# Scilab Manual for
# Data Compression and Encryption
# by Dr Saylee Gharge
# Others
# VESIT[1]

Solutions provided by
Dr Saylee Gharge
Others
V.E.S.Institute of Technology

July 16, 2024

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

# Contents

# List of Experiments

# List of Figures

# Experiment: 1

# To implement Huffman coding.

**Scilab code Solution 1.01** Huffman

```
1  //OS: Windows 7
2  //Scilab Version: Scilab 5.4.1
3  clc;
4  clear all;
5  printf("Enter the frequency of the 5 symbols in
       decreasing order (integer values only):");//Input
        the frequency of the symbols sequentially as
       5,4,3,2,1 pressing enter after each frequency
6  s=[];r=[];
7  for i=1:5
8  s(i)=input("");
9  end
10 r(4)='0';r(5)='1';
11 if (s(4)+s(5))>=s(1) then
12 for i=1:3
13 temp(i+1)=s(i);
14 end
15 temp(1)=(s(4)+s(5));
16 for i=1:4
```

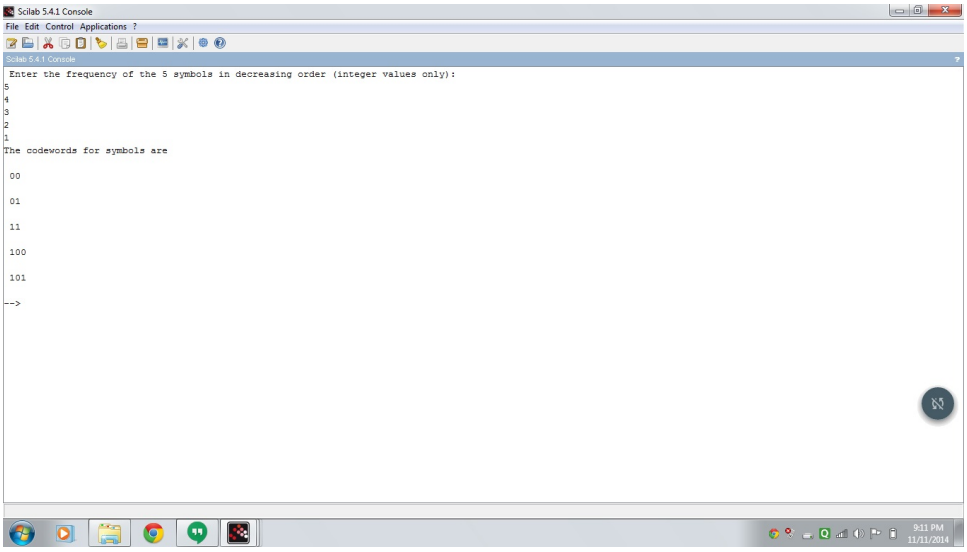


Figure 1.1: Huffman

```
17 s(i)=temp(i);
18 end
19 r(2)='0';r(3)='1';
20 if (s(3)+s(4))>=s(1) then
21 for i=1:2
22 temp(i+1)=s(i);
23 end
24 temp(1)=(s(3)+s(4));
25 for i=1:3
26 s(i)=temp(i);
27 end
28 r(1)='1';r(4)='00';r(5)='01';
29 if (s(2)+s(3))>=s(1) then
30          temp(1)=s(1);s(1)=(s(2)+s(3));s(2)=temp(1);
31          r(1)='01';r(2)='10';r(3)='11';r(4)='000';r
               (5)='001';
32 else s(2)=(s(2)+s(3));
33          r(1)='11';r(2)='00';r(3)='01';r(4)='100';r
               (5)='101';
34 end
```

```
35
36 elseif (s(3)+s(4))>=s(2) then
37     temp(2)=s(2);s(2)=(s(3)+s(4));s(3)=temp(2);
38 r(1)='1';r(2)='00';r(3)='01';
39 if (s(2)+s(3))>=s(1) then
40         temp(1)=s(1);s(1)=(s(2)+s(3));s(2)=temp(1);
41         r(1)='01';r(2)='000';r(3)='001';r(4)='10';r
               (5)='11';
42 else s(2)=(s(2)+s(3));
43         r(1)='11';r(2)='100';r(3)='101';r(4)='00';r
               (5)='01';
44 end
45 else s(3)=(s(3)+s(4));
46 r(1)='0';r(2)='10';r(3)='11';
47 if (s(2)+s(3))>=s(1) then
48         temp(1)=s(1);s(1)=(s(2)+s(3));s(2)=temp(1);
49         r(1)='00';r(2)='010';r(3)='011';r(4)='10';r
               (5)='11';
50 else s(2)=(s(2)+s(3));
51         r(1)='10';r(2)='110';r(3)='111';r(4)='00';r
               (5)='01';
52 end
53 end
54 elseif (s(4)+s(5))>=s(2) then
55 for i=2:3
56 temp(i+1)=s(i);
57 end
58 temp(2)=(s(4)+s(5));
59 for i=3:4
60 s(i)=temp(i);
61 end
62 r(2)='0';r(3)='1';
63 if (s(3)+s(4))>=s(1) then
64 for i=1:2
65 temp(i+1)=s(i);
66 end
67 temp(1)=(s(3)+s(4));
68 for i=1:3
```

```
s(i)=temp(i);
end
r(1)='0';r(4)='10';r(5)='11';
if (s(2)+s(3))>=s(1) then
        temp(1)=s(1);s(1)=(s(2)+s(3));s(2)=temp(1);
        r(1)='00';r(2)='10';r(3)='11';r(4)='010';r
           (5)='011';
else s(2)=(s(2)+s(3));
        r(1)='10';r(2)='00';r(3)='01';r(4)='110';r
           (5)='111';
end

elseif (s(3)+s(4))>=s(2) then
    temp(2)=s(2);s(2)=(s(3)+s(4));s(3)=temp(2);
r(1)='1';r(2)='00';r(3)='01';
if (s(2)+s(3))>=s(1) then
        temp(1)=s(1);s(1)=(s(2)+s(3));s(2)=temp(1);
        r(1)='01';r(2)='000';r(3)='001';r(4)='10';r
           (5)='11';
else s(2)=(s(2)+s(3));
        r(1)='11';r(2)='100';r(3)='101';r(4)='00';r
