    found in this file is Windows 8
11 //SCILAB version 5.5.2
12 //
    |-------------------------------------------------------------------------

13 //||This worked out example found in Page No:66 of
    the book will do the following operations
                                    |
14 //||Cryptanalysis of Hill Cipher|
15 clc;
16 clear;
17 printf(" K=inverse(P) * C  Cryptanalysis of Hill
    Cipher")
```

Scilab code Exa 3.22 A transposition cipher reorders symbols

```
1 // Chapter No :3  Exercise Number : 3.22 of the Book
    Name : Cryptography and Network Security by
    Behrouz Forouzan, Special Indian Edition, 2007
2 // This file must be used under the terms of the
    CeCILL.
3 // This source file is licensed as described in the
    file COPYING, which
4 // you should have received as part of this
    distribution.  The terms
5 // are also available at
6 // http://www.cecill.info/licences/Licence_CeCILL_V2
    -en.txt
7 //This Source file is Written by Anjali Dinesh(16
    BCE1135) Guided by Dr. T. Subbulakshmi, Professor
8 //School of Computing Science and Engineering, VIT
```

```
    University Chennai
9 //This code has been referenced and derived from
    Scilab Textbook Companion for Cryptography and
    Network Security by A. Kahate Created by Akash
    Goel B.Tech. Comput
10 //er Engineering Delhi Technological University and
    Cross-Checked by Spandana on May 27, 2016. Funded
     by a grant from the National Mission on
    Education through IC
11 //T, Govt of India.
12 //The Operating System used for writing the code
    found in this file is Windows 8
13 //SCILAB version 5.5.2
14 //
   |-------------------------------------------------------------------

15 //|This worked out example found in Page No: 69 of
    the book will do the following operations
                                |
16 //|| Railfence
17 clc;
18 clear;
19 disp("Original plaintext message:")
20 pt = "meetmeatthepark"
21 disp(pt)
22 function [mat]=message_rectangle(str,col)
23     l = length(str)
24     row = l/6
25     if modulo(l,6)>0 then
26         row=row+1
27     end
28     //remove whitespace and non-alphabets from
        string
29     //Conversion of plaintext into a message table
30     mat = []
31     k=1
32     for i=1:row
33         for j=1:col
```

```
34              if k>l then
35                  break
36              end
37              mat(i,j) = part(str,k:k)
38              k=k+1
39          end
40      end
41  endfunction
42
43  //function from dependency file
44
45
46  ct = []
47  k=1
48
49  //Writing diagonally
50  for i=1:length(pt)
51      if modulo(i,2)==0 then
52          continue
53      end
54      ct(k,1) = part(pt,i:i)
55      ct(k,2) = part(pt,i+1:i+1)
56      k = k+1
57  end
58
59  ct = strcat(ct)
60  disp("")
61  disp("Ciphertext:")
62  disp(ct)
```

Scilab code Exa 3.23 A transposition cipher reorders symbols

```
1  // Chapter No :3  Exercise Number : 3.23 of the Book
       Name : Cryptography and Network Security by
       Behrouz Forouzan, Special Indian Edition, 2007
```

```
2 // This file must be used under the terms of the
    CeCILL.
3 // This source file is licensed as described in the
    file COPYING, which
4 // you should have received as part of this
    distribution.  The terms
5 // are also available at
6 // http://www.cecill.info/licences/Licence_CeCILL_V2
    -en.txt
7 //This Source file is Written by Anjali Dinesh(16
    BCE1135) Guided by Dr. T. Subbulakshmi, Professor
8 //School of Computing Science and Engineering, VIT
    University Chennai
9 //This code has been referenced and derived from
    Scilab Textbook Companion for Cryptography and
    Network Security by A. Kahate Created by Akash
    Goel B.Tech. Comput
10 //er Engineering Delhi Technological University and
    Cross-Checked by Spandana on May 27, 2016. Funded
     by a grant from the National Mission on
    Education through IC
11 //T, Govt of India.
12 //The Operating System used for writing the code
    found in this file is Windows 8
13 //SCILAB version 5.5.2
14 //
    |-------------------------------------------------------------

15 //|This worked out example found in Page No: 69 of
    the book will do the following operations
                                |
16 //|1. Railfence|
17 clc;
18 clear;
19 //func to remove spaces from a string
20 function [mat]=remove_spaces(str)
21     mat=[]
22     k=1
```

```
23      for i=1:length(str)
24          if ~isletter(part(str,i:i)) then
25              continue
26          end
27          mat(k,1) = part(str,i:i)
28          k=k+1
29      end
30      mat = strcat(mat)
31  endfunction
32  function [mat]=message_rectangle(str,col)
33      l = length(str)
34      row = l/6
35      if modulo(l,6)>0 then
36          row=row+1
37      end
38      //remove whitespace and non-alphabets from
          string
39      str = remove_spaces(str)
40      //Conversion of plaintext into a message table
41      mat = []
42      k=1
43      for i=1:row
44          for j=1:col
45              if k>l then
46                  break
47              end
48              mat(i,j) = part(str,k:k)
49              k=k+1
50          end
51      end
52  endfunction
53
54
55  disp("Original plaintext message:")
56  pt = "meet me at the park"
57  disp(pt)
58  disp("")
59
```

```
60 //function from dependency file
61 pt = remove_spaces(pt)
62
63 l = length(pt)
64
65 col = 4
66
67 row = 5
68
69 //Conversion of plaintext into a message table
70 //function from dependency file
71 pt_mat = message_rectangle(pt,col)
72
73 disp("Plaintext message rectangle:")
74 printf("\n")
75 for i=1:col
76     printf(" %d ",i)
77 end
78 disp(pt_mat)
79 disp("")
80
81 //Column read order
82 col_order = [1 2 3 4]
83 disp("Column order: ")
84 disp(col_order)
85 disp("")
86 k=1
87
88 ct=[]
89 //Convert to ciphertext
90 for n = 1:length(col_order)
91     j = col_order(n)
92     for i=1:row
93         pos = (i-1)*col + j
94         if pos>l then
95             continue
96         end
97
```

```
98
99      end
100 end
101 disp("Ciphertext:")
102 ct = strcat(pt_mat)
103 disp(ct)
```

Scilab code Exa 3.24 Permutation of Each character in plaintext into the ciphertex

```
1  // Chapter No :3  Exercise Number : 3.24 of the Book
       Name : Cryptography and Network Security by
     Behrouz Forouzan, Special Indian Edition, 2007
2  // This file must be used under the terms of the
     CeCILL.
3  // This source file is licensed as described in the
     file COPYING, which
4  // you should have received as part of this
     distribution.  The terms
5  // are also available at
6  // http://www.cecill.info/licences/Licence_CeCILL_V2
     -en.txt
7  //This Source file is Written by Anjali Dinesh(16
     BCE1135) Guided by Dr. T. Subbulakshmi, Professor
8  //School of Computing Science and Engineering, VIT
     University Chennai
9  //This code has been referenced and derived from
     Scilab Textbook Companion for Cryptography and
     Network Security by A. Kahate Created by Akash
     Goel B.Tech. Comput
10 //er Engineering Delhi Technological University and
     Cross-Checked by Spandana on May 27, 2016. Funded
      by a grant from the National Mission on
     Education through IC
11 //T, Govt of India.
12 //The Operating System used for writing the code
```

```
    found in this file is Windows 8
13 //SCILAB version 5.5.2
14 //
   |-------------------------------------------------------------------------
15 //|This worked out example found in Page No: 50 of
    the book will do the following operations
                                |
16 //|| TRANPOSITION
17 clc;
18 clear;
19 //func to remove spaces from a string
20 function [mat]=remove_spaces(str)
21     mat=[]
22     k=1
23     for i=1:length(str)
24         if ~isletter(part(str,i:i)) then
25             continue
26         end
27         mat(k,1) = part(str,i:i)
28         k=k+1
29     end
30     mat = strcat(mat)
31 endfunction
32 function [mat]=message_rectangle(str,col)
33     l = length(str)
34     row = l/6
35     if modulo(l,6)>0 then
36         row=row+1
37     end
38     //remove whitespace and non-alphabets from
        string
39     str = remove_spaces(str)
40     //Conversion of plaintext into a message table
41     mat = []
42     k=1
43     for i=1:row
44         for j=1:col
```

```
45              if k>l then
46                  break
47              end
48              mat(i,j) = part(str,k:k)
49              k=k+1
50          end
51      end
52  endfunction
53
54
55  disp("Original plaintext message:")
56  pt = "meet me at the park"
57  disp(pt)
58  disp("")
59
60  //function from dependency file
61  pt = remove_spaces(pt)
62
63  l = length(pt)
64
65  col = 4
66
67  row = 5
68
69  //Conversion of plaintext into a message table
70  //function from dependency file
71  pt_mat = message_rectangle(pt,col)
72
73  disp("Plaintext message rectangle:")
74  printf("\n")
75  for i=1:col
76      printf(" %d ",i)
77  end
78  disp(pt_mat)
79  disp("")
80
81  //Column read order
82  col_order = [1 2 3 4]
```

```
disp("Column order: ")
disp(col_order)
disp("")
k=1

ct=[]
//Convert to ciphertext
for n = 1:length(col_order)
    j = col_order(n)
    for i=1:row
        pos = (i-1)*col + j
        if pos>l then
            continue
        end


    end
end
disp("Ciphertext:")
ct = strcat(pt_mat)
disp(ct)
```

Scilab code Exa 3.27 Use matrices to show the encryption decryption processes for

```
// Chapter No :3  Exercise Number : 3.27 of the Book
    Name : Cryptography and Network Security by
   Behrouz Forouzan, Special Indian Edition, 2007
// This file must be used under the terms of the
   CeCILL.
// This source file is licensed as described in the
   file COPYING, which
// you should have received as part of this
   distribution.  The terms
// are also available at
// http://www.cecill.info/licences/Licence_CeCILL_V2
```

```
    -en.txt
7 //This Source file is Written by Anjali Dinesh(16
    BCE1135) Guided by Dr. T. Subbulakshmi, Professor
8 //School of Computing Science and Engineering, VIT
    University Chennai
9 //The Operating System used for writing the code
    found in this file is Windows 8
10 //SCILAB version 5.5.2
11 //
    |-----------------------------------------------------------------

12 //|This worked out example found in Page No: 72 of
    the book will do the following operations
                                |
13 //| Check cipher text matrix accuracy|
14 clc;
15 clear;
16 enkey=[]
17 ciphtxt=[]
18 cphtxt=[]
19 pltxt=[]
20 enkey=[0 1 0 0 0 ;0 0 0 0 1; 1 0 0 0 0; 0 0 1 0 0 ;
    0 0 0 1 0]
21 ciphtxt=[04 04 12 24 13; 19 00 00 02 19;19 10 14 13
    18; 07 08 19 25 06]
22 pltxt=[04 13 04 1 24;00 19 19 00 02;10 1 19 14 13;
    08 06 07 19 25]
23
24
25 disp("Cipher text")
26 disp(ciphtxt)
```

Scilab code Exa 3.30 Additive ciphers categorized as stream ciphers

```
1
```

```
2 // Chapter No : 3  Exercise Number : 3.30 of the
    Book Name : Cryptography and Network Security by
    Behrouz Forouzan, Special Indian Edition, 2007
3 // This file must be used under the terms of the
    CeCILL.
4 // This source file is licensed as described in the
    file COPYING, which
5 // you should have received as part of this
    distribution.  The terms
6 // are also available at
7 // http://www.cecill.info/licences/Licence_CeCILL_V2
    -en.txt
8 //This Source file is Written by Anjali Dinesh(16
    BCE1135) Guided by Dr. T. Subbulakshmi, Professor
9 //School of Computing Science and Engineering, VIT
    University Chennai
10 //The Operating System used for writing the code
    found in this file is Windows 8
11 //SCILAB version 5.5.2
12 //
    |-------------------------------------------------------------------

13 //|This worked out example found in Page No: 75 of
    the book will do the following operations
                                |
14 //|1. Show facts|
15 clc;
16 clear;
17 printf("Additive ciphers are categorized as stream
    ciphers in which the key stream is the repeated
    value of the key. In this cipher, however, Each
    character in the ciphertext belongs only to the
    corresponding character in the plaintext.")
```

Scilab code Exa 3.31 Mono alphabetic substitution ciphers

```
1
2 // Chapter No : 3  Exercise Number : 3.31 of the
    Book Name : Cryptography and Network Security by
    Behrouz Forouzan, Special Indian Edition, 2007
3 // This file must be used under the terms of the
    CeCILL.
4 // This source file is licensed as described in the
    file COPYING, which
5 // you should have received as part of this
    distribution.  The terms
6 // are also available at
7 // http://www.cecill.info/licences/Licence_CeCILL_V2
    -en.txt
8 //This Source file is Written by Anjali Dinesh(16
    BCE1135) Guided by Dr. T. Subbulakshmi, Professor
9 //School of Computing Science and Engineering, VIT
    University Chennai
10 //The Operating System used for writing the code
    found in this file is Windows 8
11 //SCILAB version 5.5.2
12 //
    |-------------------------------------------------------------------------

13 //|This worked out example found in Page No: 75 of
    the book will do the following operations
                                |
14 //|1. Show facts|
15 clc;
16 clear;
17 printf("The monoalphabetic sustitution ciphers
    discussed in this chapter are also stream ciphers
    . However each character is mapped in the mapping
     table")
```

Scilab code Exa 3.32 Vigenere Ciphers are also stream ciphers

