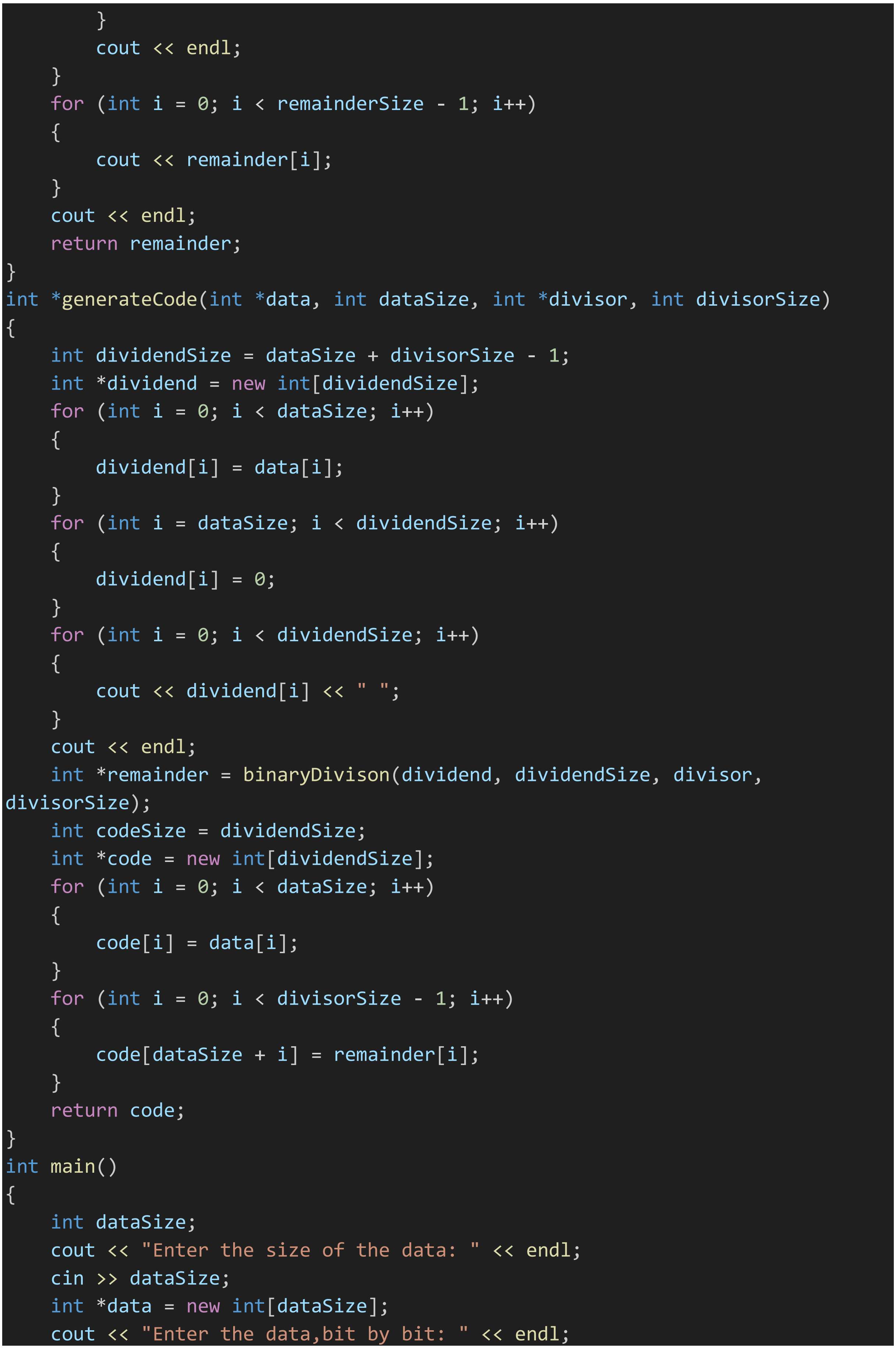
Lab Experiment 3

