Name: Brijesh Rameshbhai Rohit

Admission number: U19CS009

CN-ASSIGNMENT-04

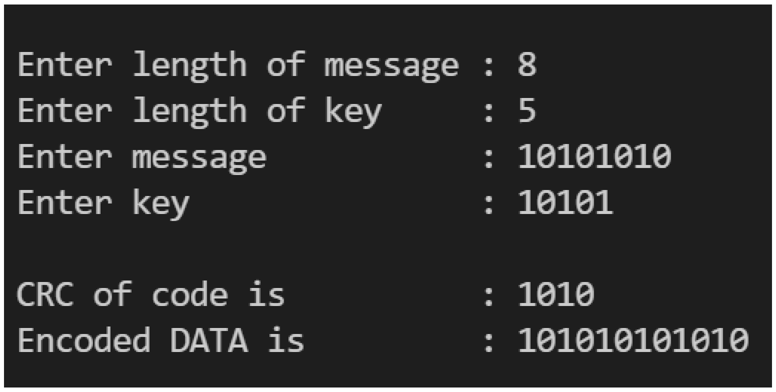
1. Implement ERROR DETECTION technique CRC in C PROGRAMMING.
Create two code files sender.c and receiver.c.

Sender.c file should accept data and key both as input in binary and encoded data(data+checksum) as output.

CODE:

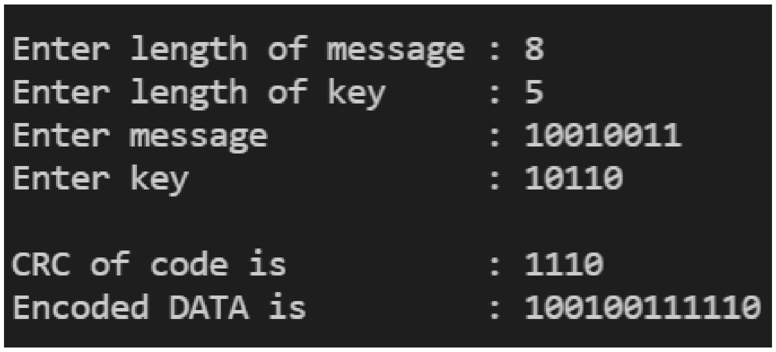
```
#include <bits/stdc++.h>
using namespace std;


void encode(vector<int> &data, vector<int> key, int n, int r)
{
    for (int i = 0; i < n; i++)
    {
        if (data[i] != key[0])
            continue;

        for (int j = 0, k = i; j < r; j++, k++)
            data[k] ^= key[j];
    }
}
int main()
{
    int n, r;

    //length of DATA
    cout << endl << "Enter length of message\t: ";
    cin >> n;

    //length of KEY
    cout << "Enter length of key\t: ";
    cin >> r;

    //vector/Array to store DATA, KEYS and ENCODED data
    vector<int> data(n + r - 1, 0);
    vector<int> key(r);
    vector<int> crcEncoded(n + r - 1, 0);
    //input data
    string temp;
    cout << "Enter message\t\t: ";
    getline(cin, temp);

    //keying DATA with respect to ASCII of '0'
    for (int i = 0; i < n; i++)
        data[i] = temp[i] - '0';

    //input KEY
    cout << "Enter key\t\t: ";
    getline(cin, temp);


    //keying KEYS with respect to ASCII of '0'
    for (int i = 0; i < r; i++)
        key[i] = temp[i] - '0';

    crcEncoded = data;


    //encoding using encode function
    encode(crcEncoded, key, n, r);


    //encoded output
    cout << endl << "CRC of code is\t\t: ";
    for (int i = 0; i < r - 1; i++)
    {
        data[n + i] = crcEncoded[n + i];
        cout << data[n + i];
    }


    cout << endl << "Encoded DATA is\t\t: ";
    for (int i = 0; i < n + r - 1; i++)
        cout << data[i];

    cout << endl << endl;
    return 0;
}
```

OUTPUT 1:



```
Enter length of message : 8
Enter length of key     : 5
Enter message           : 10101010
Enter key               : 10101

CRC of code is          : 1010
Encoded DATA is         : 101010101010
```

OUTPUT 2:

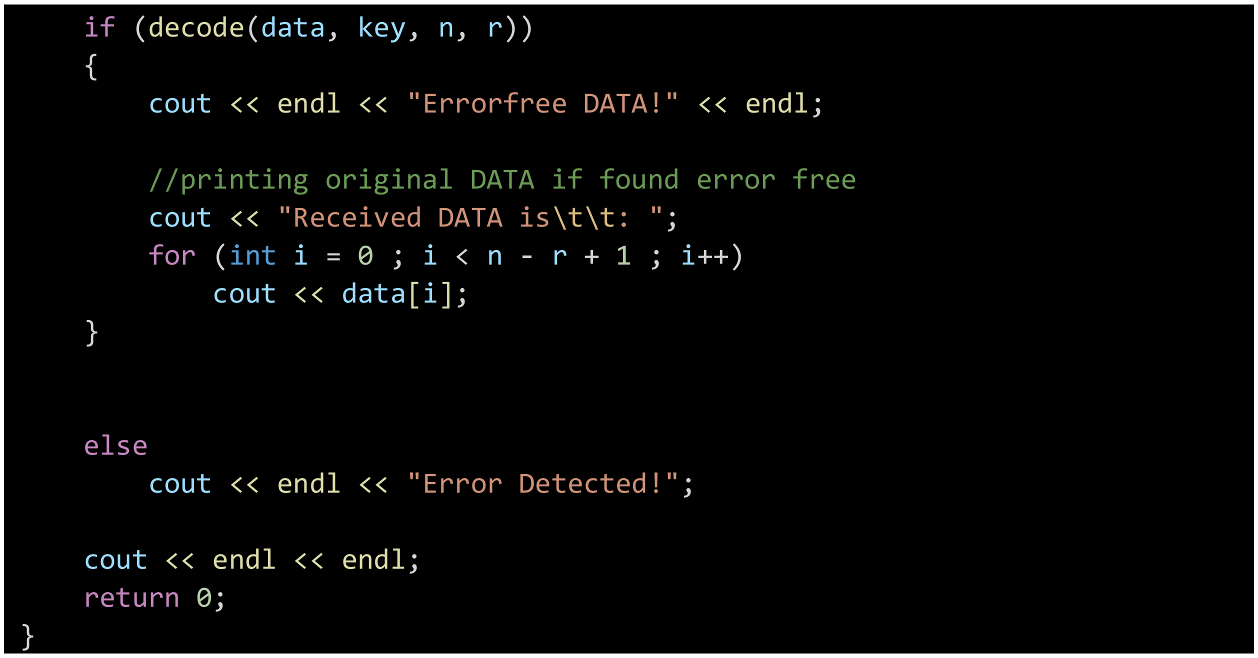


```
Enter length of message : 8
Enter length of key     : 5
Enter message           : 10010011
Enter key               : 10110

CRC of code is          : 1110
Encoded DATA is         : 100100111110
```

Receiver.c file should accept encoded data(data+checksum) and key as input and "Error Detected" OR "Error not Detected" output message.

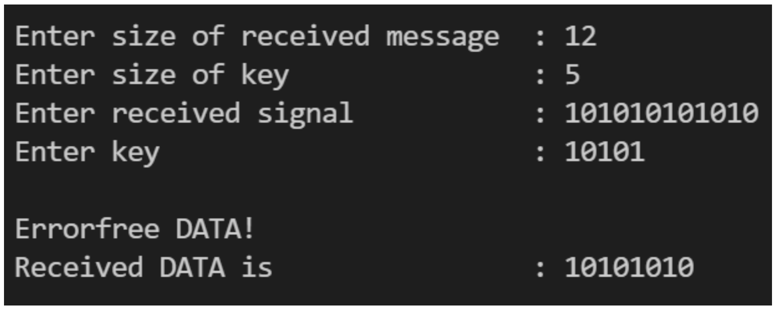
CODE:

```
#include <bits/stdc++.h>
using namespace std;


bool decode(vector<int> data, vector<int> key, int n, int r)
{
    for (int i = 0; i < n - r + 1; i++)
    {
        if (data[i] != key[0])
            continue;

        for (int j = 0, k = i; j < r; j++, k++)
            data[k] ^= key[j];
    }
    for (int i = n - r; i < n; i++)
        if (data[i] != 0)
            return false;


    return true;
}


int main()
{
    int n, r;

    //length of DATA received
    cout << endl << "Enter size of received message\t: ";
    cin >> n;

    //length of KEY
    cout << "Enter size of key\t\t: ";
    cin >> r;

    //vector/array to store DATA and KEY
    vector<int> data(n), key(r);
    string temp;

    //input DATA
    cout << "Enter received signal\t\t: ";
    getline(cin, temp);

    //keying DATA with respect to ASCII of '0'
    for (int i = 0; i < n; i++)
        data[i] = temp[i] - '0';

    //input KEY
    cout << "Enter key\t\t\t: ";
    getline(cin, temp);

    //keying KEYS with respect to ASCII of '0'
    for (int i = 0; i < r; i++)
        key[i] = temp[i] - '0';


    //checking if DATA is errorfree and decodable or not
```