```
1
2 // Chapter No : 3  Exercise Number : 3.32 of the
    Book Name : Cryptography and Network Security by
    Behrouz Forouzan, Special Indian Edition, 2007
3 // This file must be used under the terms of the
    CeCILL.
4 // This source file is licensed as described in the
    file COPYING, which
5 // you should have received as part of this
    distribution.  The terms
6 // are also available at
7 // http://www.cecill.info/licences/Licence_CeCILL_V2
    -en.txt
8 //This Source file is Written by Anjali Dinesh(16
    BCE1135) Guided by Dr. T. Subbulakshmi, Professor
9 //School of Computing Science and Engineering, VIT
    University Chennai
10 //The Operating System used for writing the code
    found in this file is Windows 8
11 //SCILAB version 5.5.2
12 //
    |-------------------------------------------------------------------------

13 //|This worked out example found in Page No: 75 of
    the book will do the following operations
                                |
14 //|1. Show facts|
15 clc;
16 clear;
17 printf("Vignere ciphers are also stream ciphers . In
     this case, the ey stream s the repetton of M
    values where M is the size of the Keyword")
```

Scilab code Exa 3.33 divide stream ciphers base on key streams

```
1
2 // Chapter No : 3  Exercise Number : 3.33 of the
    Book Name : Cryptography and Network Security by
    Behrouz Forouzan, Special Indian Edition, 2007
3 // This file must be used under the terms of the
    CeCILL.
4 // This source file is licensed as described in the
    file COPYING, which
5 // you should have received as part of this
    distribution.  The terms
6 // are also available at
7 // http://www.cecill.info/licences/Licence_CeCILL_V2
    -en.txt
8 //This Source file is Written by Anjali Dinesh(16
    BCE1135) Guided by Dr. T. Subbulakshmi, Professor
9 //School of Computing Science and Engineering, VIT
    University Chennai
10 //The Operating System used for writing the code
    found in this file is Windows 8
11 //SCILAB version 5.5.2
12 //
    |-------------------------------------------------------------------------

13 //|This worked out example found in Page No: 75 of
    the book will do the following operations
                                |
14 //|1. Show facts|
15 clc;
16 clear;
17 printf("1. additive ciphers are monoalphabetic 2.
    Monoalphabetic ciphers are monoalphabetic 3.
    Vignere ciphers are polyalphabetic")
```

Scilab code Exa 3.34 Playfair ciphers are block ciphers

```
1
2 // Chapter No : 3  Exercise Number : 3.34 of the
    Book Name : Cryptography and Network Security by
    Behrouz Forouzan, Special Indian Edition, 2007
3 // This file must be used under the terms of the
    CeCILL.
4 // This source file is licensed as described in the
    file COPYING, which
5 // you should have received as part of this
    distribution.  The terms
6 // are also available at
7 // http://www.cecill.info/licences/Licence_CeCILL_V2
    -en.txt
8 //This Source file is Written by Anjali Dinesh(16
    BCE1135) Guided by Dr. T. Subbulakshmi, Professor
9 //School of Computing Science and Engineering, VIT
    University Chennai
10 //The Operating System used for writing the code
    found in this file is Windows 8
11 //SCILAB version 5.5.2
12 //
    |-------------------------------------------------------------------------

13 //|This worked out example found in Page No: 76 of
    the book will do the following operations
                                |
14 //|1. Show facts|
15 clc;
16 clear;
17 printf("Playfair ciphers are block ciphers with
    block size 2")
```

Scilab code Exa 3.35 `Hill ciphers are block ciphers`

```
1
```

```
2 // Chapter No : 3  Exercise Number : 3.35 of the
    Book Name : Cryptography and Network Security by
    Behrouz Forouzan, Special Indian Edition, 2007
3 // This file must be used under the terms of the
    CeCILL.
4 // This source file is licensed as described in the
    file COPYING, which
5 // you should have received as part of this
    distribution.  The terms
6 // are also available at
7 // http://www.cecill.info/licences/Licence_CeCILL_V2
    -en.txt
8 //This Source file is Written by Anjali Dinesh(16
    BCE1135) Guided by Dr. T. Subbulakshmi, Professor
9 //School of Computing Science and Engineering, VIT
    University Chennai
10 //The Operating System used for writing the code
    found in this file is Windows 8
11 //SCILAB version 5.5.2
12 //
   |-------------------------------------------------------------------

13 //|This worked out example found in Page No: 76 of
    the book will do the following operations
                                |
14 //|1. Show facts|
15 clc;
16 clear;
17 printf("Hill cipher are block ciphers")
```

Scilab code Exa 3.36 Every Block cipher is a polyalphabetic cipher

```
1
2 // Chapter No : 3  Exercise Number : 3.36 of the
    Book Name : Cryptography and Network Security by
```

```
    Behrouz Forouzan, Special Indian Edition, 2007
3 // This file must be used under the terms of the
    CeCILL.
4 // This source file is licensed as described in the
    file COPYING, which
5 // you should have received as part of this
    distribution.  The terms
6 // are also available at
7 // http://www.cecill.info/licences/Licence_CeCILL_V2
    -en.txt
8 //This Source file is Written by Anjali Dinesh(16
    BCE1135) Guided by Dr. T. Subbulakshmi, Professor
9 //School of Computing Science and Engineering, VIT
    University Chennai
10 //The Operating System used for writing the code
    found in this file is Windows 8
11 //SCILAB version 5.5.2
12 //
    |-------------------------------------------------------------------

13 //|This worked out example found in Page No: 76 of
    the book will do the following operations
                                    |
14 //|1. Show facts|
15 clc;
16 clear;
17 printf("A block cipher is polyalphabetic")
```

Chapter 6

Data Encryption Standard

Scilab code Exa 6.1 Find the output of the initial permutation box when the input

```
1 // Chapter No :6  Exercise Number : 6.1 of the Book
    Name : Cryptography and Network Security by
    Behrouz Forouzan, Special Indian Edition, 2007
2 // This file must be used under the terms of the
    CeCILL.
3 // This source file is licensed as described in the
    file COPYING, which
4 // you should have received as part of this
    distribution.  The terms
5 // are also available at
6 // http://www.cecill.info/licences/Licence_CeCILL_V2
    -en.txt
7 //This Source file is Written by Pranav
    Sreenivasarao 16BCE1361  Anamay Prateek (16
    BCCE1267)) Guided by Dr. T. Subbulakshmi,
    Professor
8 //School of Computing Science and Engineering, VIT
    University Chennai
9
10 //The Operating System used for writing the code
    found in this file is Windows 8
```

```
11 //SCILAB version 5.5.2
12
13
14 clc;
15 clear;
16
17 //
    "0000000000000010000000000000000000000000000000000000000000000001"
     as input
18 //p=input("Enter bin code of 64 characters : ")
19 p = "
    0000000000000010000000000000000000000000000000000000000000000001
    "
20 for i=1:length(p)
21     ah(i)=part(p,i:i)
22 end
23 disp(length(p))
24 q="
    0000000000000000000000000000000000000000000000000000000000000000
    "
25 for i=1:length(q)
26     qq(i)=part(q,i:i)
27 end
28 disp(length(p))
29 m=[
30 58 50 42 34 26 18 10 2;
31 60 52 44 36 28 20 12 4;
32 62 54 46 38 30 22 14 6;
33 64 56 48 40 32 24 16 8;
34 57 49 41 33 25 17 09 1;
35 59 51 43 35 27 19 11 3;
36 61 53 45 37 29 21 13 5;
37 63 55 47 39 31 23 15 7]
38
39 for k=1:64
40     for i=1:8
41         for j=1:8
42             if(k==m(i,j))
```

```
43                  qq((i-1)*8+j)=ah(k)
44              end
45          end
46      end
47  end
48
49  for i=1:length(q)
50      disp(string(i)+":"+qq(i))
51  end
```

Scilab code Exa 6.2 Prove that the initial and final permutation are inverse of ea

```
1  // Chapter No :6  Exercise Number : 6.2 of the Book
     Name : Cryptography and Network Security by
     Behrouz Forouzan, Special Indian Edition, 2007
2  // This file must be used under the terms of the
     CeCILL.
3  // This source file is licensed as described in the
     file COPYING, which
4  // you should have received as part of this
     distribution.  The terms
5  // are also available at
6  // http://www.cecill.info/licences/Licence_CeCILL_V2
     -en.txt
7  //This Source file is Written by Pranav
     Sreenivasarao 16BCE1361  Anamay Prateek (16
     BCCE1267)) Guided by Dr. T. Subbulakshmi,
     Professor
8  //School of Computing Science and Engineering, VIT
     University Chennai
9
10 //The Operating System used for writing the code
     found in this file is Windows 8
11 //SCILAB version 5.5.2
12
```

```
13
14 clc;
15 clear;
16 //
     "0000000000000000000000010000000000000000000000000000000010"
      as input
17 //p=input("Enter bin code of 64 characters : ")
18
19 clc;
20 clear;
21 p = "
     0000000000000000000000010000000000000000000000000000000010
     "
22 for i=1:length(p)
23     ah(i)=part(p,i:i)
24 end
25 disp(length(p))
26 q="
     0000000000000000000000000000000000000000000000000000000000
     "
27 for i=1:length(q)
28     qq(i)=part(q,i:i)
29 end
30 disp(length(p))
31 m=[
32 40 08 48 16 56 24 64 32;
33 39 07 47 15 55 23 63 31;
34 38 06 46 14 54 22 62 30;
35 37 05 45 13 53 21 61 29;
36 36 04 44 12 52 20 60 28;
37 35 03 43 11 51 19 59 27;
38 34 02 42 10 50 18 58 26;
39 33 01 41 09 49 17 57 25]
40
41 for k=1:64
42     for i=1:8
43         for j=1:8
44             if(k==m(i,j))
```

```
45                  qq((i-1)*8+j)=ah(k)
46              end
47          end
48      end
49  end
50
51  for i=1:length(q)
52      disp(string(i)+":"+qq(i))
53  end
```

Scilab code Exa 6.3 The input to the S box is 100011 and what is the output

```
1  // Chapter No :6  Exercise Number : 6.3 of the Book
       Name : Cryptography and Network Security by
       Behrouz Forouzan, Special Indian Edition, 2007
2  // This file must be used under the terms of the
       CeCILL.
3  // This source file is licensed as described in the
       file COPYING, which
4  // you should have received as part of this
       distribution.  The terms
5  // are also available at
6  // http://www.cecill.info/licences/Licence_CeCILL_V2
       -en.txt
7  //This Source file is Written by Pranav
       Sreenivasarao 16BCE1361  Anamay Prateek (16
       BCCE1267)) Guided by Dr. T. Subbulakshmi,
       Professor
8  //School of Computing Science and Engineering, VIT
       University Chennai
9
10 //The Operating System used for writing the code
       found in this file is Windows 8
11 //SCILAB version 5.5.2
12
```

```
13
14 clc;
15 clear;
16 //"100011" as input
17 p=input("Enter bin code of 6 characters : ")
18 for i=1:length(p)
19     ah(i)=part(p,i:i)
20 end
21 disp(length(p))
22 q="0000"
23 for i=1:length(q)
24     qq(i)=part(q,i:i)
25 end
26
27 row=(ascii(ah(1))-48)*2+(ascii(ah(6))-48)*1+1
28 col=(ascii(ah(2))-48)*8+(ascii(ah(3))-48)*4+(ascii(
     ah(4))-48)*2+(ascii(ah(5))-48)*1+1
29 m=[
30 14 04 13 01 02 15 11 08 03 10 06 12 05 09 00 07;
31 00 15 07 04 14 02 13 10 03 06 12 11 09 05 03 08;
32 04 01 14 08 13 06 02 11 15 12 09 07 03 10 05 00;
33 15 12 08 02 04 09 01 07 05 11 03 14 10 00 06 13]
34 num=m(row,col)
35 disp(num)
```

Scilab code Exa 6.4 The input to the S box 8 is 100011 and what is the output

```
1 // Chapter No :6  Exercise Number : 6.4 of the Book
    Name : Cryptography and Network Security by
    Behrouz Forouzan, Special Indian Edition, 2007
2 // This file must be used under the terms of the
    CeCILL.
3 // This source file is licensed as described in the
    file COPYING, which
4 // you should have received as part of this
```

```
    distribution.  The terms
5 // are also available at
6 // http://www.cecill.info/licences/Licence_CeCILL_V2
    -en.txt
7 //This Source file is Written by Pranav
    Sreenivasarao 16BCE1361  Anamay Prateek (16
    BCCE1267)) Guided by Dr. T. Subbulakshmi,
    Professor
8 //School of Computing Science and Engineering, VIT
    University Chennai
9
10 //The Operating System used for writing the code
    found in this file is Windows 8
11 //SCILAB version 5.5.2
12
13
14 clc;
15 clear;
16 //"100011" as input
17 p=input("Enter bin code of 6 characters : ")
18 for i=1:length(p)
19     ah(i)=part(p,i:i)
20 end
21 disp(length(p))
22 q="0000"
23 for i=1:length(q)
24     qq(i)=part(q,i:i)
25 end
26
27 row=(ascii(ah(1))-48)*2+(ascii(ah(6))-48)*1+1
28 col=(ascii(ah(2))-48)*8+(ascii(ah(3))-48)*4+(ascii(
    ah(4))-48)*2+(ascii(ah(5))-48)*1+1
29 m=[
30 13 02 08 04 06 15 11 01 10 09 03 14 05 00 10 07;
31 01 15 13 08 10 03 07 04 12 05 06 11 10 14 09 02;
32 07 11 04 01 09 12 14 02 00 06 10 10 15 03 05 08;
33 02 01 14 07 04 10 08 13 15 12 09 09 03 05 06 11]
34 num=m(row,col)
```