           (5)='01';
end
else s(3)=(s(3)+s(4));
    r(2)='10';r(3)='11';r(4)='00';r(5)='01';
if (s(2)+s(3))>=s(1) then
        temp(1)=s(1);s(1)=(s(2)+s(3));s(2)=temp(1);
        r(1)='1';r(2)='010';r(3)='011';r(4)='000';r
           (5)='001';
else s(2)=(s(2)+s(3));
        r(1)='0';r(2)='110';r(3)='111';r(4)='100';r
           (5)='101';
end
end
elseif (s(4)+s(5))>=s(3) then
    temp(3)=s(3);s(3)=(s(4)+s(5));s(4)=temp(3);
r(4)='00';r(5)='01';r(3)='1';
if (s(3)+s(4))>=s(1) then
```

```
for i=1:2
temp(i+1)=s(i);
end
temp(1)=(s(3)+s(4));
for i=1:3
s(i)=temp(i);
end
r(1)='0';r(2)='1';
if (s(2)+s(3))>=s(1) then
        temp(1)=s(1);s(1)=(s(2)+s(3));s(2)=temp(1);
        r(1)='00';r(2)='01';r(3)='11';r(4)='100';r
           (5)='101';
else s(2)=(s(2)+s(3));
        r(1)='10';r(2)='11';r(3)='01';r(4)='000';r
           (5)='001';
end

elseif (s(3)+s(4))>=s(2) then
    temp(2)=s(2);s(2)=(s(3)+s(4));s(3)=temp(2);
    r(2)='1';r(3)='01';r(4)='000';r(5)='001';
if (s(2)+s(3))>=s(1) then
        temp(1)=s(1);s(1)=(s(2)+s(3));s(2)=temp(1);
        r(1)='1';r(2)='01';r(3)='001';r(4)='0000';r
           (5)='0001';
else s(2)=(s(2)+s(3));
        r(1)='0';r(2)='11';r(3)='101';r(4)='1000';r
           (5)='1001';
end
else s(3)=(s(3)+s(4));
    r(2)='0';r(3)='11';r(4)='100';r(5)='101';
if (s(2)+s(3))>=s(1) then
        temp(1)=s(1);s(1)=(s(2)+s(3));s(2)=temp(1);
        r(1)='1';r(2)='00';r(3)='011';r(4)='0100';r
           (5)='0101';
else s(2)=(s(2)+s(3));
        r(1)='0';r(2)='10';r(3)='111';r(4)='1100';r
           (5)='1101';
end
```

```
133 end
134 else s(4)=(s(4)+s(5));
135 r(3)='0';r(4)='10';r(5)='11';
136 if (s(3)+s(4))>=s(1) then
137 for i=1:2
138 temp(i+1)=s(i);
139 end
140 temp(1)=(s(3)+s(4));
141 for i=1:3
142 s(i)=temp(i);
143 end
144 r(1)='0';r(2)='1';
145 if (s(2)+s(3))>=s(1) then
146         temp(1)=s(1);s(1)=(s(2)+s(3));s(2)=temp(1);
147         r(1)='00';r(2)='01';r(3)='10';r(4)='110';r
              (5)='111';
148 else s(2)=(s(2)+s(3));
149         r(1)='10';r(2)='11';r(3)='00';r(4)='010';r
              (5)='011';
150 end
151
152 elseif (s(3)+s(4))>=s(2) then
153     temp(2)=s(2);s(2)=(s(3)+s(4));s(3)=temp(2);
154     r(2)='1';r(3)='00';r(4)='010';r(5)='011';
155 if (s(2)+s(3))>=s(1) then
156         temp(1)=s(1);s(1)=(s(2)+s(3));s(2)=temp(1);
157         r(1)='1';r(2)='01';r(3)='000';r(4)='0010';r
              (5)='0011';
158 else s(2)=(s(2)+s(3));
159         r(1)='0';r(2)='11';r(3)='100';r(4)='1010';r
              (5)='1011';
160 end
161 else s(3)=(s(3)+s(4));
162     r(2)='0';r(3)='10';r(4)='110';r(5)='111';
163 if (s(2)+s(3))>=s(1) then
164         temp(1)=s(1);s(1)=(s(2)+s(3));s(2)=temp(1);
165         r(1)='1';r(2)='00';r(3)='010';r(4)='0110';r
              (5)='0111';
```

```
166 else s(2)=(s(2)+s(3));
167          r(1)='0';r(2)='10';r(3)='110';r(4)='1110';r
                (5)='1111';
168 end
169 end
170 end
171 printf("The codewords for symbols are \n")
172 for i=1:5
173 disp(r(i));
174 end
175 //Output for ex:
176 //The codewords for symbols are :
177 //00
178 //01
179 //11
180 //100
181 //101
```