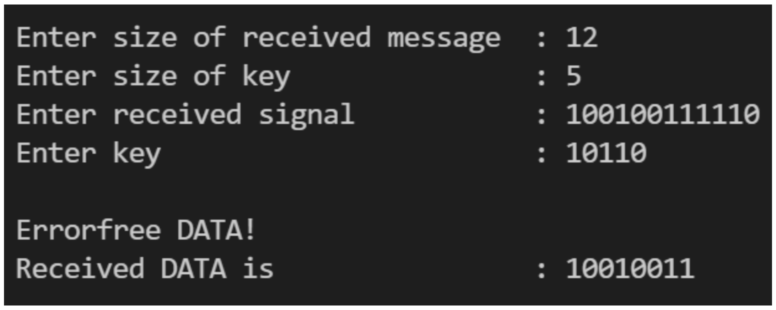
```cpp
    if (decode(data, key, n, r))
    {
        cout << endl << "Errorfree DATA!" << endl;

        //printing original DATA if found error free
        cout << "Received DATA is\t\t: ";
        for (int i = 0 ; i < n - r + 1 ; i++)
            cout << data[i];
    }


    else
        cout << endl << "Error Detected!";

    cout << endl << endl;
    return 0;
}
```

OUTPUT 1: **Corresponding to sender 1**



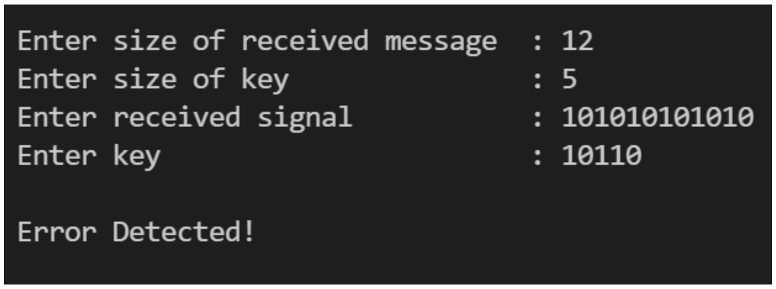
```
Enter size of received message  : 12
Enter size of key                : 5
Enter received signal            : 101010101010
Enter key                        : 10101

Errorfree DATA!
Received DATA is                 : 10101010
```

OUTPUT 2: **Corresponding to sender 2**



```
Enter size of received message  : 12
Enter size of key                : 5
Enter received signal            : 100100111110
Enter key                        : 10110

Errorfree DATA!
Received DATA is                 : 10010011
```

INVALID OUTPUT:


```
Enter size of received message  : 12
Enter size of key               : 5
Enter received signal           : 101010101010
Enter key                       : 10110

Error Detected!
```

2. Implement ERROR DETECTION technique 16-bit Checksum in C PROGRAMMING.

Create two code files sender.c and receiver.c

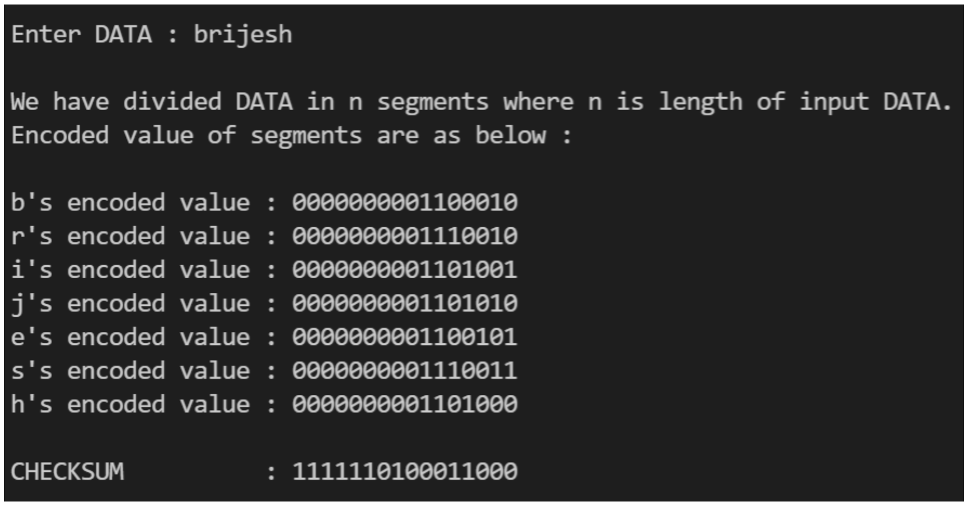
Sender.c file should accept input string (eg. Forouzan) and encoded string(Input data+checksum) as output.

