

G. H. Raison Institute of Engineering and Technology,

Wagholi, Pune

Department of Computer Science Engineering

Computer Network Laboratory

Name : Vaishnavi Bharat Jadhav

Roll No. : COTE017

PRN No. : 71916298L

Subject : Computer Network Laboratory

Subject Code : 310248

Branch : Computer Science Engineering

Class : Third Year (2015 Course)

Subject In Charge : Dr.R.Parvathi

1. A) Program name: Hamming

```
#include <stdio.h>
#include <curses.h>
#include <conio.h>
int main()
{
    int data[10];
    int dataatrec[10], c, c1, c2, c3, i;

    //clrscr();

    printf("Enter 4 bits of data one by one\n");
    scanf ("%d", &data [0]);
    scanf ("%d", &data [1]);
    scanf ("%d", &data [2]);
    scanf ("%d", &data [3]);

    //Calculation of even parity
    data [6] =data [0] ^data [2] ^data [3];
    data [5] =data [0] ^data [1] ^data [3];
    data [4] =data [0] ^data [1] ^data [2];
    printf("\nEncoded data is\n");
    for(i=0;i<7;i++)
    printf("%d",data[i]);
    printf("\n\nEnter received data bits one by one\n");
    for(i=0;i<7;i++)
    scanf("%d",&dataatrec[i]);
    c1=dataatrec[6]^dataatrec[4]^dataatrec[2]^dataatrec[0];
    c2=dataatrec[5]^dataatrec[4]^dataatrec[1]^dataatrec[0];
    c3=dataatrec[3]^dataatrec[2]^dataatrec[1]^dataatrec[0];
    c=c3*4+c2*2+c1 ;
    if(c==0) {
```

```

    printf("\nNo error while transmission of data\n");
}
else {
    printf("\nError on position %d", c);
    printf("\nData sent : ");
    for (i = 0; i < 7; i++)
        printf("%d", data[i]);
    printf("\nData received : ");
    for (i = 0; i < 7; i++)
        printf("%d", dataatrec[i]);
    printf("\nCorrect message is\n");
    //if errorneous bit is 0 we complement it else vice versa
    if (dataatrec[7 - c] == 0)
        dataatrec[7 - c] = 1;
    else
        dataatrec[7 - c] = 0;
    for (i = 0; i < 7; i++)
    {
        printf("%d", dataatrec[i]);
        //return 0;
    }
}
getchar ();
}

```

Code:

```
Run Terminal Help Hamming.c - C\Nprogram - Visual Studio Code
C Hamming.c X C hamming1.c
C Hamming.c > main()
1 #include <stdio.h>
2 //include <conio.h>
3 #include <conio.h>
4 int main()
5 {
6     int data[10];
7     int dataatrec[10], c, c1, c2, c3, i;
8
9     //clrscr();
10
11     printf("Enter 4 bits of data one by one\n");
12     scanf ("%d", &data [0]);
13     scanf ("%d", &data [1]);
14     scanf ("%d", &data [2]);
15     scanf ("%d", &data [3]);
16
17     //Calculation of even parity
18     data [6] =data [0] ^data [2] ^data [3];
19     data [5] =data [0] ^data [1] ^data [3];
20     data [4] =data [0] ^data [1] ^data [2];
21     printf("\nEncoded data is\n");
22     for(i=0;i<7;i++)
23     printf("%d",data[i]);
24     printf("\n\nEnter received data bits one by one\n");
25     for(i=0;i<7;i++)
26     scanf("%d",&dataatrec[i]);
27     c1=dataatrec[6]^dataatrec[4]^dataatrec[2]^dataatrec[0];
28     c2=dataatrec[5]^dataatrec[4]^dataatrec[1]^dataatrec[0];
29     c3=dataatrec[3]^dataatrec[2]^dataatrec[1]^dataatrec[0];
30     c=c3*4+c2*2+c1 ;
31     if(c==0) {
32         printf("\nNo error while transmission of data\n");
33     }
34     else {
35         printf("\nError on position %d", c);
36         printf("\nData sent : ");
37         for (i = 0; i < 7; i++)
38             printf("%d", data[i]);
```

```
Run Terminal Help Hamming.c - C\Nprogram - Visual Studio Code
C Hamming.c X C hamming1.c
C Hamming.c > ...
27     c1=dataatrec[6]^dataatrec[4]^dataatrec[2]^dataatrec[0];
28     c2=dataatrec[5]^dataatrec[4]^dataatrec[1]^dataatrec[0];
29     c3=dataatrec[3]^dataatrec[2]^dataatrec[1]^dataatrec[0];
30     c=c3*4+c2*2+c1 ;
31     if(c==0) {
32         printf("\nNo error while transmission of data\n");
33     }
34     else {
35         printf("\nError on position %d", c);
36         printf("\nData sent : ");
37         for (i = 0; i < 7; i++)
38             printf("%d", data[i]);
39         printf("\nData received : ");
40         for (i = 0; i < 7; i++)
41             printf("%d", dataatrec[i]);
42         printf("\nCorrect message is\n");
43         //if erroneous bit is 0 we complement it else vice versa
44         if (dataatrec[7 - c] == 0)
45             dataatrec[7 - c] = 1;
46         else
47             dataatrec[7 - c] = 0;
48         for (i = 0; i < 7; i++)
49         {
50             printf("%d", dataatrec[i]);
51             //return 0;
52         }
53     }
54     getch ();
55 }

PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE
1: Code
5
Encoded data is
1345670

Enter received data bits one by one
1
3
```