# Experiment: 2

# To implement Arithmetic Coding.

**Scilab code Solution 2.02** arithmetic

```
1  //OS: Windows 7
2  //Scilab Version: Scilab 5.4.1
3  clc;
4  clear all;
5  n=input("Enter the no. of symbols : ");//Input:
       Taking the no. of symbols (ex 5)
6  //Note:The sum of probabilities of all symbols must
       be one(1)
7  for i = 1:n
8      printf("\nEnter the probability(<=1) of symbol
           %d: ",i);//Input: Taking the probability of
           occurence
9  p(i)=input("");
10 end
11 //Sample Input for probability of symbols
12 // Symbol                    Probability
13 //    1                          0.3
```

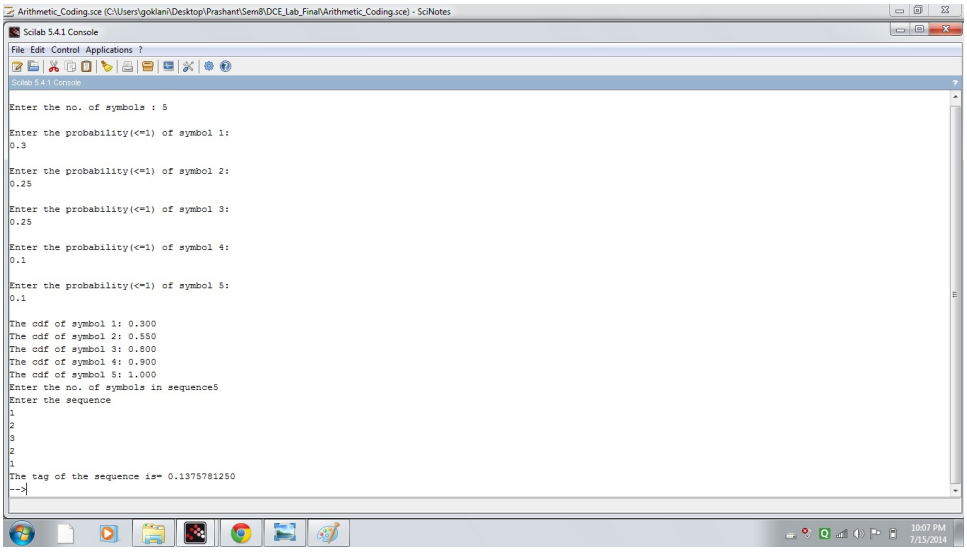


Figure 2.1: arithmetic

```
14 //     2                                    0.25
15 //     3                                    0.25
16 //     4                                    0.1
17 //     5                                    0.1
18 printf("\nThe cdf of symbol 1: %.3f ",p(1));
19 //Output CDF for example input
20 // Symbol                                  CDF
21 //    1                                    0.3
22 //    2                                    0.550
23 //    3                                    0.800
24 //    4                                    0.900
25 //    5                                    1.000
26
27 c(1)=p(1);
28 for i = 2:n
29     c(i)=p(i)+c(i-1);
30     printf("\nThe cdf of symbol %d: ",i);
31     printf("%.3f",c(i));
32 end
33 s=input("Enter the no. of symbols in sequence");//
```

```
      Input: No. of symbols(for ex if the sequence to
      be coded is: 1 2 3 2 1 where 1,2,3...are symbol
      numbers then no. of symbols are 5)
34 //ex No. of symbols in sequence=5
35 printf("Enter the sequence ");//Input: Sequence(For
      example to enter the sequence 1 2 3 2 1, press