```
35 disp(num)
```

Scilab code Exa 6.5 Choose random plaintext block and a random key and determine w

```
1 // Chapter No :6  Exercise Number : 6.5 of the Book
    Name : Cryptography and Network Security by
    Behrouz Forouzan, Special Indian Edition, 2007
2 // This file must be used under the terms of the
    CeCILL.
3 // This source file is licensed as described in the
    file COPYING, which
4 // you should have received as part of this
    distribution.  The terms
5 // are also available at
6 // http://www.cecill.info/licences/Licence_CeCILL_V2
    -en.txt
7 //This Source file is Written by Pranav
    Sreenivasarao 16BCE1361  Anamay Prateek (16
    BCCE1267)) Guided by Dr. T. Subbulakshmi,
    Professor
8 //School of Computing Science and Engineering, VIT
    University Chennai
9
10 //The Operating System used for writing the code
    found in this file is Windows 8
11 //SCILAB version 5.5.2
12
13
14 clc;
15 clear;
16 function bin=hex2bin(hex)
17     bin=""
18     for i=1:length(hex)
19         temp=dec2bin(hex2dec(part(hex,i:i)),4)
20         bin=bin+temp
```

```
21      end
22  endfunction
23
24  function hex=bin2hex(bin)
25     hex=""
26     for i=1:length(bin)/4
27          temp=""
28          for j=1:4
29              temp=temp+part(bin,(i-1)*4+j:(i-1)*4+j)
30          end
31          hex=hex+dec2hex(bin2dec(temp))
32      end
33  endfunction
34
35  function lshft=shift(shft,no)
36      lshft=""
37      for n=1:no
38          temp=shft(1)
39          for o=2:28
40              shft(o-1)=shft(o)
41          end
42          shft(28)=temp
43      end
44      lshft=shft
45  endfunction
46
47  function [i_p,l,r,i_p_b,l_b,r_b]=ini_perm(plain)
48      p_bin=hex2bin(plain)
49      q="
           0000000000000000000000000000000000000000000000000000000000000000
           "
50      for i=1:64
51          gg(i)=part(q,i:i)
52      end
53      m=[
54      58 50 42 34 26 18 10 2;
55      60 52 44 36 28 20 12 4;
56      62 54 46 38 30 22 14 6;
```

```
    64 56 48 40 32 24 16 8;
    57 49 41 33 25 17 09 1;
    59 51 43 35 27 19 11 3;
    61 53 45 37 29 21 13 5;
    63 55 47 39 31 23 15 7]
    for k=1:64
        for i=1:8
            for j=1:8
                if(k==m(i,j))
                    gg((i-1)*8+j)=part(p_bin,k:k)
                end
            end
        end
    end
    i_p_b=""
    l_b=""
    r_b=""
    for i=1:length(q)
        i_p_b=i_p_b+gg(i)
        if i<=length(q)/2
            l_b=l_b+gg(i)
        else
            r_b=r_b+gg(i)
        end
    end
    r=bin2hex(r_b)
    i_p=bin2hex(i_p_b)
    l=bin2hex(l_b)
endfunction

function res=fin_perm(inp)
    l=[
    40 8 48 16 56 24 64 32;
    39 7 47 15 55 23 63 31;
    38 6 46 14 54 22 62 30;
    37 5 45 13 53 21 61 29;
    36 4 44 12 52 20 60 28;
    35 3 43 11 51 19 59 27;
```

```
95      34 2 42 10 50 18 58 26;
96      33 1 41 9 49 17 57 25]
97      for k=1:64
98          for i=1:8
99              for j=1:8
100                 if(k==l(i,j))
101                     ss((i-1)*8+j)=part(inp,k:k)
102                 end
103             end
104         end
105     end
106     res=""
107     for i=1:64
108         res=res+ss(i)
109     end
110 endfunction
111
112 function rk=genroundkey(lb,rb)
113     kct=[
114     14 17 11 24 01 05 03 28;
115     15 06 21 10 23 19 12 04;
116     26 08 16 07 27 20 13 02;
117     41 52 31 37 47 55 30 40;
118     51 45 33 48 44 49 39 56;
119     34 53 46 42 50 36 29 32]
120     for k=1:56
121         for i=1:6
122             for j=1:8
123                 if k==kct(i,j)
124                     if k<=28
125                         uu((i-1)*8+j)=lb(k)
126                     else
127                         uu((i-1)*8+j)=rb(k-28)
128                     end
129                 end
130             end
131         end
132     end
```

```
133     rk=""
134     for i=1:48
135         rk=rk+uu(i)
136     end
137     l=""
138     r=""
139     for i=1:28
140         l=l+lb(i)
141         r=r+rb(i)
142     end
143 endfunction
144
145 function ru=func(y,test)
146     sbox=[
147     14 4 13 1 2 15 11 8 3 10 6 12 5 9 0 7;
148     0 15 7 4 14 2 13 1 10 6 12 11 9 5 3 8;
149     4 1 14 8 13 6 2 11 15 12 9 7 3 10 5 0;
150     15 12 8 2 4 9 1 7 5 11 3 14 10 0 6 13;
151
152     15 1 8 14 6 11 3 4 9 7 2 13 12 0 5 10;
153     3 13 4 7 15 2 8 14 12 0 1 10 6 9 11 5;
154     0 14 7 11 10 4 13 1 5 8 12 6 9 3 2 15;
155     13 8 10 1 3 15 4 2 11 6 7 12 0 5 14 9;
156
157     10 0 9 14 6 3 15 5 1 13 12 7 11 4 2 8;
158     13 7 0 9 3 4 6 10 2 8 5 14 12 11 15 1;
159     13 6 4 9 8 15 3 0 11 1 2 12 5 10 14 7;
160     1 10 13 0 6 9 8 7 4 15 14 3 11 5 2 12;
161
162     7 13 14 3 0 6 9 10 1 2 8 5 11 12 4 15;
163     13 8 11 5 6 15 0 3 4 7 2 12 1 10 14 9;
164     10 6 9 0 12 11 7 13 15 1 3 14 5 2 8 4;
165     3 15 0 6 10 1 13 8 9 4 5 11 12 7 2 14;
166
167     2 12 4 1 7 10 11 6 8 5 3 15 13 0 14 9;
168     14 11 2 12 4 7 13 1 5 0 15 10 3 9 8 6;
169     4 2 1 11 10 13 7 8 15 9 12 5 6 3 0 14;
170     11 8 12 7 1 14 2 13 6 15 0 9 10 4 5 3;
```

```
171
172       12 1 10 15 9 2 6 8 0 13 3 4 14 7 5 11;
173       10 15 4 2 7 12 9 5 6 1 13 14 0 11 3 8;
174       9 14 15 5 2 8 12 3 7 0 4 10 1 13 11 6;
175       4 3 2 12 9 5 15 10 11 14 1 7 6 0 8 13;
176
177       4 11 2 14 15 0 8 13 3 12 9 7 5 10 6 1;
178       13 0 11 7 4 9 1 10 14 3 5 12 2 15 8 6;
179       1 4 11 13 12 3 7 14 10 15 6 8 0 5 9 2;
180       6 11 13 8 1 4 10 7 9 5 0 15 14 2 3 12;
181
182       13 2 8 4 6 15 11 1 10 9 3 14 5 0 12 7;
183       1 15 13 8 10 3 7 4 12 5 6 11 0 14 9 2;
184       7 11 4 1 9 12 14 2 0 6 10 13 15 3 5 8;
185       2 1 14 7 4 10 8 13 15 12 9 0 3 5 6 11]
186       spt=[
187       16 7 20 21 29 12 28 17;
188       1 15 23 26 5 18 31 10;
189       2 8 24 14 32 27 3 9;
190       19 13 30 6 22 11 4 25]
191       epbt=[
192       32 1 2 3 4 5;
193       4 5 6 7 8 9;
194       8 9 10 11 12 13;
195       12 13 14 15 16 17;
196       16 17 18 19 20 21;
197       20 21 22 23 24 25;
198       24 25 26 27 28 29;
199       28 29 30 31 32 1]
200       for k=1:32
201           for i=1:8
202               for j=1:6
203                   if k==epbt(i,j)
204                       ff((i-1)*6+j)=part(y,k:k)
205                   end
206               end
207           end
208       end
```

```
209     for i=1:48
210         if part(test,i:i)==ff(i)
211             nn(i)="0"
212         else
213             nn(i)="1"
214         end
215     end
216     temp=""
217     for i=1:8
218         row=bin2dec(nn((i-1)*6+1)+nn((i-1)*6+6))
219         col=bin2dec(nn((i-1)*6+2)+nn((i-1)*6+3)+nn((
              i-1)*6+4)+nn((i-1)*6+5))
220         temp=temp+dec2bin(sbox((i-1)*4+row+1,col+1)
              ,4)
221     end
222     for k=1:32
223         for i=1:4
224             for j=1:8
225                 if k==spt(i,j)
226                     dd((i-1)*8+j)=part(temp,k:k)
227                 end
228             end
229         end
230     end
231     ru=""
232     for i=1:32
233         ru=ru+dd(i)
234     end
235 endfunction
236
237 p="123456ABCD132536"
238 key="AABB09182736CCDD"
239 disp("Plain Text : "+p)
240 disp("Key : "+key)
241
242 keyb=hex2bin(key)
243 for i=1:length(keyb)
244     ah(i)=part(keyb,i:i)
```

```
end

q="
   0000000000000000000000000000000000000000000000000000000000000000
   "
for i=1:length(q)
    qq(i)=part(q,i:i)
end
pbdt=[
57 49 41 33 25 17 09 01;
58 50 42 34 26 18 10 02;
59 51 43 35 27 19 11 03;
60 52 44 36 63 55 47 39;
31 23 15 07 62 54 46 38;
30 22 14 06 61 53 45 37;
29 21 13 05 28 20 12 04]

for k=1:64
    for i=1:7
        for j=1:8
            if k==pbdt(i,j)
                qq((i-1)*8+j)=ah(k)
            end
        end
    end
end

i_p_b=""
l_b=""
r_b=""
for i=1:length(q)
    i_p_b=i_p_b+qq(i)
    if i<=length(q)/2
        l_b=l_b+qq(i)
    else
        r_b=r_b+qq(i)
    end
end
```

```
r=bin2hex(r_b)
i_p=bin2hex(i_p_b)
l=bin2hex(l_b)

[i,l,r,ib,lb,rb]=ini_perm(p)
disp("After initial permutation : "+i+"     "+ib)
disp("L0 = "+l+"     "+lb)
disp("R0 = "+r+"     "+rb)

lbn=qq(1:28)
rbn=qq(29:56)
disp("")
disp("Round          Left        Right          RoundKey
   ")
for z=1:16
    if z==1
        n=1
    elseif z==2
        n=1
    elseif z==9
        n=1
    elseif z==16
        n=1
    else
        n=2
    end
    lbn=shift(lbn,n)
    rbn=shift(rbn,n)
    rkn=genroundkey(lbn,rbn)
    lb_n=func(rb,rkn)
    lb=dec2bin(bitxor(bin2dec(lb_n),bin2dec(lb)),32)
    if z<16
        temp=rb
        rb=lb
        lb=temp
    end
    disp("Round"+string(z)+"      "+bin2hex(lb)+"
       "+bin2hex(rb)+"     "+bin2hex(rkn))
```

```
317 end
318 fp=fin_perm(lb+rb)
319 disp("")
320 disp("(L16+R16) : "+bin2hex(lb+rb)+"    "+lb+rb)
321 disp("Cipher Text Afer Final Permutation : "+bin2hex
    (fp))
```

Chapter 9

Mathematics of Cryptography

Scilab code Exa 9.2 List the Primes smaller than 10

```
1 // Chapter No : 9  Exercise Number : 9.2 of the Book
    Name : Cryptography and Network Security by
    Behrouz Forouzan, Special Indian Edition, 2007
2 // This file must be used under the terms of the
    CeCILL.
3 // This source file is licensed as described in the
    file COPYING, which
4 // you should have received as part of this
    distribution.  The terms
5 // are also available at
6 // http://www.cecill.info/licences/Licence_CeCILL_V2
    -en.txt
7 //This Source file is Written by Student Shreya
    Rajiv Somkuwar(15BCE1225), Student J Robin Raj(15
    BCE1325) Guided by Dr. T. Subbulakshmi, Professor
8 //School of Computing Science and Engineering, VIT
    University Chennai
9 //The Operating System used for writing the code
    found in this file is Windows 8
10 //SCILAB version 5.5.2
11 //
```

```
12 //|This worked out example found in Page No: 230 of
   the book will do the following operations
                                |
13 //|1. Find the all primes which are less than 10
14 //|2. Print the result in the scilab command line
15 clc;
16 clear;
17 n=10
18 for i=2:n
19     error=0
20     for j=2:i-1
21         if(pmodulo(i,j)==0)
22             error=1
23             break
24         end
25
26     end
27     if(error==0)
28         disp(i)
29     end
30 end
```

Scilab code Exa 9.3 Three primes greater than 17

```
1 // Chapter No : 9  Exercise Number : 9.3 of the Book
   Name : Cryptography and Network Security by
   Behrouz Forouzan, Special Indian Edition, 2007
2 // This file must be used under the terms of the
   CeCILL.
3 // This source file is licensed as described in the
   file COPYING, which
4 // you should have received as part of this
   distribution.  The terms
```

