Computer Networks Laboratory
(CSDC-0236)

B.Tech IVth Semester
(January – June 2025)

Submitted by

Ritik Gupta(23103122)
Group-G2

Submitted to

Dr. Samayveer Singh

Department of Computer Science & Engineering
Dr. B. R. Ambedkar National Institute of Technology Jalandhar - 144008, Punjab, India

ASSIGNMENT – 05

File: 1.c++ (Page 1/3)

```
#include<iostream>
#include<string>
#include<vector>
using namespace std;
vector<int>complement(vector<int>num){
    for(int i=0;i<num.size();i++){
        num[i]=num[i]==0?1:0;
    }
    return num;
}
vector<int>sumF(vector<int>f1,vector<int>f2){
    int n=f1.size();
    int carry=0;
    vector<int>sumX(n,0);

    for(int i=n-1;i>=0;i--){
        sumX[i]=f1[i]+f2[i]+carry;
        if(sumX[i]>1){
            sumX[i]=0;
            carry=1;
        }else{
            carry=0;
        }
    }

    for(int i=n-1;i>=0;i--){
        sumX[i]=sumX[i]+carry;
        if(sumX[i]>1){
            sumX[i]=0;
            carry=1;
        }else{
            carry=0;
        }
    }

    return sumX;
```

1.c++ (continued, Page 2/3)

```
}
vector<int>toVector(string str){
    vector<int>vec;

    for(int i=0;i<str.size();i++){
        vec.push_back(int(str[i]-'0'));
    }
    return vec;
}
vector<int> getCheckSum(vector<string>data){
    int n=data.size();
    vector<int>f;
    vector<int>sumX=toVector(data[0]);

    for(int i=1;i<n;i++){
        f=toVector(data[i]);
        sumX=sumF(sumX,f);
    }
    sumX=complement(sumX);
    return sumX;
}
int main(){
    int n;
    vector<string>data;

    cout<<"Enter number of frames :";
    cin>>n;

    while(n>0){
        cout<<"Enter Frame :";
        string f;
        getline(cin,f,',');
        data.push_back(f);
        n--;
    }
    vector<int>checkSum=getCheckSum(data);
    cout<<"CheckSum :";
```

1.c++ (continued, Page 3/3)

```
    for(int i=0;i<checkSum.size();i++){
        cout<<checkSum[i];
    }
    cout<<endl;
    return 0;
}
```

Output of 1.c++ (Page 1/1)

```
Unsupported language.
```

File: checksum.c++ (Page 1/1)

```
#include <iostream>
using namespace std;
string  complement(string bin){
    for(char &i :bin){
        i=i==0?1:0;} }
int main() {
    int  data[] = {
        11,15,13,12
    };
int sum =0;
    for(int i=0;i<4;i++){
        sum = sum + data[i];}
    int bin_sum;
    while(sum!=0){
        int dig =  sum%2;
        bin_sum = dig + bin_sum*2;
        sum = sum/2;}
    string temp = to_string(bin_sum);

    return 0;
}
```

Output of checksum.c++ (Page 1/1)

```
Unsupported language.
```

File: crc.c++ (Page 1/2)

```
#include <iostream>
using namespace std;
void xorOperation(char* dividend, const char* divisor, int
  divisorLength) {
    for (int i = 0; i < divisorLength; i++) {
        dividend[i] = (dividend[i] == divisor[i]) ? '0' : '1';
    }
}
string computeCRC(string data, string divisor) {
    int dataLength = data.length();
    int divisorLength = divisor.length();
    string dividend = data + string(divisorLength - 1, '0');
    for (int i = 0; i <= dataLength - 1; i++) {
        if (dividend[i] == '1') {
            xorOperation(&dividend[i], divisor.c_str(), divisorLength);
        }
    }
    return dividend.substr(dataLength, divisorLength - 1);
}
int main() {
    string data = "11010011101100";
    string divisor = "1011";

    cout << "Original Data: " << data << endl;
    cout << "Divisor: " << divisor << endl;

    string crc = computeCRC(data, divisor);
    cout << "CRC Checksum: " << crc << endl;

    string transmittedData = data + crc;
    cout << "Transmitted Data: " << transmittedData << endl;

    string remainder = computeCRC(transmittedData, divisor);
    if (remainder.find('1') == string::npos) {
        cout << "No error detected in received data."<<endl;
    } else {
        cout <<"Error detected in received data!"<<endl;
```

crc.c++ (continued, Page 2/2)

```
    }

    return 0;
}
```

Output of crc.c++ (Page 1/1)

Unsupported language.

File: hamming_code.c++ (Page 1/2)

```
#include <iostream>
using namespace std;
int getParityBitsCount(int dataBits) {
    int r = 0;
    while ((1 << r) < (dataBits + r + 1)) {
        r++;
    }
    return r;
}
void generateHammingCode(string data) {
    int dataBits = data.length();
    int parityBits = getParityBitsCount(dataBits);
    int totalBits = dataBits + parityBits;
    char hammingCode[totalBits + 1];
    int j = 0;
    for (int i = 1; i <= totalBits; i++) {
        if ((i & (i - 1)) == 0) {
            hammingCode[i] = '0';
        } else {
            hammingCode[i] = data[j++];
        }
    }
    for (int i = 0; i < parityBits; i++) {
        int position = (1 << i);
        int parity = 0;
        for (int k = position; k <= totalBits; k++) {
            if (k & position) {
                parity ^= (hammingCode[k] - '0');
            }
        }
        hammingCode[position] = parity + '0';
    }
    cout << "Hamming Code: ";
    for (int i = 1; i <= totalBits; i++) {
        cout << hammingCode[i];
    }
    cout << endl;
```

hamming_code.c++ (continued, Page 2/2)

```
}
int main() {
    string data = "1011";
    cout << "Original Data: " << data << endl;
    generateHammingCode(data);
    return 0;
}
```

Output of hamming_code.c++ (Page 1/1)

Unsupported language.

File: lrc.c++ (Page 1/1)

```
#include <iostream>
using namespace std;
int main() {
    int rows = 4, cols = 4;
    int data[4][4] = {
        {1, 0, 1, 1},
        {0, 1, 0, 1},
        {1, 1, 1, 0},
        {0, 0, 1, 1}
    };
    int lrc[4] = {0, 0, 0, 0};
    for (int j = 0; j < cols; j++) {
        for (int i = 0; i < rows; i++) {
            lrc[j] = lrc[j]^data[i][j];
        }
    }
    cout << "Data Block:\n";
    for (int i = 0; i < rows; i++) {
        for (int j = 0; j < cols; j++) {
            cout << data[i][j] << " ";
        }
        cout << endl;
    }
    cout << "LRC Row: ";
    for (int j = 0; j < cols; j++) {
        cout << lrc[j] << " ";
    }
    cout << endl;

    return 0;
}
```

Output of lrc.c++ (Page 1/1)

```
Unsupported language.
```