      each symbol and then enter. So for our case,
      press 1 and then enter then similarly 2 then
      enter and so on)
36 //Input ex Sequence: 1 (press Enter)
37 //                   2 (press Enter)
38 //                   3 (press Enter)
39 //                   2 (press Enter)
40 //                   1 (press Enter)
41 for j = 1:s
42 b(j)=input("");//Inserting the sequence
43 end
44 //Setting the lower and upper limit for 1st stage
45 if b(1) == 1 then
46 l(1)=0;
47 u(1)=c(b(1));
48 else
49 l(1)=c(b(1)-1);
50 u(1)=c(b(1));
51 end
52 //Calculating lower and upper limits for 2nd stage
      and ahead
53 for k = 2:s
54 if b(k) == 1 then
55 l(k)=l(k-1);
56 u(k)=l(k-1)+((u(k-1)-l(k-1))*c(b(k)));
57 else
58 l(k)=l(k-1)+((u(k-1)-l(k-1))*c(b(k)-1));
59 u(k)=l(k-1)+((u(k-1)-l(k-1))*c(b(k)));
60 end
61 end
62
63 tag=(l(s)+u(s))/2;//Generating tag
```

```
64 printf("The tag of the sequence is= %.10f",tag);//
      Output: The tag of the sequence
65 //Output for ex tag=0.1375781250
```

---

# Experiment: 3

# To implement μ law encoding

**Scilab code Solution 3.03** ULAW

```
1  //OS: Windows 7
2  //Scilab Version: Scilab 5.4.1
3  clc;
4  clear all;
5  n=input("Enter the input sample(no. to be coded) : "
       );//Input: Taking Input Sample from user
6  //Input ex. n=-656
7  if n<0 then
8      P=49;//ascii code for 1=49
9  else
10     P=48;//ascii code for 0=48
11 end
12 Pc=asciimat(P);//Sign Bit
13 printf("The encoded word is : ");//Output: The
       encoded word
14 printf("%c",Pc);
15 s1=abs(n)+33;
16 s1b=dec2bin(s1);
17 l=length(s1b);
18 s1bl=length(s1b)-1;
19 s1bls=s1bl-5;
```

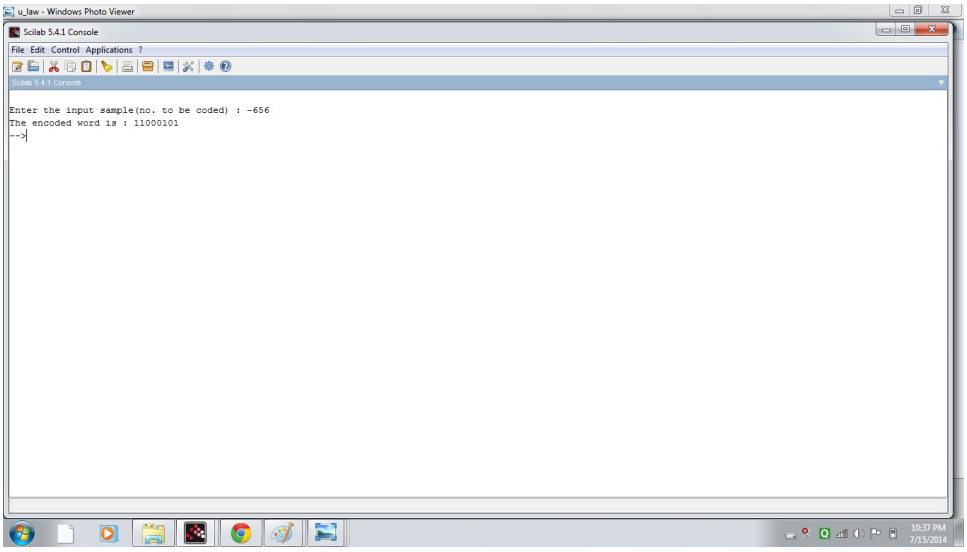


Figure 3.1: ULAW

```
segcod=dec2bin(s1bls,3);//segment code
printf("%s",dec2bin(s1bls,3));
qrev=part(s1b,1:4);
q=strrev(qrev);//quantization code
printf("%s",q);
//Output for ex. 11000101
```