```
5 // are also available at
6 // http://www.cecill.info/licences/Licence_CeCILL_V2
    -en.txt
7 //This Source file is Written by Student Shreya
    Rajiv Somkuwar(15BCE1225), Student J Robin Raj(15
    BCE1325) Guided by Dr. T. Subbulakshmi, Professor
8 //School of Computing Science and Engineering, VIT
    University Chennai
9 //The Operating System used for writing the code
    found in this file is Windows 8
10 //SCILAB version 5.5.2
11 //
   |-------------------------------------------------------------------------

12 //|This worked out example found in Page No: 230 of
    the book will do the following operations
                                  |
13 //|1. Prove that there exists infinite number of
    primes
14 //|2. Print the result in the scilab command line
15 clc;
16 clear;
17 n=510511
18 disp(19)
19 disp(97)
20 disp(277)
```

Scilab code Exa 9.4 Find the number of primes less than 1000000

```
1 // Chapter No : 9  Exercise Number : 9.4 of the Book
    Name : Cryptography and Network Security by
    Behrouz Forouzan, Special Indian Edition, 2007
2 // This file must be used under the terms of the
    CeCILL.
3 // This source file is licensed as described in the
```

```
    file COPYING, which
4 // you should have received as part of this
    distribution.  The terms
5 // are also available at
6 // http://www.cecill.info/licences/Licence_CeCILL_V2
    -en.txt
7 //This Source file is Written by Student Shreya
    Rajiv Somkuwar(15BCE1225), Student J Robin Raj(15
    BCE1325) Guided by Dr. T. Subbulakshmi, Professor
8 //School of Computing Science and Engineering, VIT
    University Chennai
9 //The Operating System used for writing the code
    found in this file is Windows 8
10 //SCILAB version 5.5.2
11 //
    |-------------------------------------------------------------------------

12 //||This worked out example found in Page No: 231 of
    the book will do the following operations
                                    |
13 //||1. Find the number of primes less than 1,000,000
14 //||2. Print the result in the scilab command line
15 clc;
16 clear;
17 n=1000000
18 x=n/log(n)
19 y=n/(log(n)-1.08366)
20 disp(int32(x))
21 disp(int32(y))
```

Scilab code Exa 9.5 is 97 a prime

```
1 // Chapter No : 9  Exercise Number : 9.5 of the Book
    Name : Cryptography and Network Security by
    Behrouz Forouzan, Special Indian Edition, 2007
```

```
2 // This file must be used under the terms of the
    CeCILL.
3 // This source file is licensed as described in the
    file COPYING, which
4 // you should have received as part of this
    distribution.  The terms
5 // are also available at
6 // http://www.cecill.info/licences/Licence_CeCILL_V2
    -en.txt
7 //This Source file is Written by Student Shreya
    Rajiv Somkuwar(15BCE1225), Student J Robin Raj(15
    BCE1325) Guided by Dr. T. Subbulakshmi, Professor
8 //School of Computing Science and Engineering, VIT
    University Chennai
9 //The Operating System used for writing the code
    found in this file is Windows 8
10 //SCILAB version 5.5.2
11 //
   |-------------------------------------------------------------------

12 //|This worked out example found in Page No: 231 of
    the book will do the following operations
                                  |
13 //|1. Check whether 97 is a prime or not
14 //|2. Print the result in the scilab command line
15 clc;
16 clear;
17 n=97
18 error=0
19 for i=2:sqrt(n)
20     if(pmodulo(n,i)==0)
21         error=1
22         break
23     end
24 end
25
26 if(error==0)
27     disp("Yes Prime!")
```

```
28 else
29     disp("No not a Prime!")
30 end
```

Scilab code Exa 9.6 is 301 a prime

```
1 // Chapter No : 9  Exercise Number : 9.6 of the Book
     Name : Cryptography and Network Security by
    Behrouz Forouzan, Special Indian Edition, 2007
2 // This file must be used under the terms of the
    CeCILL.
3 // This source file is licensed as described in the
    file COPYING, which
4 // you should have received as part of this
    distribution.  The terms
5 // are also available at
6 // http://www.cecill.info/licences/Licence_CeCILL_V2
    -en.txt
7 //This Source file is Written by Student Shreya
    Rajiv Somkuwar(15BCE1225), Student J Robin Raj(15
    BCE1325) Guided by Dr. T. Subbulakshmi, Professor
8 //School of Computing Science and Engineering, VIT
    University Chennai
9 //The Operating System used for writing the code
    found in this file is Windows 8
10 //SCILAB version 5.5.2
11 //
    |-------------------------------------------------------------------

12 //|This worked out example found in Page No: 231 of
    the book will do the following operations
                                  |
13 //|1. Check whether 301 is a prime or not
14 //|2. Print the result in the scilab command line
15 clc;
```

```
clear;
n=301
error=0
for i=2:sqrt(n)
    if(pmodulo(n,i)==0)
        error=1
        break
    end
end

if(error==0)
    disp("Yes Prime!")
else
    disp("No not a Prime!")
end
```

Scilab code Exa 9.7 Find the Eulers phi function value for 13

```
// Chapter No : 9  Exercise Number : 9.7 of the Book Name : Cryptography and Network Security by Behrouz Forouzan, Special Indian Edition, 2007
// This file must be used under the terms of the CeCILL.
// This source file is licensed as described in the file COPYING, which
// you should have received as part of this distribution.  The terms
// are also available at
// http://www.cecill.info/licences/Licence_CeCILL_V2-en.txt
//This Source file is Written by Student Shreya Rajiv Somkuwar(15BCE1225), Student J Robin Raj(15BCE1325) Guided by Dr. T. Subbulakshmi, Professor
//School of Computing Science and Engineering, VIT University Chennai
```

```
9  //The Operating System used for writing the code
     found in this file is Windows 8
10 //SCILAB version 5.5.2
11 //
     |-----------------------------------------------------------------

12 //|This worked out example found in Page No: 232 of
     the book will do the following operations
                                    |
13 //|1. Find the Euler's phi-function value for 13
14 //|2. Print the result in the scilab command line
15 clc;
16 clear;
17 function result=phi(n,e)
18     x=factor(n)
19     if(n==1)
20         result=0
21     elseif(x==n)
22         result=(n^e)-(n^(e-1))
23     else
24        [a,b]=unique(x)
25        for i=1:length(a)
26            if(i<length(a))
27                result(i)=phi(a(i),b(i+1)-b(i))
28            else
29                result(i)=phi(a(i),length(x)+1-b(i))
30            end
31        end
32     end
33 endfunction
34
35 n=13
36 result=phi(n,1)
37
38 ans=1
39 for i=1:length(result)
40     ans=ans*result(i)
41 end
```

```
42
43 disp(ans)
```

Scilab code Exa 9.8 Find the Eulers phi function value for 10

```
1 // Chapter No : 9  Exercise Number : 9.8 of the Book
     Name : Cryptography and Network Security by
    Behrouz Forouzan, Special Indian Edition, 2007
2 // This file must be used under the terms of the
    CeCILL.
3 // This source file is licensed as described in the
    file COPYING, which
4 // you should have received as part of this
    distribution.  The terms
5 // are also available at
6 // http://www.cecill.info/licences/Licence_CeCILL_V2
    -en.txt
7 //This Source file is Written by Student Shreya
    Rajiv Somkuwar(15BCE1225), Student J Robin Raj(15
    BCE1325) Guided by Dr. T. Subbulakshmi, Professor
8 //School of Computing Science and Engineering, VIT
    University Chennai
9 //The Operating System used for writing the code
    found in this file is Windows 8
10 //SCILAB version 5.5.2
11 //
    |-------------------------------------------------------------------------

12 //|This worked out example found in Page No: 232 of
    the book will do the following operations
                                  |
13 //|1. Find the Euler's phi-function value for 10
14 //|2. Print the result in the scilab command line
15
16 clc;
```

```
clear;
function result=phi(n,e)
    x=factor(n)
    if(n==1)
        result=0
    elseif(x==n)
        result=(n^e)-(n^(e-1))
    else
       [a,b]=unique(x)
       for i=1:length(a)
           if(i<length(a))
               result(i)=phi(a(i),b(i+1)-b(i))
           else
               result(i)=phi(a(i),length(x)+1-b(i))
           end
       end
    end
endfunction

n=10
result=phi(n,1)

ans=1
for i=1:length(result)
    ans=ans*result(i)
end

disp(ans)
```

Scilab code Exa 9.9 Find the Eulers phi function for 240

```
// Chapter No : 9  Exercise Number : 9.9 of the Book
    Name : Cryptography and Network Security by
   Behrouz Forouzan, Special Indian Edition, 2007
// This file must be used under the terms of the
```

```
    CeCILL.
3 // This source file is licensed as described in the
    file COPYING, which
4 // you should have received as part of this
    distribution.  The terms
5 // are also available at
6 // http://www.cecill.info/licences/Licence_CeCILL_V2
    -en.txt
7 //This Source file is Written by Student Shreya
    Rajiv Somkuwar(15BCE1225), Student J Robin Raj(15
    BCE1325) Guided by Dr. T. Subbulakshmi, Professor
8 //School of Computing Science and Engineering, VIT
    University Chennai
9 //The Operating System used for writing the code
    found in this file is Windows 8
10 //SCILAB version 5.5.2
11 //
    |-------------------------------------------------------------------

12 //|This worked out example found in Page No: 232 of
    the book will do the following operations
                                  |
13 //|1. Find the Euler's phi-function for 240
14 //|2. Print the result in the scilab command line
15
16 clc;
17 clear;
18 function result=phi(n,e)
19     x=factor(n)
20     if(n==1)
21         result=0
22     elseif(x==n)
23         result=(n^e)-(n^(e-1))
24     else
25        [a,b]=unique(x)
26        for i=1:length(a)
27            if(i<length(a))
28                result(i)=phi(a(i),b(i+1)-b(i))
```

```
29              else
30                  result(i)=phi(a(i),length(x)+1-b(i))
31              end
32          end
33      end
34  endfunction
35
36  n=240
37  result=phi(n,1)
38
39  ans=1
40  for i=1:length(result)
41      ans=ans*result(i)
42  end
43
44  disp(ans)
```

Scilab code Exa 9.10 Check the given derivation

```
1  // Chapter No : 9  Exercise Number : 9.10 of the
       Book Name : Cryptography and Network Security by
       Behrouz Forouzan, Special Indian Edition, 2007
2  // This file must be used under the terms of the
       CeCILL.
3  // This source file is licensed as described in the
       file COPYING, which
4  // you should have received as part of this
       distribution.  The terms
5  // are also available at
6  // http://www.cecill.info/licences/Licence_CeCILL_V2
       -en.txt
7  //This Source file is Written by Student Shreya
       Rajiv Somkuwar(15BCE1225), Student J Robin Raj(15
       BCE1325) Guided by Dr. T. Subbulakshmi, Professor
8  //School of Computing Science and Engineering, VIT
```

```
    University Chennai
//The Operating System used for writing the code
    found in this file is Windows 8
//SCILAB version 5.5.2
//
  |-------------------------------------------------------------------------

//|This worked out example found in Page No: 232 of
    the book will do the following operations
                                  |
//|1. Check whether   (49) =   (7) *    (7) = 6*6 =
    36.
//|2. Print the result in the scilab command line

clc;
clear;
function result=phi(n,e)
    x=factor(n)
    if(n==1)
        result=0
    elseif(x==n)
        result=(n^e)-(n^(e-1))
    else
       [a,b]=unique(x)
       for i=1:length(a)
           if(i<length(a))
               result(i)=phi(a(i),b(i+1)-b(i))
           else
               result(i)=phi(a(i),length(x)+1-b(i))
           end
       end
    end
endfunction

n=49
expectedAns=36
result=phi(n,1)

```

```
40 ans=1
41 for i=1:length(result)
42     ans=ans*result(i)
43 end
44
45 if(expectedAns ~= ans)
46     printf("No, the answer is %d",ans)
47 else
48     printf("Yes")
49 end
```

Scilab code Exa 9.11 Find the number of elements in Z of 14

```
1 // Chapter No : 9  Exercise Number : 9.11 of the
    Book Name : Cryptography and Network Security by
    Behrouz Forouzan, Special Indian Edition, 2007
2 // This file must be used under the terms of the
    CeCILL.
3 // This source file is licensed as described in the
    file COPYING, which
4 // you should have received as part of this
    distribution.  The terms
5 // are also available at
6 // http://www.cecill.info/licences/Licence_CeCILL_V2
    -en.txt
7 //This Source file is Written by Student Shreya
    Rajiv Somkuwar(15BCE1225), Student J Robin Raj(15
    BCE1325) Guided by Dr. T. Subbulakshmi, Professor
8 //School of Computing Science and Engineering, VIT
    University Chennai
9 //The Operating System used for writing the code
    found in this file is Windows 8
10 //SCILAB version 5.5.2
11 //
    |----------------------------------------------------------------
```

```
12 //|This worked out example found in Page No: 233 of
      the book will do the following operations
                                  |
13 //|1. Find the number of elements in Z(14)*
14 //|2. Print the result in the scilab command line
15
16 clc;
17 clear;
18 function result=phi(n,e)
19     x=factor(n)
20     if(n==1)
21         result=0
22     elseif(x==n)
23         result=(n^e)-(n^(e-1))
24     else
25        [a,b]=unique(x)
26        for i=1:length(a)
27            if(i<length(a))
28                result(i)=phi(a(i),b(i+1)-b(i))
29            else
30                result(i)=phi(a(i),length(x)+1-b(i))
31            end
32        end
33     end
34 endfunction
35
36 n=14
37 result=phi(n,1)
38
39 ans=1
40 for i=1:length(result)
41     ans=ans*result(i)
42 end
43
44 disp(ans)
```

Scilab code Exa 9.12 Find the result of 6 Pow 10 and mod 11