OUTPUT:

```
Run Terminal Help Hamming.c - CNprogram - Visual Studio Code
C Hamming.c X C CRC.cpp
C Hamming.c > main()
1 #include <stdio.h>

PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE 2: Code
Microsoft Windows [Version 10.0.19042.928]
(c) Microsoft Corporation. All rights reserved.

E:\CNprogram>cd "e:\CNprogram\" && gcc Hamming.c -o Hamming && "e:\CNprogram\Hamming
Enter 4 bits of data one by one
1
1
1
1
1

Encoded data is
11111111

Enter received data bits one by one
1
2
1
1
1
1
1

Error on position 18
Data sent : 11111111
Data received : 12111111
Correct message is
12111111
E:\CNprogram>
```

```
Run Terminal Help Hamming.c - CNprogram - Visual Studio Code
C Hamming.c X C hamming1.c
C Hamming.c > main()
20 printf("%d\n",dataatrec[0]);
27 printf("Data received: %d\n",dataatrec[0]);

PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE 1: Code
1345670

Enter received data bits one by one
1
3
4
5
6
7
0

Error on position 21
Data sent : 1345670
Data received : 1345670
Correct message is
1345670
E:\CNprogram>cd "e:\CNprogram\" && gcc Hamming.c -o Hamming && "e:\CNprogram\Hamming
Enter 4 bits of data one by one
1
0
1
0
1
0

Encoded data is
1010010

Enter received data bits one by one
1
0
1
0
0
1
0

No error while transmission of data

E:\CNprogram>
```

B) CRC CODE

```
#include <iostream>
using namespace std;
int main()
{
    int i, j, k, l;
    //Get Frame
    int fs;
    cout << "\n Enter Size of data: ";
    cin >> fs;
    int f[20];
    cout << " Enter data:";
    for (i = 0; i < fs; i++)
    {
        cin >> f[i];
    }
    //Get Generator
    int gs;
    cout << "\n Enter key size: ";
    cin >> gs;
    int g[20];
    cout << "\n Enter key:";
    for (i = 0; i < gs; i++)
    {
        cin >> g[i];
    }
    cout << "\n Sender Side:";
    cout << "\n data: ";
    for (i = 0; i < fs; i++)
    {
        cout << f[i];
    }
    cout << "\n key :";
    for (i = 0; i < gs; i++)
```

```

{
    cout << g[i];
}
//Append 0's
int rs = gs;
cout << "\n Number of 0's to be appended: " << rs;
for (i = fs; i < fs + rs; i++)
{
    f[i] = 0;
}
int temp[20];
for (i = 0; i < 20; i++)
{
    temp[i] = f[i];
}
cout << "\n Message after appending 0's :";
for (i = 0; i < fs + rs; i++)
{
    cout << temp[i];
}
//Division
for (i = 0; i < fs; i++)
{
    j = 0;
    k = i;
    //check whether it is divisible or not
    if (temp[k] >= g[j])
    {
        for (j = 0, k = i; j < gs; j++, k++)
        {
            if ((temp[k] == 1 && g[j] == 1) || (temp[k] == 0 && g[j]
== 0))
            {
                temp[k] = 0;
            }
        }
    }
}