# Experiment: 4

# To implement one dimension DCT

**Scilab code Solution 4.04** 1DDCT

```
1  //OS: Windows 7
2  //Scilab Version: Scilab 5.4.1
3  //one dimensional cosine transform
4  clc;
5  clear all;
6  f=[1 2 4 7];//Input: A row matrix
7  //Input ex. f=[1 2 4 7]
8  N=length(f);//finding length ofinput sequence
9  F=zeros(1,N);//cosine transform of input
10 //C=zeros(N,N);
11 for k=1:N
12     for n=1:N
13         if (k-1)==0
14             C(k,n)=inv(sqrt(N));//cosine transform
                   matrix
15         else
16             C(k,n)=sqrt(2)*inv(sqrt(N))*cos(%pi*(2*(n
                   -1)+1)*(k-1)/(2*N));
17         end
```

```
18      end
19  end
20
21  for u=1:N
22      for x=1:N
23        if (u-1)==0
24             F(u)=F(u)+inv(sqrt(N))*f(x)*cos(%pi*(2*(x
                  -1)+1)*(u-1)/(2*N));
25          else
26              F(u)=F(u)+sqrt(2)*inv(sqrt(N))*f(x)*cos(
                   %pi*(2*(x-1)+1)*(u-1)/(2*N));
27          end
28       end
29  end
30
31  disp(F,"is",f,"Discrete Cosine Transform of");//
       Output: The discrete cosine transform of 1D
       sequence
32  //Output for ex.:[7  -4.460885  1  -0.3170253]
```

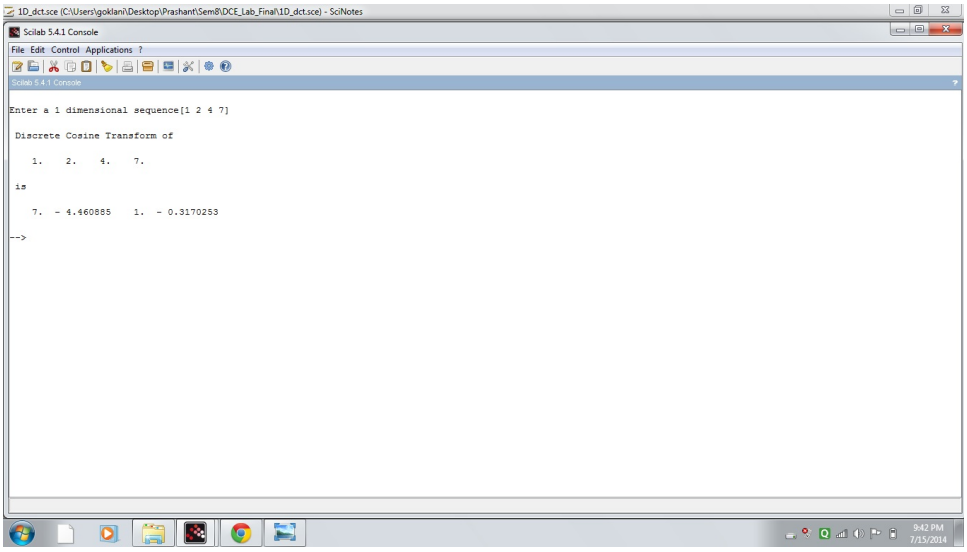


Figure 4.1: 1DDCT

# Experiment: 5

# To implement two dimension DCT

**Scilab code Solution 5.05** 2DDCT

```
1  //OS: Windows 7
2  //Scilab Version: Scilab 5.4.1
3  //two dimensional cosine transform
4  clc;
5  clear all;
6  f=[2 4 4 2;4 6 8 3;2 8 10 4;3 8 6 2];//Input: Enter
       a square matrix
7  //Input ex. f=[2 4 4 2;4 6 8 3;2 8 10 4;3 8 6 2]
8  [M N]=size(f);//finding length of input sequence
9  for k=1:N
10     for n=1:N
11         if (k-1)==0
12            C(k,n)=inv(sqrt(N));//cosine transform
                  matrix
13         else
14            C(k,n)=sqrt(2)*inv(sqrt(N))*cos(%pi*(2*(n
                  -1)+1)*(k-1)/(2*N));
15         end
16     end
```