```
1 // Chapter No : 9  Exercise Number : 9.12 of the
    Book Name : Cryptography and Network Security by
    Behrouz Forouzan, Special Indian Edition, 2007
2 // This file must be used under the terms of the
    CeCILL.
3 // This source file is licensed as described in the
    file COPYING, which
4 // you should have received as part of this
    distribution.  The terms
5 // are also available at
6 // http://www.cecill.info/licences/Licence_CeCILL_V2
    -en.txt
7 //This Source file is Written by Student Shreya
    Rajiv Somkuwar(15BCE1225), Student J Robin Raj(15
    BCE1325) Guided by Dr. T. Subbulakshmi, Professor
8 //School of Computing Science and Engineering, VIT
    University Chennai
9 //The Operating System used for writing the code
    found in this file is Windows 8
10 //SCILAB version 5.5.2
11 //
   |-------------------------------------------------------------------

12 //|This worked out example found in Page No: 233 of
    the book will do the following operations
                                |
13 //|1. Find the result of 6^10 mod 11
14 //|2. Print the result in the scilab command line
15 clc;
16 clear;
17 a=6
18 b=10
```

```
19 n=11
20 //a^b mod n
21 if b==n then
22     p=b
23     x=factor(p)
24     if size(x,"*")==1 then
25         disp(a)
26     end
27 elseif b==n-1 then
28     p=b+1
29     y=factor(p)
30     if size(y,"*")==1 & pmodulo(a,p)~=0 then
31         disp(1)
32     end
33 end
```

Scilab code Exa 9.13 Find the result of 3 Pow 12 and mod 11

```
1 // Chapter No : 9  Exercise Number : 9.13 of the
    Book Name : Cryptography and Network Security by
    Behrouz Forouzan, Special Indian Edition, 2007
2 // This file must be used under the terms of the
    CeCILL.
3 // This source file is licensed as described in the
    file COPYING, which
4 // you should have received as part of this
    distribution.  The terms
5 // are also available at
6 // http://www.cecill.info/licences/Licence_CeCILL_V2
    -en.txt
7 //This Source file is Written by Student Shreya
    Rajiv Somkuwar(15BCE1225), Student J Robin Raj(15
    BCE1325) Guided by Dr. T. Subbulakshmi, Professor
8 //School of Computing Science and Engineering, VIT
    University Chennai
```

```
9  //The Operating System used for writing the code
     found in this file is Windows 8
10 //SCILAB version 5.5.2
11 //
     |-----------------------------------------------------------------------

12 //|This worked out example found in Page No: 233 of
     the book will do the following operations
                                    |
13 //|1. Find theresult of 3^12 mod 11
14 //|2. Print the result in the scilab command line
15 clc;
16 clear;
17 a=3
18 b=12
19 n=11
20 //a^b mod n
21
22 b2=b-n
23 b=b-b2
24
25
26 if b==n then
27     p=b
28     x=factor(p)
29     if size(x,"*")==1 then
30         ans=a
31     end
32 elseif b==n-1 then
33     p=b+1
34     y=factor(p)
35     if size(y,"*")==1 & pmodulo(a,p)~=0 then
36         ans=1
37     end
38 end
39
40 disp(ans*pmodulo(a^b2,n))
```

Scilab code Exa 9.14 Multiplicative inverse modulo a prime number can be found wit

```
1 // Chapter No : 9  Exercise Number : 9.14 of the
    Book Name : Cryptography and Network Security by
    Behrouz Forouzan, Special Indian Edition, 2007
2 // This file must be used under the terms of the
    CeCILL.
3 // This source file is licensed as described in the
    file COPYING, which
4 // you should have received as part of this
    distribution.  The terms
5 // are also available at
6 // http://www.cecill.info/licences/Licence_CeCILL_V2
    -en.txt
7 //This Source file is Written by Student Shreya
    Rajiv Somkuwar(15BCE1225), Student J Robin Raj(15
    BCE1325) Guided by Dr. T. Subbulakshmi, Professor
8 //School of Computing Science and Engineering, VIT
    University Chennai
9 //The Operating System used for writing the code
    found in this file is Windows 8
10 //SCILAB version 5.5.2
11 //
    |-------------------------------------------------------------

12 //|This worked out example found in Page No: 234 of
    the book will do the following operations
                                |
13 //|1. Find the multiplicative inverse of 8 in Z(17),
     5 in Z(23), 60 in Z(101) and 22 in Z(211)
    without using extended euclid's algorithm
14 //|2. Print the result in the scilab command line
15 clc;
16 clear;
```

```
17 function result = fermatMI(a,n)
18     x=factor(n)
19     if size(x,"*") == 1 then
20         t=n-2
21         result=pmodulo(a^t,n)
22     else
23         disp("Multiplicative inverse can not be
               found for non prime modulus!!")
24     end
25 endfunction
26
27 //(a) Multiplicative Inverse of 8 modulus 17
28 a=8
29 n=17
30 disp(fermatMI(a,n))
31
32 //(b) Multiplicative Inverse of 5 modulus 23
33 a=5
34 n=23
35 disp(fermatMI(a,n))
36
37 //(c) Multiplicative Inverse of 60 modulus 101
38 a=60
39 n=101
40 disp("To Big Calculation!")
41
42 //(d) Multiplicative Inverse of 22 modulus 211
43 a=22
44 n=211
45 disp("To Big Calculation!")
```

Scilab code Exa 9.15 Find the result of 6 pow 24 mod 35

```
1 // Chapter No : 9  Exercise Number : 9.15 of the
    Book Name : Cryptography and Network Security by
```

```
1  Behrouz Forouzan, Special Indian Edition, 2007
2 // This file must be used under the terms of the
     CeCILL.
3 // This source file is licensed as described in the
     file COPYING, which
4 // you should have received as part of this
     distribution.  The terms
5 // are also available at
6 // http://www.cecill.info/licences/Licence_CeCILL_V2
     -en.txt
7 //This Source file is Written by Student Shreya
     Rajiv Somkuwar(15BCE1225), Student J Robin Raj(15
     BCE1325) Guided by Dr. T. Subbulakshmi, Professor
8 //School of Computing Science and Engineering, VIT
     University Chennai
9 //The Operating System used for writing the code
     found in this file is Windows 8
10 //SCILAB version 5.5.2
11 //
     |-------------------------------------------------------------------------

12 //|This worked out example found in Page No: 234 of
     the book will do the following operations
                                  |
13 //|1. Find the result of 6^24 mod 35
14 //|2. Print the result in the scilab command line
15
16 clc;
17 clear;
18 function result=phi(n,e)
19     x=factor(n)
20     if(n==1)
21         result=0
22     elseif(x==n)
23         result=(n^e)-(n^(e-1))
24     else
25        [a,b]=unique(x)
26        for i=1:length(a)
```

```
27              if(i<length(a))
28                  result(i)=phi(a(i),b(i+1)-b(i))
29              else
30                  result(i)=phi(a(i),length(x)+1-b(i))
31              end
32          end
33      end
34  endfunction
35
36  a = 6
37  b = 24
38  n = 35
39
40  result=phi(n,1)
41  ans=1
42  for i=1:length(result)
43      ans=ans*result(i)
44  end
45
46  if(ans == b) then
47      disp(1)
```

Scilab code Exa 9.16 Find the result of 20 pow 62 Mod 77

```
1  // Chapter No : 9  Exercise Number : 9.16 of the
       Book Name : Cryptography and Network Security by
       Behrouz Forouzan, Special Indian Edition, 2007
2  // This file must be used under the terms of the
       CeCILL.
3  // This source file is licensed as described in the
       file COPYING, which
4  // you should have received as part of this
       distribution.  The terms
5  // are also available at
6  // http://www.cecill.info/licences/Licence_CeCILL_V2
```

```
    -en.txt
7 //This Source file is Written by Student Shreya
    Rajiv Somkuwar(15BCE1225), Student J Robin Raj(15
    BCE1325) Guided by Dr. T. Subbulakshmi, Professor
8 //School of Computing Science and Engineering, VIT
    University Chennai
9 //The Operating System used for writing the code
    found in this file is Windows 8
10 //SCILAB version 5.5.2
11 //
   |-------------------------------------------------------------

12 //|This worked out example found in Page No: 235 of
    the book will do the following operations
                                  |
13 //|1. Find the result of 20^62 mod 77
14 //|2. Print the result in the scilab command line
15 clc;
16 clear;
17 function result=phi(n,e)
18     x=factor(n)
19     if(n==1)
20         result=0
21     elseif(x==n)
22         result=(n^e)-(n^(e-1))
23     else
24        [a,b]=unique(x)
25        for i=1:length(a)
26            if(i<length(a))
27                result(i)=phi(a(i),b(i+1)-b(i))
28            else
29                result(i)=phi(a(i),length(x)+1-b(i))
30            end
31        end
32     end
33 endfunction
34
35 a = 20
```

```
36 b = 62
37 n = 77
38
39 result=phi(n,1)
40 ans=1
41 for i=1:length(result)
42     ans=ans*result(i)
43 end
44
45 trySuccess = 0
46 finalAns = 1
47 for i=1:5
48     tempB = b-i
49     if(pmodulo(tempB,ans)==0) then
50         finalAns = pmodulo(a,n)
51         finalAns = finalAns * pmodulo(a^(i-1),n)
52         finalAns = pmodulo(finalAns,n)
53         trySuccess = 1
54         break
55     end
56 end
57 if(trySuccess==1)
58     disp(finalAns)
```

Scilab code Exa 9.17 Multiplicative inverses

```
1 // Chapter No : 9  Exercise Number : 9.17 of the
    Book Name : Cryptography and Network Security by
    Behrouz Forouzan, Special Indian Edition, 2007
2 // This file must be used under the terms of the
    CeCILL.
3 // This source file is licensed as described in the
    file COPYING, which
4 // you should have received as part of this
    distribution.  The terms
```

```
5 // are also available at
6 // http://www.cecill.info/licences/Licence_CeCILL_V2
    -en.txt
7 //This Source file is Written by Student Shreya
    Rajiv Somkuwar(15BCE1225), Student J Robin Raj(15
    BCE1325) Guided by Dr. T. Subbulakshmi, Professor
8 //School of Computing Science and Engineering, VIT
    University Chennai
9 //The Operating System used for writing the code
    found in this file is Windows 8
10 //SCILAB version 5.5.2
11 //
   |-------------------------------------------------------------------

12 //|This worked out example found in Page No: 235 of
    the book will do the following operations
                                  |
13 //|1. Find the multiplicative inverse of 8 in Z(77),
    7 in Z(15), 60 in Z(187) and 71 in Z(100)
14 //|2. Print the result in the scilab command line
15 clc;
16 clear;
17 function result=phi(n,e)
18     x=factor(n)
19     if(n==1)
20         result=0
21     elseif(x==n)
22         result=(n^e)-(n^(e-1))
23     else
24        [a,b]=unique(x)
25        for i=1:length(a)
26            if(i<length(a))
27                result(i)=phi(a(i),b(i+1)-b(i))
28            else
29                result(i)=phi(a(i),length(x)+1-b(i))
30            end
31        end
32     end
```

```
33 endfunction
34
35
36
37 //(a) Multiplicative Inverse of 8 modulus 77
38 a1=8
39 n1=77
40
41 result1=phi(n1,1)
42 ans1=1
43 for i=1:length(result1)
44     ans1=ans1*result1(i)
45 end
46
47 printf("The answer is (%d^%d) mod %d! To big
      calculation for final Answer!!",a1,ans1-1,n1)
48
49 //(b) Multiplicative Inverse of 7 modulus 15
50 a2=7
51 n2=15
52
53 result2=phi(n2,1)
54 ans2=1
55 for i=1:length(result2)
56     ans2=ans2*result2(i)
57 end
58
59 printf("\nThe answer is %d!",pmodulo(a2^(ans2-1),n2)
      )
60
61
62 //(c) Multiplicative Inverse of 60 modulus 187
63 a3=60
64 n3=187
65
66 result3=phi(n3,1)
67 ans3=1
68 for i=1:length(result3)
```

```
69      ans3=ans3*result3(i)
70  end
71
72  printf("\nThe answer is (%d^%d) mod %d! To big
       calculation for final Answer!!",a3,ans3-1,n3)
73
74
75  //(d) Multiplicative Inverse of 71 modulus 100
76  a4=71
77  n4=100
78
79  result4=phi(n4,1)
80  ans4=1
81  for i=1:length(result4)
82      ans4=ans4*result4(i)
83  end
84
85  printf("\nThe answer is (%d^%d) mod %d! To big
       calculation for final Answer!!",a4,ans4-1,n4)
```

Chapter 10

Asymmetric key Cryptography

Scilab code Exa 10.3 how the tuple x is found using inverse knapsacksum routine