CODE:

```
#include <bits/stdc++.h>
using namespace std;


vector<int> sum_bit(vector<int> a, vector<int> b)
{
    vector<int> add(16, 0);
    int carry = 0;

    for (int i = 15 ; i >= 0 ; i--)
    {
        add[i] = (a[i] + b[i] + carry) % 2;
        carry = (a[i] + b[i] + carry) / 2;
    }

    return add;
}


void complement(vector<int> &v)
{
    for (int i = 0 ; i < v.size() ; i++)
        v[i] = (v[i] == 1) ? 0 : 1;


}
```

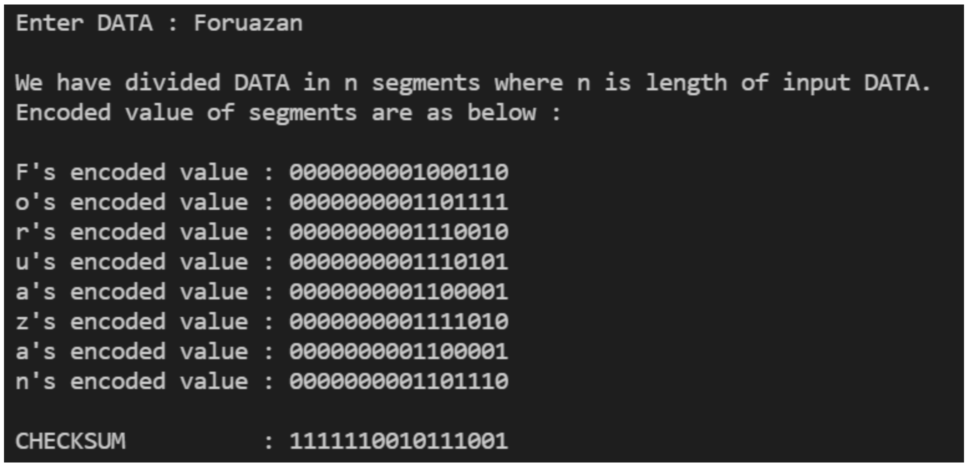
```cpp
int main()
{
    //DATA input
    string s;
    cout << endl << "Enter DATA : ";
    getline(cin, s);
    int n = s.length();


    //2-D array/vector to store and manipulate DATA's encoding
    vector<vector<int>> v(n);

    for (int i = 0 ; i < n ; i++)
    {
        int val = s[i];
        vector<int> encode;


        while (val != 0)
        {
            encode.push_back(val % 2);
            val /= 2;
        }

        while (encode.size() != 16)
            encode.push_back(0);

        reverse(encode.begin(), encode.end());
        v[i] = encode;
    }


    //check sum bits
    vector<int> checksum(16, 0);


    for (int i = 0 ; i < n ; i++)
        checksum = sum_bit(v[i], checksum);

    complement(checksum);


    cout << endl << "We have divided DATA in n segments where n is length of
input DATA." << endl;
    cout << "Encoded value of segments are as below :" << endl << endl;
    for (int i = 0 ; i < n ; i++)
    {
        cout << s[i] << "'s encoded value : ";


        for (int j = 0 ; j < 16 ; j++)
            cout << v[i][j];

        cout << endl;
    }
    cout << endl << "CHECKSUM\t  : ";


    for (int i = 0 ; i < 16 ; i++)
        cout << checksum[i];

    cout << endl << endl;
    return 0;
}
```