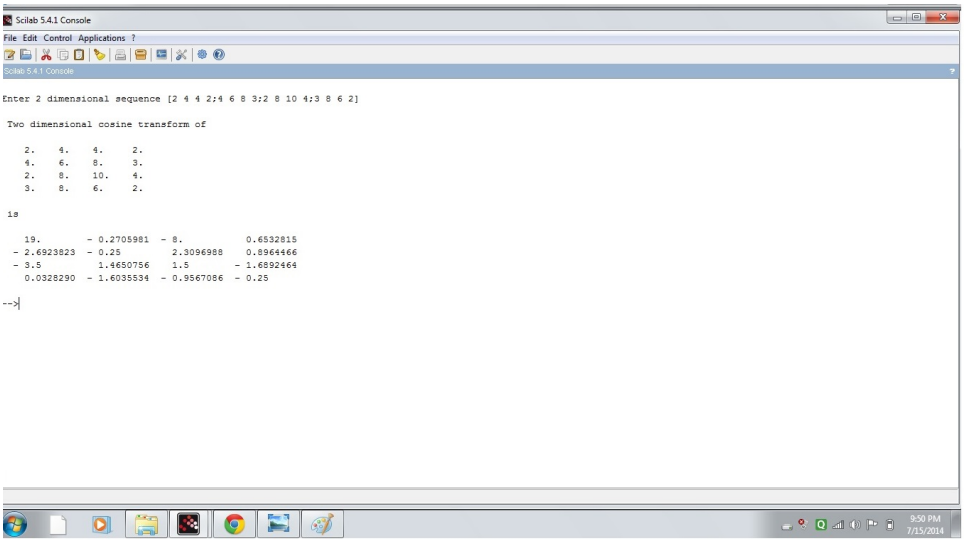


Figure 5.1: 2DDCT

```
17 end
18
19 F=C*f*C';//discrete cosine transform of input for
      non-symmetric DCT
20 disp(F,"is",f,"Two dimensional cosine transform of")
      ;//Output: Two dimensional cosine transform of
      the matrix
21 //Output for ex. :
22 //     19.          - 0.2705981   - 8.
      0.6532815
23 //   - 2.6923823    - 0.25          2.3096988
      0.8964466
24 //   - 3.5          1.4650756     1.5          -
      1.6892464
25 //     0.0328290    - 1.6035534   - 0.9567086   - 0.25
```

# Experiment: 6

# To implement Chinese Remainder Theorem

**Scilab code Solution 6.06** CRT

```
1  //OS: Windows 7
2  //Scilab Version: Scilab 5.4.1
3  clc;
4  clear all;
5  //Standard Equations for CRT
6  //x=a1 mod m1
7  //x=a2 mod m2
8  //x=a3 mod m3
9  //Taking the parameters of standard equation from
       the user
10 m1=[3];//input("Enter the value m1 : ");//Input:
       Value of m1 from equations
11 //Input ex. m1=3
12 m2=[4];//input("Enter the value m2 : ");//Input:
       Value of m2 from equations
13 //Input ex. m2=4
14 m3=[5];//("Enter the value m3 : ");//Input: Value of
```

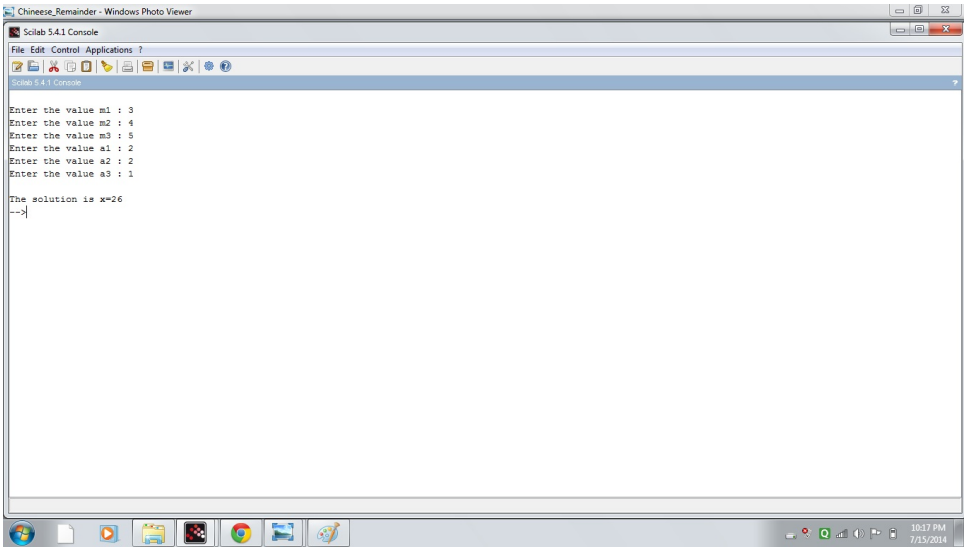


Figure 6.1: CRT

```
    m3 from equations
15 //Input ex. m3=5
16 a1=[2];//input("Enter the value a1 : ");//Input:
    Value of a1 from equations
17 //Input ex. a1=2
18 a2=[2];//input("Enter the value a2 : ");//Input:
    Value of a2 from equations
19 //Input ex. a2=2
20 a3=[1];//("Enter the value a3 : ");//Input: Value of
    a3 from equations
21 //Input ex. a3=1
22 M=m1*m2*m3;
23 M1=M/m1;
24 M2=M/m2;
25 M3=M/m3;
26
27 for i = 1:10
28     if modulo(((M1*i)-1),m1) == 0 then
                                        //Calculating
       M1inverse
```

```
29          M1in=i;
30          break;
31          end
32  end
33  for i = 1:10
34      if modulo(((M2*i)-1),m2) == 0 then
            ..................................//
            Calculating M2inverse
35          M2in=i;
36          break;
37          end
38  end
39  for i = 1:10
40      if modulo(((M3*i)-1),m3) == 0 then
            ..................................//
            Calculating M3inverse
41          M3in=i;
42          break;
43          end
44  end
45  x=modulo(((a1*M1*M1in)+(a2*M2*M2in)+(a3*M3*M3in)),M)
      ;//Solution of equations
46  printf("\nThe solution is x=%d",x);// Output: The
      solution for the set of equations
47  //Output for ex. x=26
```