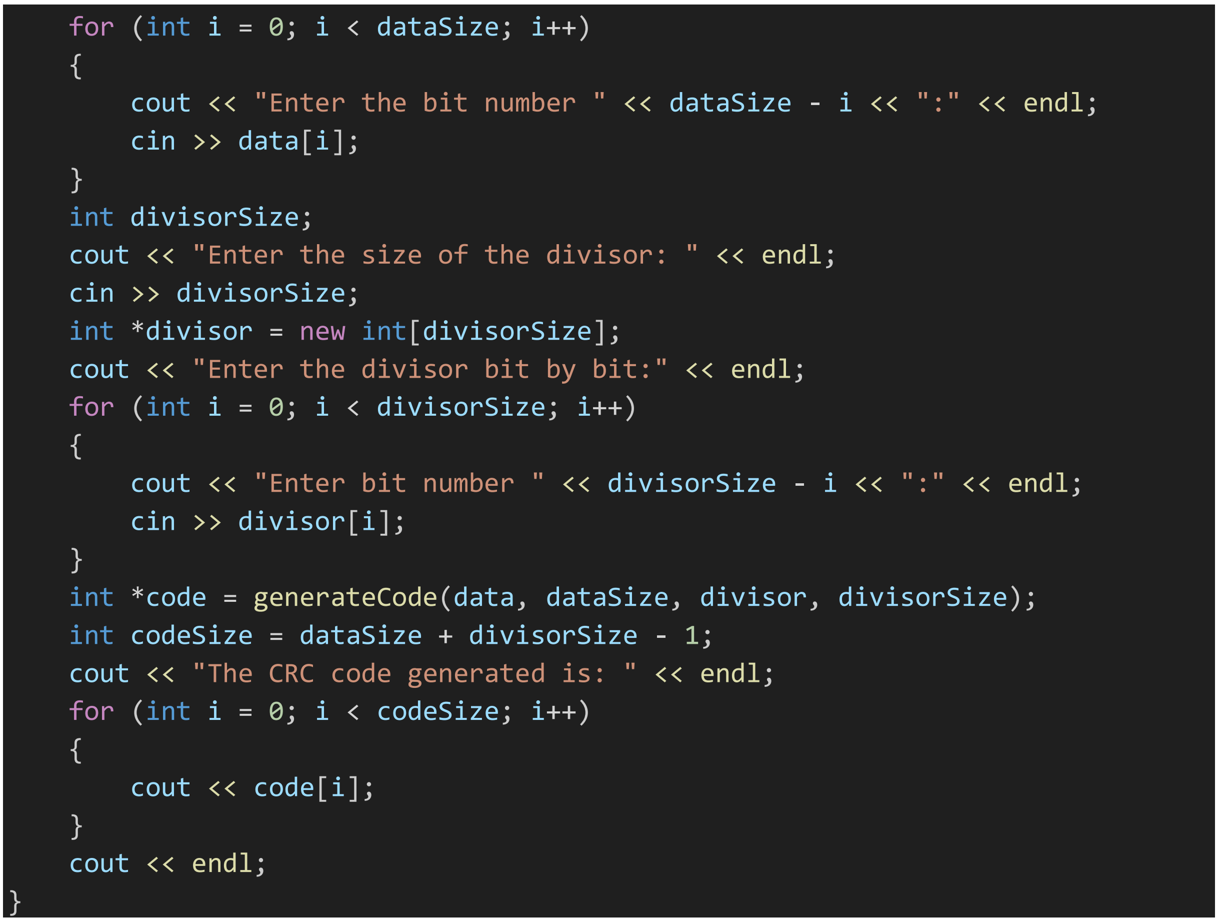
1. Write a program to implement Error Detection Technique using CRC Algorithm.

Program :

```
#include <iostream>
using namespace std;
int *binaryDivison(int *dividend, int dividendSize, int *divisor, int
divisorSize)
{
    int remainderSize = divisorSize;
    int *remainder = new int[remainderSize];
    for (int i = 0; i < remainderSize; i++)
    {
        remainder[i] = dividend[i];
    }
    for (int i = 0; i < remainderSize; i++)
    {
        cout << remainder[i];
    }
    cout << endl;
    for (int i = 0; i < dividendSize - divisorSize + 1; i++)
    {
        cout << i + 1 << ".) First data bit is : " << remainder[0] << endl;
        if (remainder[0] == 1)
        {
            for (int i = 1; i < remainderSize; i++)
            {
                remainder[i - 1] = remainder[i] ^ divisor[i];
            }
        }
        else
        {
            for (int i = 1; i < remainderSize; i++)
            {
                remainder[i - 1] = remainder[i] ^ 0;
            }
        }
        if (i + remainderSize < dividendSize)
        {
            remainder[remainderSize - 1] = dividend[i + remainderSize];
        }
        cout << "Remainder : ";
        for (int i = 0; i < remainderSize; i++)
        {
            cout << remainder[i];