```
1 // Chapter No : 10  Exercise Number : 10.3 of the
    Book Name : Cryptography and Network Security by
    Behrouz Forouzan, Special Indian Edition, 2007
2 // This file must be used under the terms of the
    CeCILL.
3 // This source file is licensed as described in the
    file COPYING, which
4 // you should have received as part of this
    distribution.  The terms
5 // are also available at
6 // http://www.cecill.info/licences/Licence_CeCILL_V2
    -en.txt
7 //This Source file is Written by Snigdha gupta (15
    BCE1087) Guided by Dr. T. Subbulakshmi, Professor
8 //School of Computing Science and Engineering, VIT
    University Chennai
9 //The Operating System used for writing the code
    found in this file is Windows 10
10 //SCILAB version 5.5.2
11 //---inverse knapsack Cryptosystem---
12 clc;
```

```
13 clear;
14 printf("\n ---------Ex 10.3--------\n\n\n");
15 s=272;//initial value of inverse knapsack sum
16 st=[0 0 0 0 0 0];//array to store various inverrse
      knapsack sums that will be generated
17 a=[17 25 46 94 201 400]; //given tuple values
18 x=[0 0 0 0 0 0];//variable to check where the
      knapsack routine is true or not
19 function[]=inv_knapsack(s,a,x,st)
20     for i=1:length(a)
21         len=7-i;
22         st(len)=s;//storing the sum after every
              iteration
23         if(s>=a(len))//checking if the inverse
              knapsack sum is true(1) or false(0)
24             x(len)=1;
25             s=s-a(len);//generating value of the
                  next inverse knapsack sum
26         else
27             x(len)=0;
28         end
29     end
30     printf("i\tai\ts\ts>=ai\tx\n\n");//printing all
          the values of inverse knapsack sum along with
           tules a and x in tabular form
31     for i=1:length(a)
32         len=7-i;
33         if (x(len)==0)
34                 printf("%d\t%d\t%d\tfalse\t%d",len,a(
                      len),st(len),x(len));//printing
                      all the inverse knapsacks that are
                       false
35                 printf("\n");
36         else
37                  printf("%d\t%d\t%d\ttrue\t%d",len,a(
                      len),st(len),x(len));//printing
                      all the inverse knapsacks that
                      are true
```

```
38                  printf("\n");
39              end
40
41      end
42      printf("\nThe knapsacks are as follows:");//
           printing the correct knapsack values.
43      for i=1:length(a)
44          if(x(i)==1)
45              printf("    %d\t",a(i));
46          end
47      end
48  endfunction
49  inv_knapsack(s,a,x,st);//calling the inwerse
       knapsack function
```

Scilab code Exa 10.5 Proof of RSA

```
1  // Chapter No : 10  Exercise Number : 10.5 of the
       Book Name : Cryptography and Network Security by
       Behrouz Forouzan, Special Indian Edition, 2007
2  // This file must be used under the terms of the
       CeCILL.
3  // This source file is licensed as described in the
       file COPYING, which
4  // you should have received as part of this
       distribution.  The terms
5  // are also available at
6  // http://www.cecill.info/licences/Licence_CeCILL_V2
       -en.txt
7  //This Source file is Written by Snigdha gupta (15
       BCE1087) Guided by Dr. T. Subbulakshmi, Professor
8  //School of Computing Science and Engineering, VIT
       University Chennai
9  //The Operating System used for writing the code
       found in this file is Windows 10
```

```
10 //SCILAB version 5.5.2
11 //--- Proof of RSA---
12 clc;
13 clear;
14 printf("\n ---------Ex 10.5---------\n\n\n");
15 //Bob chooses value of p and q
16 p=7;
17 q=11;
18 // Value of n is found by multiplying p and q
19 n=p*q;
20 //fi(n) calculated by multiplying p-1 and q-1
21 fi=(p-1)*(q-1);
22 e=13;//given encryptn key
23 d=37;//given decryption key
24 m=1;//value for checking inverse modulo
25 l=(e*d);//multiplying encryption and decrytion
26 lm= modulo(l,fi);//modulo of encryption decryption
      product with fi(n)
27 pt=5;//given plain text
28 ct=0;//declaring ciphertext variable
29 if(lm==m)//checking if encryption decryptin modulo
      with fi(n) if 1 or not
30     printf("\n The plaintext to be sent from Alice
          to Bob is %d ",pt);
31     ct=modulo((pt^e),n);// calculating the cipher
          text
32     printf("\n The cipher text sent is %d",ct);
33 end
34 c=26;
35 if (c==ct)
36     printf("\n The ciphertext recieved by Bob from
          Alice is %d",ct);
37     ptt=modulo((ct^d),n);//calculating the plain
          text
38 end
```

Scilab code Exa 10.7 Proof of RSA

```
1 // Chapter No : 10  Exercise Number : 10.7 of the
    Book Name : Cryptography and Network Security by
    Behrouz Forouzan, Special Indian Edition, 2007
2 // This file must be used under the terms of the
    CeCILL.
3 // This source file is licensed as described in the
    file COPYING, which
4 // you should have received as part of this
    distribution.  The terms
5 // are also available at
6 // http://www.cecill.info/licences/Licence_CeCILL_V2
    -en.txt
7 //This Source file is Written by Snigdha gupta (15
    BCE1087) Guided by Dr. T. Subbulakshmi, Professor
8 //School of Computing Science and Engineering, VIT
    University Chennai
9 //The Operating System used for writing the code
    found in this file is Windows 10
10 //SCILAB version 5.5.2
11 //---Proof of RSA---
12 clc;
13 clear;
14 //Bob chooses value of p and q
15 p=397;
16 q=401;
17 // Value of n is found by multiplying p and q
18 n=p*q;
19 //fi(n) calculated by multiplying p-1 and q-1
20 fi=(p-1)*(q-1);
21 e=343;//given encryptn key
22 d=12007;//given decryption key
23 m=1;
```

```
24 s=0;//rndom valu to
25 l=(e*d);
26 lm= modulo(l,fi);//modulo of encryption decryption
      product with fi(n)
27 printf("\n Let the input string by Jennifer is NO");
28 printf("\n We convert NO into (00-25) ie NO=1314.");
29 pt=1314;//given plain text
30 t=1;
31 //procedure to calculate cipher text
32 for i=1:49
33     s=(1314^4);
34     t=t*(modulo(s,n));
35 end
36 //tu=modulo(t,n);
37 u=1;
38 for i=1:49
39     s=(1314^3);
40     u=u*(modulo(s,n));
41 end
42 ut=modulo(u,n);
43 ct=0;
44 if(lm==m)//checking if encryption decryptin modulo
      with fi(n) if 1 or not
45     printf("\n The plaintext to be sent from Alice
          to Bob is %d ",pt);
46     ct=u*t;// calculating the cipher text
47     printf("\n The cipher text sent is %d",ct);
48 end
```

Scilab code Exa 10.8 RSA encryption and decryption

```
1 // Chapter No : 10  Exercise Number : 10.8 of the
     Book Name : Cryptography and Network Security by
     Behrouz Forouzan, Special Indian Edition, 2007
2 // This file must be used under the terms of the
```

```
    CeCILL.
3 // This source file is licensed as described in the
    file COPYING, which
4 // you should have received as part of this
    distribution.  The terms
5 // are also available at
6 // http://www.cecill.info/licences/Licence_CeCILL_V2
    -en.txt
7 //This Source file is Written by Snigdha gupta (15
    BCE1087) Guided by Dr. T. Subbulakshmi, Professor
8 //School of Computing Science and Engineering, VIT
    University Chennai
9 //The Operating System used for writing the code
    found in this file is Windows 10
10 //SCILAB version 5.5.2
11 //--- RSA encryption and decryption---
12 clc;
13 clear;
14 printf("\n\nWe choose 512 bit p and q and calculate
    n and fi(n), wthen we choose e and test for
    relative primeness with fi(n). We then calculate
    d. Finally we show the results of encryption and
    decryption. The intege p is a 159 digit number.")
    ;
15 //We choose value of p and q
16 printf("\n\n p
    =961303453135835045741915812806154279093098455949962158225831508
    ");
17 printf("\n\n q is a 160 digit number:\n q
    =120601919572314469182767942044508960015559250546370339360617983
    ");
18 // Value of n is found by multiplying p and q
19 printf("\n \nThe value of n=p*q\n =
    115935041739676149688925098646588752377145737545414477548552613761
    ");
20 //fi(n) calculated by multiplying p-1 and q-1
21 printf("\n \nfi(n)
    =115935041739676149676149688925098646158875237714737545414477548
```

```
    ");
22 //given encryptn key
23 printf("\n \ne=35535");
24 //given decryption key
25 printf("\n\nd
    =58008302860037763936093661289677946690620896509621804228661113 80
    ");
26 //given plain text
27 printf("\n\n Alice wants to send the message THIS IS
     A TEST which can be changed to a numr=eric value
     using the 00-26 encoding scheme.");
28 //plaintext converted in numeric values
    corresponding to their alphabet position
29 printf("\n\n Plaintext= 19070818260818260026190418 19
    ");
30 // calculating the cipher text
31 printf("\n\n The cipher text caculated by Alice is C
    =P^e which is:");
32 //cipher text is as follows
33 printf("\n
    47530912364622682720635550610545119741125200568297979457173603610
    ");
34 // calculating plain text from cipher text
35 printf("\n\n Bob can recover the plain text from the
     cipher text using P=C^d which is as follows\n
    Plaintext= 19070818260818260026190418 19");
36 //converting plaintext to alphabetic message
37 printf("\n\n Recovered plaintext is THIS IS A TEST
    after decoding");
```

Scilab code Exa 10.9 Rabin Cryptosystem

```
1 // Chapter No : 10  Exercise Number : 10.9 of the
    Book Name : Cryptography and Network Security by
    Behrouz Forouzan, Special Indian Edition, 2007
```

```
2 // This file must be used under the terms of the
    CeCILL.
3 // This source file is licensed as described in the
    file COPYING, which
4 // you should have received as part of this
    distribution.  The terms
5 // are also available at
6 // http://www.cecill.info/licences/Licence_CeCILL_V2
    -en.txt
7 //This Source file is Written by Snigdha gupta (15
    BCE1087) Guided by Dr. T. Subbulakshmi, Professor
8 //School of Computing Science and Engineering, VIT
    University Chennai
9 //The Operating System used for writing the code
    found in this file is Windows 10
10 //SCILAB version 5.5.2
11 //--- Rabin Cryptosytem---
12 clc;
13 clear;
14 printf("\n----------Ex10.9----------");
15 //Bob selects the value of p and q which are
    congruent to eah other
16 p=23;//value of p
17 q=7;//value of q
18 //value of n is found by multiplying p and q
19 n=p*q;
20 printf("\n The value of n=%d",n);
21 printf("\n The plain text to be sent by ALice is pt
    =24");
22 pt=24;//given plain from Alice
23 //plaintext and the value of n are relatively prime
24 printf("\n 24 and 161 are relatively prime");
25 c=modulo((pt^2),161);//calculating ciphertext
26 printf("\n Cipertext sent to Bob is %d",c);
27 //Bob calculates all the possible combinations using
    p and q
28 //the combinations through which plain text can be
    generated using the cipher text
```

```
29 p1=(p+1)/4;//powers are being calclated
30 p2=(q+1)/4;
31 a1=modulo((c^p1),23);//values are being generated
32 a2=-(modulo((c^p1),23));
33 a2=a2+23;//finding the positive value for the modulo
       function
34 b1=modulo((c^p2),7);//values are being generated
35 b2=-(modulo((c^p2),7));
36 b2=b2+7;//finding the posiitve value for the modulo
     function
37 //all the values are then printed
38 printf("\n possible answers:\n");
39 printf("\n(%d,%d)",a1,b1);
40 printf("\n(%d,%d)",a1,b2);
41 printf("\n(%d,%d)",a2,b1);
42 printf("\n(%d,%d)",a2,b2);
43 printf("\n Note that only these four answers, when
     squared modulo n,give the cypher text 93 sent by
     Alice");
```

Scilab code Exa 10.10 Elgamal Algorithm to develop ciphertext and plaintext

```
1 // Chapter No : 10  Exercise Number : 10.10 of the
     Book Name : Cryptography and Network Security by
     Behrouz Forouzan, Special Indian Edition, 2007
2 // This file must be used under the terms of the
     CeCILL.
3 // This source file is licensed as described in the
     file COPYING, which
4 // you should have received as part of this
     distribution.  The terms
5 // are also available at
6 // http://www.cecill.info/licences/Licence_CeCILL_V2
     -en.txt
7 //This Source file is Written by Snigdha gupta (15
```

```
    BCE1087) Guided by Dr. T. Subbulakshmi, Professor
8 //School of Computing Science and Engineering, VIT
    University Chennai
9 //The Operating System used for writing the code
    found in this file is Windows 10
10 //SCILAB version 5.5.2
11 //---Elgamal Algorithm to develop ciphertext and
    plaintext---
12 clc;
13 clear;
14 printf("\n----------Ex10.10----------");
15 //Bob chooses the value of a large prime number p
    and e1- encryption key 1
16 p=11;
17 e1=2;//2 is a premitive root of Z-11
18 //Bob chooses the value of decryption key
19 d=3;
20 //encryption key e2 is calculated using e1 and d
21 e2=e1^d;
22 printf("\n public key are %d %d %d",e1,e2,p);
23 printf("\n private key is %d",d);
24 //public keys p,e1,e2 sent to aLice
25 //decryption key kept private
26 r=4;//Alice chooses the value of r
27 pt=7;//given plain text
28 mp=e2^r;//value for e2 with power r is calculated
29 c1=modulo((e1^r),p);//cipher text c1 is calculated
30 c2=modulo((pt*mp),p);//cipher text c2 is calculated
31 //Bob recieves the ciphertexts c1 and c2 and
    calculates the plain text
32 printf("\n The converted cipher text is %d %d",c1,c2
    );
33 cc=c1^(d-1);//calculation on cipher text c1 is done
34 ccd=modulo(cc,p);//plaintext is calculated frm the
    given cipher text
35 npt=modulo((c2*ccd),p);
36 printf("\n Recieved plain text by Bob is %d",npt);
```

Scilab code Exa 10.11 Elgamal Algorithm to develop ciphertext and plaintext