# Experiment: 7

# To implement Ceaser Cipher Algorithm

**Scilab code Solution 7.07** CCA

```
1  //OS: Windows 7
2  //Scilab Version: Scilab 5.4.1
3  clc;
4  clear all;
5  n=input("Enter the length of the text:");//Input:
       Taking the length of the text to be encoded from
       the user (for ex to encode abcde the length is 5)
6  //Input ex n=5
7  t=tokens(input("Please type string wih a space in
       between alphabets:","string"));//Input: Taking
       the string to be encoded
8  //Inpu ex. t=a b c d e
9  printf("The encrypted string is: ");//Output: The
       encrypted string
10 for i = 1:n
11     c(i)=ascii(t(i))+3;//Caeser Cipher encoding with
           offset=3
```

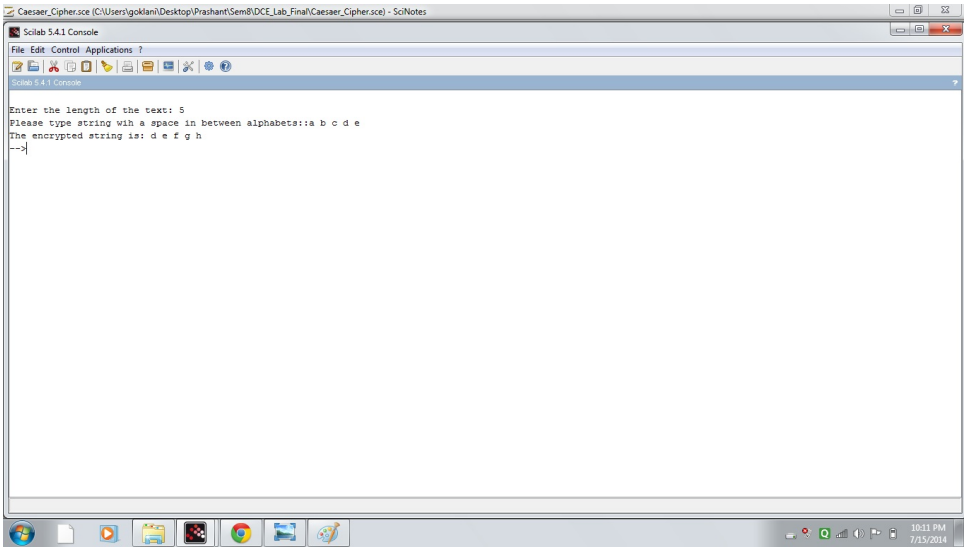


Figure 7.1: CCA

```
12      c1(i)=asciimat(c(i));
13      printf("%.c ",c1(i));//Printing the encoded word
14  end
15  //Output for ex. d e f g h
```

# Experiment: 8

# To implement RSA Algorithm

**Scilab code Solution 8.08** RSA

```
1  //OS: Windows 7
2  //Scilab Version: Scilab 5.4.1
3  clc;
4  clear all;
5  p=input("Enter the 1st prime no.");//Input: Taking
       the first prime no. for RSA
6  //Input ex. p=11
7  q=input("Enter the 2nd prime no.");//Input: Taking
       the second prime no. for RSA
8  //Input ex. q=5
9  n=p*q;
10 phi=(p-1)*(q-1);//Tuotient Function
11 printf("Enter the value of e");//Input: value of e (
       such that phi and the no. entered by you are
       relatively prime)
12 //Input ex. phi=7
13 e=input("");
14 for i = 1:n
15     z=modulo((i*e),phi);
```

