PROGRAM : Check Sum - Sender Side Algo…

```
#include <iostream>
#include <cstring>
#define SEG_LEN 8
using namespace std;

void invert(string &a) {
    for(int i=0; i<SEG_LEN; i++) {
        a[i] = (a[i] == '0')? '1' : '0';
    }
}
char add(char bitA, char bitB, char& carry)
{
    if(carry == '1') {
        if(bitB == '1' && bitA == '1') {
            carry = '1';
            return '1';
        }
        else if(bitA == '0' && bitB == '0') {
            carry = '0';
            return '1';
        }
        else if(bitA != bitB) {
            carry = '1';
            return '0';
        }
    } else {
        if(bitB == '1' && bitA == '1') {
            carry = '1';
            return '0';
        }
        else if(bitA == '0' && bitB == '0') {
            carry = '0';
            return '0';
        }
        else {
            carry = '0';
            return '1';
        }
    }
}
void addComplement(string &result, char carry, int n) {
    for(int i=n-1; i>=0; i--) {
        result[i] = add(result[i], '0', carry);
    }
}
void checksum(string data) {
    int n = SEG_LEN;
    int m = data.length() / SEG_LEN;
    string result = data.substr(0, SEG_LEN);
    char carry = '0';

    for(int j=1; j<=m-1; j++) {
```

```
        string s = data.substr(j * SEG_LEN, SEG_LEN);
        for(int i=n-1; i>=0; i--)
        {
            result[i] = add(result[i], s[i], carry);
            if(i == 0 && carry == '1') {
                addComplement(result, carry, n);
            }
        }
        carry = '0';
    }
    invert(result);
    cout << "\nData     : " << data << endl;
    data += result;
    cout << "Checksum : " << result << endl;
    cout << "Codeword : " << data;
}

int main() {
    string s;
    cout << "Enter the data : ";
    cin >> s;
    checksum(s);
    return 0;
}
```

OUTPUT :

```
Enter the data : 10110011101010110101101011010101

Data     : 10110011101010110101101011010101
Checksum : 01110000
Codeword : 10110011101010110101101011010101110000
```

PROGRAM : Check Sum - Receiver Side Algo…

```
#include <iostream>
#include <cstring>
#define SEG_LEN 8
using namespace std;

void invert(string &a) {
    for(int i=0; i<SEG_LEN; i++) {
        a[i] = (a[i] == '0')? '1' : '0';
    }
}
char add(char bitA, char bitB, char& carry)
{
    if(carry == '1') {
        if(bitB == '1' && bitA == '1') {
            carry = '1';
            return '1';
        }
        else if(bitA == '0' && bitB == '0') {
            carry = '0';
            return '1';
        }
        else if(bitA != bitB) {
            carry = '1';
            return '0';
        }
    } else {
        if(bitB == '1' && bitA == '1') {
            carry = '1';
            return '0';
        }
        else if(bitA == '0' && bitB == '0') {
            carry = '0';
            return '0';
        }
        else {
            carry = '0';
            return '1';
        }
    }
}
void addComplement(string &result, char carry, int n) {
    for(int i=n-1; i>=0; i--) {
        result[i] = add(result[i], '0', carry);
    }
}
void checksum(string data) {
    int n = SEG_LEN;
    int m = data.length() / SEG_LEN;
    string result = data.substr(0, SEG_LEN);
    char carry = '0';

    for(int j=1; j<=m-1; j++) {
        string s = data.substr(j * SEG_LEN, SEG_LEN);
```

```
            for(int i=n-1; i>=0; i--)
            {
                result[i] = add(result[i], s[i], carry);
                if(i == 0 && carry == '1') {
                    addComplement(result, carry, n);
                }
            }
            carry = '0';
        }
        invert(result);
        for(int i = 0; i < SEG_LEN; i++) {
            if(result[i] == '1') {
                cout << "\nError in the Data!";
                return;
            }
        }
        cout << "\nNo error in received data!\n";
        cout << "Codeword : " << data;
        data = data.substr(0, data.length() - SEG_LEN);
        cout << "\nData : " << data<< endl;
}

int main() {
        string s;
        cout << "Enter the codeword : ";
        cin >> s;
        checksum(s);
        return 0;
}
```

OUTPUT :

```
Enter the codeword : 10110011101010110101101011010101110000

No error in received data!
Codeword : 10110011101010110101101011010101110000
Data : 1011001110101011010110101101010
```