```

```

        else
        {
            temp[k] = 1;
        }
    }
}
//CRC
int crc[15];
for (i = 0, j = fs; i < rs; i++, j++)
{
    crc[i] = temp[j];
}
cout << "\n CRC bits: ";
for (i = 0; i < rs; i++)
{
    cout << crc[i];
}
cout << "\n Transmitted Frame: ";
int tf[15];
for (i = 0; i < fs; i++)
{
    tf[i] = f[i];
}
for (i = fs, j = 0; i < fs + rs; i++, j++)
{
    tf[i] = crc[j];
}
for (i = 0; i < fs + rs; i++)
{
    cout << tf[i];
}
cout << "\n Receiver side : ";
cout << "\n Received Frame: ";
for (i = 0; i < fs + rs; i++)

```



```

{
    cout << tf[i];
}
for (i = 0; i < fs + rs; i++)
{
    temp[i] = tf[i];
}
//Division
for (i = 0; i < fs + rs; i++)
{
    j = 0;
    k = i;
    if (temp[k] >= g[j])
    {
        for (j = 0, k = i; j < gs; j++, k++)
        {
            if ((temp[k] == 1 && g[j] == 1) || (temp[k] == 0 && g[j]
== 0))
            {
                temp[k] = 0;
            }
            else
            {
                temp[k] = 1;
            }
        }
    }
}
cout << "\n Reminder: ";
int rrem[15];
for (i = fs, j = 0; i < fs + rs; i++, j++)
{
    rrem[j] = temp[i];
}
for (i = 0; i < rs; i++)

```

```

{
    cout << rrem[i];
}
int flag = 0;
for (i = 0; i < rs; i++)
{
    if (rrem[i] != 0)
    {
        flag = 1;
    }
}
if (flag == 0)
{
    cout << "\n Since Remainder Is 0 Hence Message
Transmitted From Sender To Receriver Is Correct";
}
else
{
    cout << "\n Since Remainder Is Not 0 Hence Message
Transmitted From Sender To Receriver Contains Error";
}
return 0;
}

```

CODE:

```
Run Terminal Help • CRC.cpp - CNprogram - Visual Studio Code
C Hamming.c C++ CRC.cpp
C++ CRC.cpp > main()
1 #include <iostream>
2 using namespace std;
3 int main()
4 {
5     int i, j, k, l;
6     //Get Frame
7     int fs;
8     cout << "\n Enter Size of data: ";
9     cin >> fs;
10    int f[20];
11    cout << " Enter data:";
12    for (i = 0; i < fs; i++)
13    {
14        cin >> f[i];
15    }
16    //Get Generator
17    int gs;
18    cout << "\n Enter key size: ";
19    cin >> gs;
20    int g[20];
21    cout << "\n Enter key:";
22    for (i = 0; i < gs; i++)
23    {
24        cin >> g[i];
25    }
26    cout << "\n Sender Side:";
27    cout << "\n data: ";
28    for (i = 0; i < fs; i++)
29    {
30        cout << f[i];
31    }
32    cout << "\n key :";
33    for (i = 0; i < gs; i++)
34    {
35        cout << g[i];
36    }
37    //Append 0's
38    int rs = gs;
```

```
Run Terminal Help • CRC.cpp - CNprogram - Visual Studio Code
C Hamming.c C++ CRC.cpp
C++ CRC.cpp > main()
37 //Append 0's
38 int rs = gs;
39 cout << "\n Number of 0's to be appended: " << rs;
40 for (i = fs; i < fs + rs; i++)
41 {
42     f[i] = 0;
43 }
44 int temp[20];
45 for (i = 0; i < 20; i++)
46 {
47     temp[i] = f[i];
48 }
49 cout << "\n Message after appending 0's :";
50 for (i = 0; i < fs + rs; i++)
51 {
52     cout << temp[i];
53 }
54 //Division
55 for (i = 0; i < fs; i++)
56 {
57     j = 0;
58     k = i;
59     //check whether it is divisible or not
60     if (temp[k] >= g[j])
61     {
62         for (j = 0, k = i; j < gs; j++, k++)
63         {
64             if ((temp[k] == 1 && g[j] == 1) || (temp[k] == 0 && g[j] == 0))
65             {
66                 temp[k] = 0;
67             }
68             else
69             {
70                 temp[k] = 1;
71             }
72         }
73     }
74 }
```