```

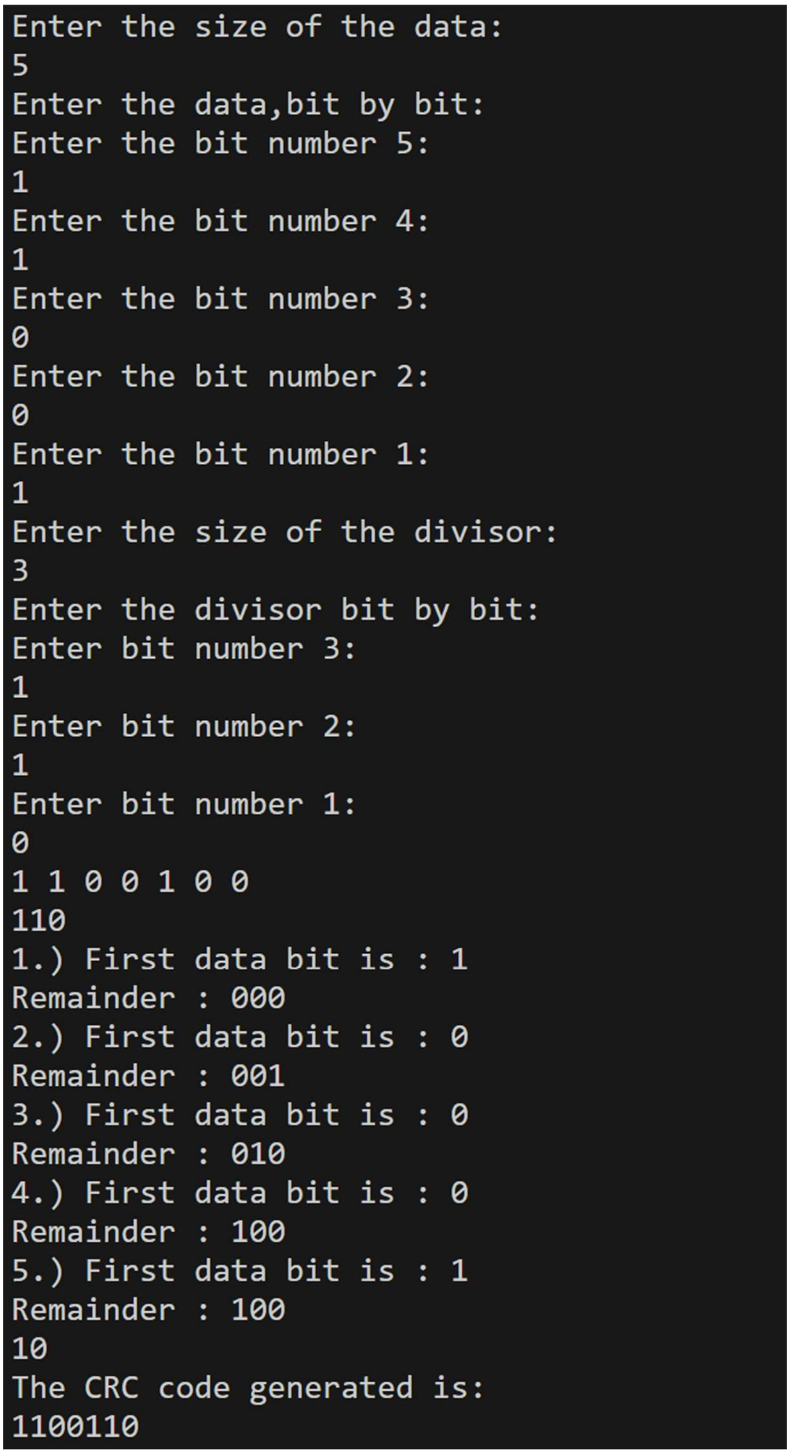


```
        }
        cout << endl;
    }
    for (int i = 0; i < remainderSize - 1; i++)
    {
        cout << remainder[i];
    }
    cout << endl;
    return remainder;
}
int *generateCode(int *data, int dataSize, int *divisor, int divisorSize)
{
    int dividendSize = dataSize + divisorSize - 1;
    int *dividend = new int[dividendSize];
    for (int i = 0; i < dataSize; i++)
    {
        dividend[i] = data[i];
    }
    for (int i = dataSize; i < dividendSize; i++)
    {
        dividend[i] = 0;
    }
    for (int i = 0; i < dividendSize; i++)
    {
        cout << dividend[i] << " ";
    }
    cout << endl;
    int *remainder = binaryDivison(dividend, dividendSize, divisor,
divisorSize);
    int codeSize = dividendSize;
    int *code = new int[dividendSize];
    for (int i = 0; i < dataSize; i++)
    {
        code[i] = data[i];
    }
    for (int i = 0; i < divisorSize - 1; i++)
    {
        code[dataSize + i] = remainder[i];
    }
    return code;
}
int main()
{
    int dataSize;
    cout << "Enter the size of the data: " << endl;
    cin >> dataSize;
    int *data = new int[dataSize];
    cout << "Enter the data,bit by bit: " << endl;
```



```cpp
    for (int i = 0; i < dataSize; i++)
    {
