Name: Nishant Kumar Giri

Roll No: AC-1254

Course Name: BSc (Hons) Computer Science

Paper Name: Computer Networks Practical Exam (2021 – 22)

Semester: 3rd

CODE

```
/*
Write the program Hamming code for a given
input message (sender side and receiver side).
*/

#include <iostream>
#include <cmath>
using namespace std;
int main()
{
    cout << "----------SENDER SIDE----------";
    int m, r;
    do
    {
        cout << "\nEnter no. of data bits: ";
        cin >> m;
    } while (m <= 0);

    int message[m]; //Original Message
    cout << "\nEnter the binary message: ";
    for (int i = m - 1; i >= 0; i--)
        cin >> message[i];

    for (int i = 2; i; i++) //To Calculate no. of redundant bits!
    {
        if ((int)pow(2, i) >= (m + i + 1)) //2^r >= m+r+1, pow() returns a
double value!
```

```
        {
            r = i;
            break;
        }
    }

    int sentData[m + r];
    for (int i = 0, x = 0, j = 0; j < m + r; j++) //Initialising the data string
with the original message at position of the data bits!
    {
        if ((int)pow(2, i) != j + 1)
            sentData[j] = message[x++];
        else
            i++;
    }

    for (int i = 0; i < r; i++) //To calculate the even parity of the parity
bits and initialising them into the data string!
    {
        sentData[(int)pow(2, i) - 1] = 0;
        for (int j = 1; j <= m + r; j++) //Calculating even parity of the ith
parity bit!
            if ((j & (int)pow(2, i)) == (int)pow(2, i))
                sentData[(int)pow(2, i) - 1] = sentData[(int)pow(2, i) - 1] ^
sentData[j - 1];
    }

    cout << "\nMessage Sent by Sender: ";
    for (int i = (m + r - 1); i >= 0; i--) //Printing the message sent by
sender!
        cout << sentData[i] << " ";
    cout << "\n";

    cout << "\n----------RECIEVER SIDE----------";
    int recievedData[m + r];
    cout << "\nEnter Recieved Message: "; //Retrieving the recieved message from
user!
    for (int i = m + r - 1; i >= 0; i--)
        cin >> recievedData[i];

    int c = 0;
```

```
    for (int i = 0; i < r; i++) //To check For Errors in the recieved message!
    {
        int p = recievedData[(int)pow(2, i) - 1];
        for (int j = 1; j <= m + r; j++)
            if (j != (int)pow(2, i) && (j & (int)pow(2, i)) == (int)pow(2, i))
                p = p ^ recievedData[j - 1];

        c += (int)pow(2, i) * p;
    }

    if (c != 0) //Printing the corrected recieved message in case of any error!
    {
        cout << "\nError in bit " << c << "!";
        recievedData[c - 1] = !recievedData[c - 1];
        cout << "\nThe Corrected Recieved Message: ";
        for (int j = m + r - 1; j >= 0; j--)
            cout << recievedData[j] << " ";
        cout << "\n";
    }
    else
        cout << "\nNo Errors Found in the Recieved Message!\n";
    return 0;
}
```

OUTPUT

```
PS C:\Users\nisha\Desktop\CNPracticals> g++ .\hammingCode.cpp -o .\hammingCode.exe
PS C:\Users\nisha\Desktop\CNPracticals> .\hammingCode.exe
----------SENDER SIDE----------
Enter no. of data bits: 4

Enter the binary message: 1 0 1 0

Message Sent by Sender: 1 0 1 0 0 1 0

----------RECIEVER SIDE----------
Enter Recieved Message: 1 1 1 0 0 1 0

Error in bit 6!
The Corrected Recieved Message: 1 0 1 0 0 1 0
PS C:\Users\nisha\Desktop\CNPracticals>
```

Code

```
/*
Write the program IP addressing by taking the
IP address as input and print the corresponding
class, Net –ID and Host-ID.
*/

#include <iostream>

using namespace std;

int main()
{

    int arr[4];
    cout << "Please enter the IP Address\n";
    for (int i = 0; i < 4; i++)
        cin >> arr[i];

    cout << "\nYou Have Entered\n";
    for (int i = 0; i < 4; i++)
    {
        cout << arr[i];
        if (i < 3)
        {
            cout << ".";
        }
    }
    cout << endl;

    cout << "\nDetermining Class\n";

    if ((arr[0] > 0) && (arr[0] <= 127))
    {
        cout << "class A\n";

        cout << "\n\nNet-ID\n";
        for (int i = 0; i < 3; i++)
        {
```

```
            cout << arr[i];
            if (i < 2)
                cout << ".";
        }
        cout << "\n\nHost-ID\n";
        cout << arr[3];
    }

    if ((arr[0] > 127) && (arr[0] <= 191))
    {
        cout << "class B\n";

        cout << "\n\nNet-ID\n";
        for (int i = 0; i < 2; i++)
            cout << arr[i] << ".";

        cout << "\n\nHost-ID\n";
        cout << arr[2] << "." << arr[3];

        cout << endl;
    }

    if ((arr[0] > 191) && (arr[0] <= 223))
    {
        cout << "class C\n";

        cout << "\n\nNet-ID\n";
        for (int i = 0; i < 3; i++)
            cout << arr[i] << ".";

        cout << "\n\nHost-ID\n";
        cout << arr[3];

        cout << endl;
    }

    if ((arr[0] > 223) && (arr[0] <= 239))
    {
        cout << "class D\n";
        cout << "No Net-ID And Host-ID\n";
    }
```

```
    if ((arr[0] > 239) && (arr[0] <= 255))
    {
        cout << "class E\n";

        cout << "No Net-ID And Host-ID\n";
    }
}
```

OUTPUT

```
PS C:\Users\nisha\Desktop\CNPracticals> g++ .\IP.cpp -o .\IP.exe
PS C:\Users\nisha\Desktop\CNPracticals> .\IP.exe
Please enter the IP Address
191 255 255 255

You Have Entered
191.255.255.255

Determining Class
class B


Net-ID
191.255.

Host-ID
255.255
PS C:\Users\nisha\Desktop\CNPracticals>
```