Ln 17, Col 12 Spaces: 4 UTF-8 CRLF C++ Win32

```
Run Terminal Help • CRC.cpp - CNprogram - Visual Studio Code
C Hamming.c C++ CRC.cpp
C++ CRC.cpp > main()
72     }
73     }
74 }
75 //CRC
76 int crc[15];
77 for (i = 0, j = fs; i < rs; i++, j++)
78 {
79     crc[i] = temp[j];
80 }
81 cout << "\n CRC bits: ";
82 for (i = 0; i < rs; i++)
83 {
84     cout << crc[i];
85 }
86 cout << "\n Transmitted Frame: ";
87 int tf[15];
88 for (i = 0; i < fs; i++)
89 {
90     tf[i] = f[i];
91 }
92 for (i = fs, j = 0; i < fs + rs; i++, j++)
93 {
94     tf[i] = crc[j];
95 }
96 for (i = 0; i < fs + rs; i++)
97 {
98     cout << tf[i];
99 }
100 cout << "\n Receiver side : ";
101 cout << "\n Received Frame: ";
102 for (i = 0; i < fs + rs; i++)
103 {
104     cout << tf[i];
105 }
106 for (i = 0; i < fs + rs; i++)
107 {
108     temp[i] = tf[i];
```

```
Run Terminal Help • CRC.cpp - CNprogram - Visual Studio Code
C Hamming.c C++ CRC.cpp
C++ CRC.cpp > ...
128     }
129 }
130 cout << "\n Reminder: ";
131 int rrem[15];
132 for (i = fs, j = 0; i < fs + rs; i++, j++)
133 {
134     rrem[j] = temp[i];
135 }
136 for (i = 0; i < rs; i++)
137 {
138     cout << rrem[i];
139 }
140 int flag = 0;
141 for (i = 0; i < rs; i++)
142 {
143     if (rrem[i] != 0)
144     {
145         flag = 1;
146     }
147 }
148 if (flag == 0)
149 {
150     cout << "\n Since Remainder Is 0 Hence Message Transmitted From Sender To Receiver Is Correct";
151 }
152 else
153 {
154     cout << "\n Since Remainder Is Not 0 Hence Message Transmitted From Sender To Receiver Contains Error";
155 }
156 return 0;
157 }
158
```

OUTPUT:

```
Run Terminal Help Hamming.c - CNprogram - Visual Studio Code
C Hamming.c x C++ CRC.cpp
C Hamming.c > main()
1 #include <stdio.h>

PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE 2: Code
Microsoft Windows [Version 10.0.19042.928]
(c) Microsoft Corporation. All rights reserved.

E:\CNprogram>cd "e:\CNprogram\" && gcc Hamming.c -o Hamming && "e:\CNprogram\"Hamming
Enter 4 bits of data one by one
1
1
1
1

Encoded data is
11111111

Enter received data bits one by one
1
2
1
1
1
1
1

1

Error on position 18
Data sent : 11111111
Data received : 12111111
Correct message is
12111111
E:\CNprogram>
```