        cout << "Enter the bit number " << dataSize - i << ":" << endl;
        cin >> data[i];
    }
    int divisorSize;
    cout << "Enter the size of the divisor: " << endl;
    cin >> divisorSize;
    int *divisor = new int[divisorSize];
    cout << "Enter the divisor bit by bit:" << endl;
    for (int i = 0; i < divisorSize; i++)
    {
        cout << "Enter bit number " << divisorSize - i << ":" << endl;
        cin >> divisor[i];
    }
    int *code = generateCode(data, dataSize, divisor, divisorSize);
    int codeSize = dataSize + divisorSize - 1;
    cout << "The CRC code generated is: " << endl;
    for (int i = 0; i < codeSize; i++)
    {
        cout << code[i];
    }
    cout << endl;
}
```

Output:



```
Enter the size of the data:
5
Enter the data,bit by bit:
Enter the bit number 5:
1
Enter the bit number 4:
1
Enter the bit number 3:
0
Enter the bit number 2:
0
Enter the bit number 1:
1
Enter the size of the divisor:
3
Enter the divisor bit by bit:
Enter bit number 3:
1
Enter bit number 2:
1
Enter bit number 1:
0
1 1 0 0 1 0 0
110
1.) First data bit is : 1
Remainder : 000
2.) First data bit is : 0
Remainder : 001
3.) First data bit is : 0
Remainder : 010
4.) First data bit is : 0
Remainder : 100
5.) First data bit is : 1
Remainder : 100
10
The CRC code generated is:
1100110
```