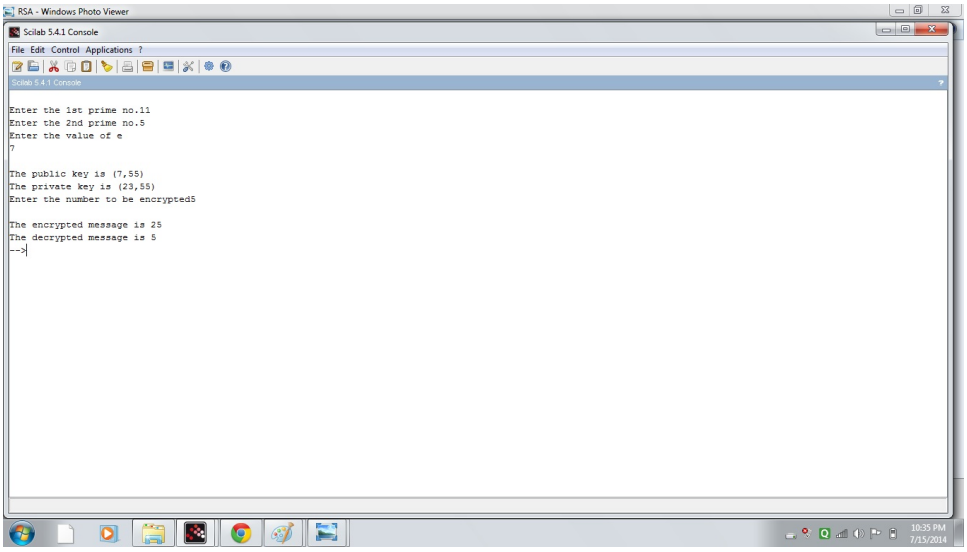


Figure 8.1: RSA

```
16      if z == 1 then
17  break;
18  end
19  end
20   printf("\nThe public key is (%.d",e);//Output: The
        public key is (e,n)
21   //Output for ex. public key (7,55)
22   printf(",%.d)",n);
23    printf("\nThe private key is (%.d",i);//Output:
         The private key is (i,n)
24    //Output ex. private key (23,55)
25   printf(",%.d)",n);
26   m=input("Enter the number to be encrypted");//Input
        : Taking the message to be encrypted
27   //Input ex. 5
28  a=m^e;
29  c=modulo(a,n);
30  printf("\nThe encrypted message is %.d ",c);//Output
      : Printing the encrypted message
31  //Output for ex. 25
```

```
32 b=c^i;
33 t=modulo(b,n);
34 printf("\nThe decrypted message is %.d ",m);//Output
      : Decrypted Message
35 //Output for ex. 5
```

---

# Experiment: 9

# To implement Diffie-Hellman Key exchange

**Scilab code Solution 9.09** DIFFIE

```
1  //OS: Windows 7
2  //Scilab Version: Scilab 5.4.1
3  clc;
4  clear all;
5  p=[13];//input("Enter the common prime number(p) ")
       ;//Input: taking common prime number as input
6  //Input ex. p=13
7  g=[6];//input("Enter the primitive root(g) (any no.)
       ");//Input: taking primitive root as input
8  //Input ex. g=6
9  a=[3];//input("Enter secret key of first user (any
       no.) ");//Input: Taking secret key for user 1
10 //Input ex. a=3
11 b=[10];//input("Enter secret key of second user (any
        no.) ");//Input: Taking secret key for user 2
12 //Input ex. b=10
13 A=modulo(g^a,p);//public key of user 1
14 B=modulo(g^b,p);//public key of user 2
15 common_key=modulo(A^b,p);//common key
```

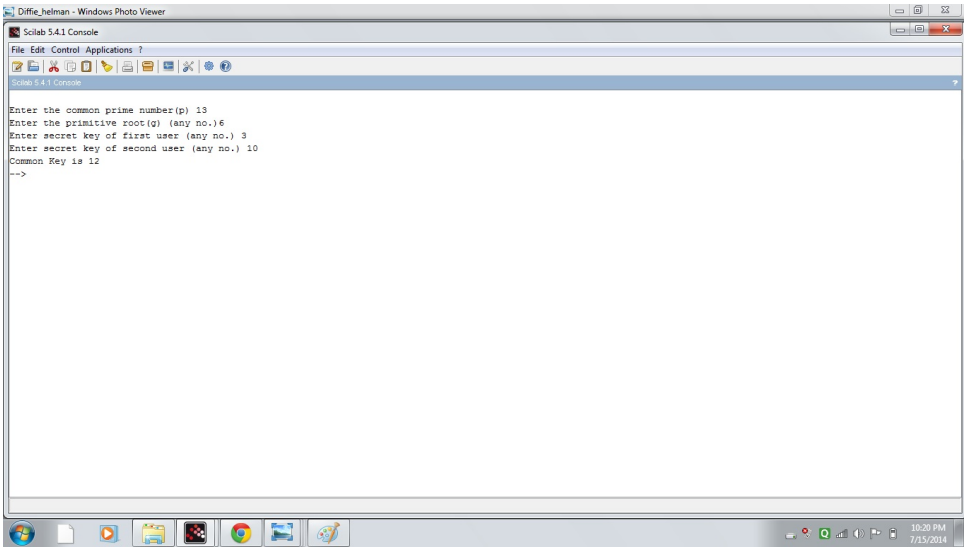


Figure 9.1: DIFFIE

```
16 printf("Common Key is %.d",common_key) ;//Output:
       Produced common key
17 //Output for ex. Common Key=12
```