```
1 // Chapter No : 10  Exercise Number : 10.11 of the
    Book Name : Cryptography and Network Security by
    Behrouz Forouzan, Special Indian Edition, 2007
2 // This file must be used under the terms of the
    CeCILL.
3 // This source file is licensed as described in the
    file COPYING, which
4 // you should have received as part of this
    distribution.  The terms
5 // are also available at
6 // http://www.cecill.info/licences/Licence_CeCILL_V2
    -en.txt
7 //This Source file is Written by Snigdha gupta (15
    BCE1087) Guided by Dr. T. Subbulakshmi, Professor
8 //School of Computing Science and Engineering, VIT
    University Chennai
9 //The Operating System used for writing the code
    found in this file is Windows 10
10 //SCILAB version 5.5.2
11 //---Elgamal Algorithm to develop ciphertext and
    plaintext (Fermats Theorem)---
12 clc;
13 clear;
14 printf("\n----------Ex10.11----------");
15 //Bob chooses the value of a large prime number p
    and encryption key e1
16 p=11;
17 e1=2;//2 is a premitive root of Z-11
18 d=3;//Bob chooses the value of decryption key
19 e2=e1^d;//value of encryption key e2 is calculated
    using encryption key e1 and decryption key d
20 //public keys p,e1,e2 sent to aLice
```

```
21 //decryption key kept private
22 printf("\n public key are %d %d %d",e1,e2,p);
23 printf("\n private key is %d",d);
24 r=4;//Alice chooses the value of r
25 pt=7;//given plain text
26 mp=e2^r;//value for e2 with power r is calculated
27 c1=modulo((e1^r),p);//cipher text c1 is calculated
28 c2=modulo((pt*mp),p);//cipher text c2 is calculated
29 //Bob recieves the ciphertexts c1 and c2 and
     calculates the plain text
30 printf("\n The converted cipher text is %d %d",c1,c2
     );
31 cc=c1^(p-d-1);//calculation on cipher text c1 is
     done using Fermats Theorem
32 npt=modulo((c2*cc),p);//plaintext is calculated from
     the given cipher text
33 printf("\n Recieved plain text by Bob is %d",npt);
```

Scilab code Exa 10.12 Elgamal Algorithm to develop ciphertext and plaintex

```
1 // Chapter No : 10  Exercise Number : 10.12 of the
    Book Name : Cryptography and Network Security by
    Behrouz Forouzan, Special Indian Edition, 2007
2 // This file must be used under the terms of the
    CeCILL.
3 // This source file is licensed as described in the
    file COPYING, which
4 // you should have received as part of this
    distribution.  The terms
5 // are also available at
6 // http://www.cecill.info/licences/Licence_CeCILL_V2
    -en.txt
7 //This Source file is Written by Snigdha gupta (15
    BCE1087) Guided by Dr. T. Subbulakshmi, Professor
8 //School of Computing Science and Engineering, VIT
```

```
    University Chennai
9  //The Operating System used for writing the code
    found in this file is Windows 10
10 //SCILAB version 5.5.2
11 //---Elgamal Algorithm to develop ciphertext and
    plaintext---
12 clc;
13 clear;
14 printf("\n----------Ex10.12----------");
15 //Bob chooses the value of a large prime number p
    and encryption key e1
16 printf("\n public keys announced by Bob are as
    follows:\n");
17 printf("p
    =11534899272561676244925313717014331740490094532609834959814346692
    ");
18 e1=2;
19 printf("\ne1=%d",e1);
20 //e2
    =97886413043009189508766856938097739043880062887337687610022062223

21 ////Bob chooses the value of decryption key
22 d=1007;
23 //printf("\n public key are \n%d \n%d \n%d",e1,e2,p)
    ;
24 printf("\n\nprivate key is d=%d",d);
25 //value of encryption key e2 is calculated using
    encryption key e1 and decryption key d
26 printf("\ne2
    =97886413043009189508766856938097739043880062887337687610022062223
    ");
27 //public keys p,e1,e2 sent to aLice
28 //Alice chooses the value of r and the plain text
29 printf("\n\nAlice chooses the following \nr=545131\
    nplaintext=3200");
30 //plain text is converted into cipher text using
    elgamal crptosystems with Fermats Theorem
31  printf("\nThe converted cipher text is \nc1
```

```
    =887297069383528471022570471492275663120260067256562125018188351
    nc2
    =708454333048929944577016012380794999567436021836192446961774506
    ");
32 //plain text is calculated back from the generated
    cipher text
33 printf("\n\nCalculated plain text by Bob is %d"
    ,3200);
```

Scilab code Exa 10.13 Elliptic curve cryptosystem

```
1 // Chapter No : 10  Exercise Number : 10.13 of the
    Book Name : Cryptography and Network Security by
    Behrouz Forouzan, Special Indian Edition, 2007
2 // This file must be used under the terms of the
    CeCILL.
3 // This source file is licensed as described in the
    file COPYING, which
4 // you should have received as part of this
    distribution.  The terms
5 // are also available at
6 // http://www.cecill.info/licences/Licence_CeCILL_V2
    -en.txt
7 //This Source file is Written by Snigdha Gupta (15
    bce1087) Guided by Dr. T. Subbulakshmi, Professor
8 //School of Computing Science and Engineering, VIT
    University Chennai
9 //The Operating System used for writing the code
    found in this file is Windows 8
10 //SCILAB version 5.5.2
11 //---elliptic curve cryptosystem---
12 clc;
13 clear;
14 printf("\n Figure 10.12 sows two elliptic curves
    with equations y^2=x^3-4x and y^2=x^3-1. Both are
```

```
nonsingular. However, the first has three real
roots (x=-2,x=0 and x=2), but the second has only
one real root (x=1) and two imaginary ones.");
```

Scilab code Exa 10.14 Finding points on curve using the given elliptic curve equat

```
1 // Chapter No : 10  Exercise Number : 10.14 of the
    Book Name : Cryptography and Network Security by
    Behrouz Forouzan, Special Indian Edition, 2007
2 // This file must be used under the terms of the
    CeCILL.
3 // This source file is licensed as described in the
    file COPYING, which
4 // you should have received as part of this
    distribution.  The terms
5 // are also available at
6 // http://www.cecill.info/licences/Licence_CeCILL_V2
    -en.txt
7 //This Source file is Written by Snigdha gupta (15
    BCE1087) Guided by Dr. T. Subbulakshmi, Professor
8 //School of Computing Science and Engineering, VIT
    University Chennai
9 //The Operating System used for writing the code
    found in this file is Windows 10
10 //SCILAB version 5.5.2
11 //---Finding points on curve using the given
    elliptic curve equation---
12 clc;
13 clear;
14 //values of x and coefficients for the given
    elliptic curve equation
15 x=0;
16 a=1;//coefficient of x^3
17 b=1;//coefficient of x
18 p=13;//given modulo for calculation
```

```
19 //declaring variables to be used
20 x3=0;
21 ax=0;
22 w=0;
23 y=0;
24 y2=0;
25 ww=0;
26 printf("\n points on the given elliptic curve are as
       follows: ")
27 while x<(p-1) // finding the points on elliptc curve
28     x3=x*x*x;//value of x^3
29     ax=a*x;//
30     ww=x3+ax+b;//function of given ellipic curve
31     w=modulo((ww),p);//modulo of elliptic function
          wrt to given modulo
32     if w==1|w==4|w==9|w==3|w==12|w==10|w==0 //cases
          with the values that fall within the modulo
          set
33         y=sqrt(w);
34         y2=modulo(p-y,p);//calculating and finding
              the points
35         printf("\n(%d,%d)",x,y);//printing the
              points
36         printf("\t(%d,%d)",x,y2);
37     end
38     x=x+1;
39 end;
40 //notes related to elliptic curve
41 printf("\n Note the following: ");
42 printf("\n a. Some values of y^2 donot have a square
       root in modulo 13 arithmatic. These are not
      points on this elliptic curve.");
43 printf("\n b. Each points defined for the curve has
      an inverse. The inverses are listed as pairs. Not
       that (7,0) is the inverse of itself.");
44 printf("\n c. Note that for a pair of inverse points
      , the y value are additive inverses of each other
       in Zp");
```

```
45 printf("\n d. The inverse are on the same vertical
      lines.")
```

Chapter 12

Cryptographic Hash function

Scilab code Exa 12.1 example to show message length limitation in SHA 512

```
1
2 // Chapter No : 12  Exercise Number : 12.1 of the
    Book Name : Cryptography and Network Security by
    Behrouz Forouzan, Special Indian Edition, 2007
3 // This file must be used under the terms of the
    CeCILL.
4 // This source file is licensed as described in the
    file COPYING, which
5 // you should have received as part of this
    distribution.  The terms
6 // are also available at
7 // http://www.cecill.info/licences/Licence_CeCILL_V2
    -en.txt
8 //This Source file is Written by Snigdha gupta (15
    BCE1087) Guided by Dr. T. Subbulakshmi, Professor
9 //School of Computing Science and Engineering, VIT
    University Chennai
10 //The Operating System used for writing the code
    found in this file is Windows 10
11 //SCILAB version 5.5.2
12 //--- example to show message length limitation in
```

```
    SHA-512---
clc;
clear;
disp("This example shows that the message length
    limitation of SHA-512 is not a serious problem.
    ")
disp("A communication network that can send 2^64
    bits per second is not yet available Even if it
    were, it would take many years to send this
    message. This tells us that we do not need to
    worry about the SHA-512 message length
    restriction");
```

Scilab code Exa 12.2 pages occupied by message 2 pow 128 bits in SHA 512

```

// Chapter No : 12  Exercise Number : 12.2 of the
    Book Name : Cryptography and Network Security by
    Behrouz Forouzan, Special Indian Edition, 2007
// This file must be used under the terms of the
    CeCILL.
// This source file is licensed as described in the
    file COPYING, which
// you should have received as part of this
    distribution.  The terms
// are also available at
// http://www.cecill.info/licences/Licence_CeCILL_V2
    -en.txt
//This Source file is Written by Snigdha gupta (15
    BCE1087) Guided by Dr. T. Subbulakshmi, Professor
//School of Computing Science and Engineering, VIT
    University Chennai
//The Operating System used for writing the code
    found in this file is Windows 10
//SCILAB version 5.5.2
```

```
12 //---pages occupied by message 2^128 bits in SHA
     -512---
13 clc;
14 clear;
15 printf("This example also concerns the message
     length in SHA-512. \n\nSuppose that a character
     is 32, or 2^6 bits. \nEach page is less than
     2048, or approximately 2^12, characters. \nSo
     2^128/2^18, or 2^110, pages. \nThis again shows
     that we need not worry about the message length
     restriction.")
```

Scilab code Exa 12.3 Padding bit generation in SHA 512 for a given message

```
1
2 // Chapter No : 12  Exercise Number : 12.3 of the
     Book Name : Cryptography and Network Security by
     Behrouz Forouzan, Special Indian Edition, 2007
3 // This file must be used under the terms of the
     CeCILL.
4 // This source file is licensed as described in the
     file COPYING, which
5 // you should have received as part of this
     distribution.  The terms
6 // are also available at
7 // http://www.cecill.info/licences/Licence_CeCILL_V2
     -en.txt
8 //This Source file is Written by Snigdha gupta (15
     BCE1087) Guided by Dr. T. Subbulakshmi, Professor
9 //School of Computing Science and Engineering, VIT
     University Chennai
10 //The Operating System used for writing the code
     found in this file is Windows 10
11 //SCILAB version 5.5.2
12 //---Padding bit generation in SHA-512 for a given
```

```
    message---
clc;
clear;
printf("\n Given Message = 2590");
m=2590;//given message bits
a=-2590-128;//the message subtracted by 128 (SHA
    bits)
p1=modulo(a,1024);//calculating padding bits
p=p1+1024;//finding the positive value
printf("\n Padding = %d ",p);//padding value
p=p-1;//finding number of 0's in padding
printf("\n Padding has one 1 and %d number of 0",p);
```

Scilab code Exa 12.4 Need of padding if original message length multiple of 1024 b

```

// Chapter No : 12  Exercise Number : 12.4 of the
    Book Name : Cryptography and Network Security by
    Behrouz Forouzan, Special Indian Edition, 2007
// This file must be used under the terms of the
    CeCILL.
// This source file is licensed as described in the
    file COPYING, which
// you should have received as part of this
    distribution.  The terms
// are also available at
// http://www.cecill.info/licences/Licence_CeCILL_V2
    -en.txt
//This Source file is Written by Snigdha gupta (15
    BCE1087) Guided by Dr. T. Subbulakshmi, Professor
//School of Computing Science and Engineering, VIT
    University Chennai
//The Operating System used for writing the code
    found in this file is Windows 10
//SCILAB version 5.5.2
```

```
12 //---Need of padding if original message length
      multiple of 1024 bits---
13 clc;
14 clear;
15 printf("\n We need padding if the length of the
      original message is already a multiple of 1024
      bits.\n \nThis is because we need to add the
      length field.\n So padding is needed to make the
      new block a multiple of 1024 bits.")
```

Scilab code Exa 12.5 Minimum and maximum number of padding bits

```
1 // Chapter No : 12  Exercise Number : 12.5 of the
     Book Name : Cryptography and Network Security by
     Behrouz Forouzan, Special Indian Edition, 2007
2 // This file must be used under the terms of the
     CeCILL.
3 // This source file is licensed as described in the
     file COPYING, which
4 // you should have received as part of this
     distribution.  The terms
5 // are also available at
6 // http://www.cecill.info/licences/Licence_CeCILL_V2
     -en.txt
7 //This Source file is Written by Snigdha gupta (15
     BCE1087) Guided by Dr. T. Subbulakshmi, Professor
8 //School of Computing Science and Engineering, VIT
     University Chennai
9 //The Operating System used for writing the code
     found in this file is Windows 10
10 //SCILAB version 5.5.2
11 //---Minimum and maximum number of padding bits---
12 clc;
13 clear;
14 printf("\n a. This minimum length of padding is 0
```

```
    and it happens when (-M-128) mod 1024 is 0.\n
    This means that |M| = -128 mod 1024 = 896 mod
    1024 bits.\n We add a 128 bit length field to
    make the block complete.");
15  printf("\n\n b. The maximum length of padding is
    1023 and it happens when (-M-128)= 1023 mod 1024.
     \nThis means that the length of the original
    message is |M|= (-128-1023) mod 1024 or the
    length is |M| =897 mod 1024.\n In this case , we
    cannot just add the length field because the
    length of the last block exceeds one bit more
    than 1024. \n So we need to add 897 bits o
    complete this block and create a second block of
    896 bits. \n Now the length can be added to make
    this block complete.")
```

Scilab code Exa 12.6 how to develop W60

```
1  // Chapter No : 12  Exercise Number : 12.6 of the
   Book Name : Cryptography and Network Security by
   Behrouz Forouzan, Special Indian Edition, 2007
2  // This file must be used under the terms of the
   CeCILL.
3  // This source file is licensed as described in the
   file COPYING, which
4  // you should have received as part of this
   distribution.  The terms
5  // are also available at
6  // http://www.cecill.info/licences/Licence_CeCILL_V2
   -en.txt
7  //This Source file is Written by Snigdha gupta (15
   BCE1087) Guided by Dr. T. Subbulakshmi, Professor
8  //School of Computing Science and Engineering, VIT
   University Chennai
9  //The Operating System used for writing the code
```

```
    found in this file is Windows 10
10 //SCILAB version 5.5.2
11 //---how to develop W60---
12 clc;
13 clear;
14 printf("\n Each word in range W-16 to W-79 is made
    from forur previously made words. W-60 is made as
    :");
15 printf("\n \nW60= W44 XOR RotShift(1-8-7) W45 XOR
    W53 XOR RotShift(19-61-6)W58");
```

Scilab code Exa 12.7 conditional function

```
1 // Chapter No : 12  Exercise Number : 12.7 of the
    Book Name : Cryptography and Network Security by
    Behrouz Forouzan, Special Indian Edition, 2007
2 // This file must be used under the terms of the
    CeCILL.
3 // This source file is licensed as described in the
    file COPYING, which
4 // you should have received as part of this
    distribution.  The terms
5 // are also available at
6 // http://www.cecill.info/licences/Licence_CeCILL_V2
    -en.txt
7 //This Source file is Written by Snigdha gupta (15
    BCE1087) Guided by Dr. T. Subbulakshmi, Professor
8 //School of Computing Science and Engineering, VIT
    University Chennai
9 //The Operating System used for writing the code
    found in this file is Windows 10
10 //SCILAB version 5.5.2
11 //---conditional function---
12 clc;
13 clear;
```

```
14 printf("\n Digits in hexadecimal are 0x7,0xA,0xE");
15 printf("\n The digits in binary are 1001,1010,1111\n
     \n");
16 m=[0 1 1 1];
17 n=[1 0 1 0];
18 p=[1 1 1 0];
19 for i=1:length(m)
20 a=bitand(m(i),n(i));
21 b=bitand(p(i),n(i));
22 c=bitand(m(i),p(i));
23 ab=bitxor(a,b);
24 abc=bitxor(ab,c);
25 printf("\n The result for the bits on bit position
     %d is %d",i,abc);
26 end
```

Scilab code Exa 12.8 conditional function

```
1 // Chapter No : 12  Exercise Number : 12.8 of the
    Book Name : Cryptography and Network Security by
    Behrouz Forouzan, Special Indian Edition, 2007
2 // This file must be used under the terms of the
    CeCILL.
3 // This source file is licensed as described in the
    file COPYING, which
4 // you should have received as part of this
    distribution.  The terms
5 // are also available at
6 // http://www.cecill.info/licences/Licence_CeCILL_V2
    -en.txt
7 //This Source file is Written by Snigdha gupta (15
    BCE1087) Guided by Dr. T. Subbulakshmi, Professor
8 //School of Computing Science and Engineering, VIT
    University Chennai
9 //The Operating System used for writing the code
```

```
    found in this file is Windows 10
//SCILAB version 5.5.2
//--- Conditional Function---
clc;
clear;
printf("\n The digits in hexadecimal are 0x9, 0xA, 0
    xF");
printf("\n The digits in binary are 1001,1010,1111\n
    \n");
m=[1 0 0 1];
n=[1 0 1 0];
p=[1 1 1 1];
for i=1:length(m)
a=bitand(m(i),n(i));
if(n(i)==1)
    b=0;
else
    b=1;
end
c=bitand(b,p(i));
ab=bitxor(a,c);//similar procedure is done for all
    the bits.
printf("\n The result for the bits on bit position
    %d is %d",i,ab);
end
```

Chapter 15

Key Management

Scilab code Exa 15.1 SYMMETRIC KEY AGREEMENT

```
1
2 // Chapter No : 15  Exercise Number : 15.1 of the
    Book Name : Cryptography and Network Security by
    Behrouz Forouzan, Special Indian Edition, 2007
3 // This file must be used under the terms of the
    CeCILL.
4 // This source file is licensed as described in the
    file COPYING, which
5 // you should have received as part of this
    distribution.  The terms
6 // are also available at
7 // http://www.cecill.info/licences/Licence_CeCILL_V2
    -en.txt
8 //This Source file is Written by Snigdha gupta (15
    BCE1087) Guided by Dr. T. Subbulakshmi, Professor
9 //School of Computing Science and Engineering, VIT
    University Chennai
10 //The Operating System used for writing the code
    found in this file is Windows 10
11 //SCILAB version 5.5.2
12 //SYMMETRIC KEY AGREEMENT
```

```
13 clc;
14 clear;
15
16 g=7;
17 p=23;
18 printf("\n g=%d  \n p=%d",g,p);
19 x=3; //Alice chooses x=3
20 printf("\n x=%d",x);
21 r1=modulo((g^x),p);//step 1
22 printf("\n r1=%d",r1);
23 y=6;//Bob chooses y=6
24 printf("\n y=%d",y);
25 r2=modulo((g^y),p);//step 2
26 printf("\n r2=%d",r2);
27 printf("\n\n Alice sends r1 to Bob.");//step 3
28 printf("\n Bob sends r2 to Alice");//step 4
29 xy=x*y;
30 //Calculation of symmetric key
31
32 k1=modulo((r2^x),p); // Alice calculates symmetric
      key
33 printf("\n k1=%d",k1);//step 5
34 k2=modulo((r1^y),p); //Bob clculates symmetric key
35 printf("\n k2=%d",k2);//step 6
36 if(k1==k2)
37     printf("\n\n Value of K is same for both Alice
         and Bob");
38     gxy=g^xy;
39     res=modulo(gxy,p);//final comparing of key on
         both Alice and Bob side
40     printf("\n Value of the key after checking with
         xy is %d",res);
41 end
```

Scilab code Exa 15.2 SYMMETRIC KEY AGREEMENT PROGRAM TO CREATE RANDOM INTEGER

```
1
2  // Chapter No : 15  Exercise Number : 15.2 of the
       Book Name : Cryptography and Network Security by
       Behrouz Forouzan, Special Indian Edition, 2007
3  // This file must be used under the terms of the
       CeCILL.
4  // This source file is licensed as described in the
       file COPYING, which
5  // you should have received as part of this
       distribution.  The terms
6  // are also available at
7  // http://www.cecill.info/licences/Licence_CeCILL_V2
       -en.txt
8  //This Source file is Written by Snigdha gupta (15
       BCE1087) Guided by Dr. T. Subbulakshmi, Professor
9  //School of Computing Science and Engineering, VIT
       University Chennai
10 //The Operating System used for writing the code
       found in this file is Windows 10
11 //SCILAB version 5.5.2
12 //---SYMMETRIC KEY AGREEMENT PROGRAM TO CREATE
       RANDOM INTEGER----
13 //creation of R1,R2,K
14 clc;
15 clear;
16 //very large prime number p is taken along with g, x
       and y
17 printf("\n PROGRAM TO CREATE A RANDOM INTEGER")
18 printf("\n p
       =7646242985634935721824937659550305074763380967269497489235737728
       ");
19 printf("\n g=2");
20 printf("\n x=557");
21 printf("\n y=273");
22 //Following show the valu of R1,R2,K
23 printf("\n\n Following show the values of R1,R2,and
       K");
24 printf("\n R1
```

```
    =8449202842056655052161794749110350941434336985200126608628636310
    ");
25 printf("\n R2
    =4352628387092003794707471148955816276363891162621155579751233792
    ");
26 printf("\n K
    =1556380006645222905962258275232707652732180469444236785203204001
    ");
```

Scilab code Exa 15.3 how user 1 obtains verified copy of User 3 public key

```
1
2 // Chapter No : 15  Exercise Number : 15.3 of the
    Book Name : Cryptography and Network Security by
    Behrouz Forouzan, Special Indian Edition, 2007
3 // This file must be used under the terms of the
    CeCILL.
4 // This source file is licensed as described in the
    file COPYING, which
5 // you should have received as part of this
    distribution.  The terms
6 // are also available at
7 // http://www.cecill.info/licences/Licence_CeCILL_V2
    -en.txt
8 //This Source file is Written by Snigdha gupta (15
    BCE1087) Guided by Dr. T. Subbulakshmi, Professor
9 //School of Computing Science and Engineering, VIT
    University Chennai
10 //The Operating System used for writing the code
    found in this file is Windows 10
11 //SCILAB version 5.5.2
12 //---how user 1 obtains verified copy of User3
    public key---
13 clc;
14 clear;
```

```
15 printf("\n This example shows how User1 knowing only
       the public key \n of the CA(the root), can
      obtain a verified copy of User3 public key\n")
16 printf("\n User one sends a chain of certificate, CA
      <<CA1>> and CA!<<User3>>, to User 1.");
17 printf("\n a. User1 validates CA<<CA1>> using the
      public key of CA. ");
18 printf("\n b. User1 extracts the public key of CA1
      from CA<<CA1>>.");
19 printf("\n c. User1 validates CA1<<User3>> using the
       public jey of CA1.");
20 printf("\n d. User1 extracs the public key of User3
      from CA1<<User3>>");
```

Scilab code Exa 15.4 finding list of roots n the internet explorer

```
1
2 // Chapter No : 15  Exercise Number : 15.4 of the
     Book Name : Cryptography and Network Security by
     Behrouz Forouzan, Special Indian Edition, 2007
3 // This file must be used under the terms of the
     CeCILL.
4 // This source file is licensed as described in the
     file COPYING, which
5 // you should have received as part of this
     distribution.  The terms
6 // are also available at
7 // http://www.cecill.info/licences/Licence_CeCILL_V2
     -en.txt
8 //This Source file is Written by Snigdha gupta (15
     BCE1087) Guided by Dr. T. Subbulakshmi, Professor
9 //School of Computing Science and Engineering, VIT
     University Chennai
10 //The Operating System used for writing the code
     found in this file is Windows 10
```

```
11 //SCILAB version 5.5.2
12 //---finding list of roots n the internet explorer
     ---
13 clc;
14 clear;
15 disp("Some web browsers, such as Netscape and
     internet explorer, include a set of certificates
     from independent roots without a single high
     level, authority to certify each root. One can
     find the list of these roots in the internet
     explorer at tools/internet options/contents/
     certificate/trusted roots (using pull down menu).
      The user then can choose any of this root and
     view the certificate.");
```

Scilab code Exa 15.5 How Alice obatains Bobs verified public key

```
1
2 // Chapter No : 13  Exercise Number : 13.1 of the
     Book Name : Cryptography and Network Security by
     Behrouz Forouzan, Special Indian Edition, 2007
3 // This file must be used under the terms of the
     CeCILL.
4 // This source file is licensed as described in the
     file COPYING, which
5 // you should have received as part of this
     distribution.  The terms
6 // are also available at
7 // http://www.cecill.info/licences/Licence_CeCILL_V2
     -en.txt
8 //This Source file is Written by Snigdha gupta (15
     BCE1087) Guided by Dr. T. Subbulakshmi, Professor
9 //School of Computing Science and Engineering, VIT
     University Chennai
10 //The Operating System used for writing the code
```

```
    found in this file is Windows 10
//SCILAB version 5.5.2
//---How Alice obatains Bob's verified public key---
clc;
clear;
printf("\n Bob sends a chain of certificates from
    Root4 to Bob.");
printf("\n Alice looks at the directory of Root1 to
    find Root1<<Root1>> and root1<<Root4>>
    certificates.");
printf("\n Using the process in Fig. 15.21, Alice
    can verify public key from Bob");
```