OUTPUT 1:



```
Enter DATA : brijesh

We have divided DATA in n segments where n is length of input DATA.
Encoded value of segments are as below :

b's encoded value : 0000000001100010
r's encoded value : 0000000001110010
i's encoded value : 0000000001101001
j's encoded value : 0000000001101010
e's encoded value : 0000000001100101
s's encoded value : 0000000001110011
h's encoded value : 0000000001101000

CHECKSUM          : 1111110100011000
```

OUTPUT 2:



```
Enter DATA : Foruazan

We have divided DATA in n segments where n is length of input DATA.
Encoded value of segments are as below :

F's encoded value : 0000000001000110
o's encoded value : 0000000001101111
r's encoded value : 0000000001110010
u's encoded value : 0000000001110101
a's encoded value : 0000000001100001
z's encoded value : 0000000001111010
a's encoded value : 0000000001100001
n's encoded value : 0000000001101110

CHECKSUM          : 1111110010111001
```

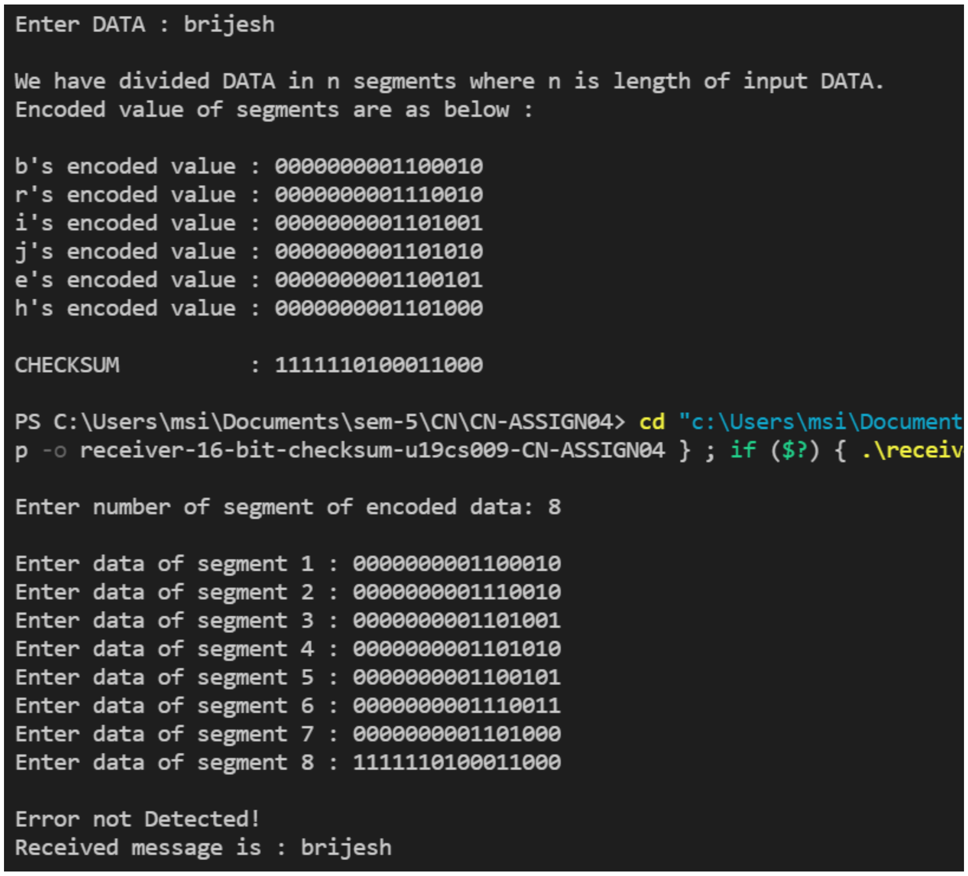
Receiver.c file should accept encoded data(data+checksum) and "Error Detected" OR "Error not Detected" output message.

CODE:

```
#include <bits/stdc++.h>
using namespace std;


vector<int> sum(vector<int> a, vector<int> b)
{
    vector<int> add(16, 0);
    int carry = 0;


    for (int i = 15 ; i >= 0 ; i--)
    {
        add[i] = (a[i] + b[i] + carry) % 2;
        carry = (a[i] + b[i] + carry) / 2;
    }

    return add;
}


void complement(vector<int> &v)
{
    for (int i = 0 ; i < v.size() ; i++)
        v[i] = (v[i] == 1) ? 0 : 1;


}
```

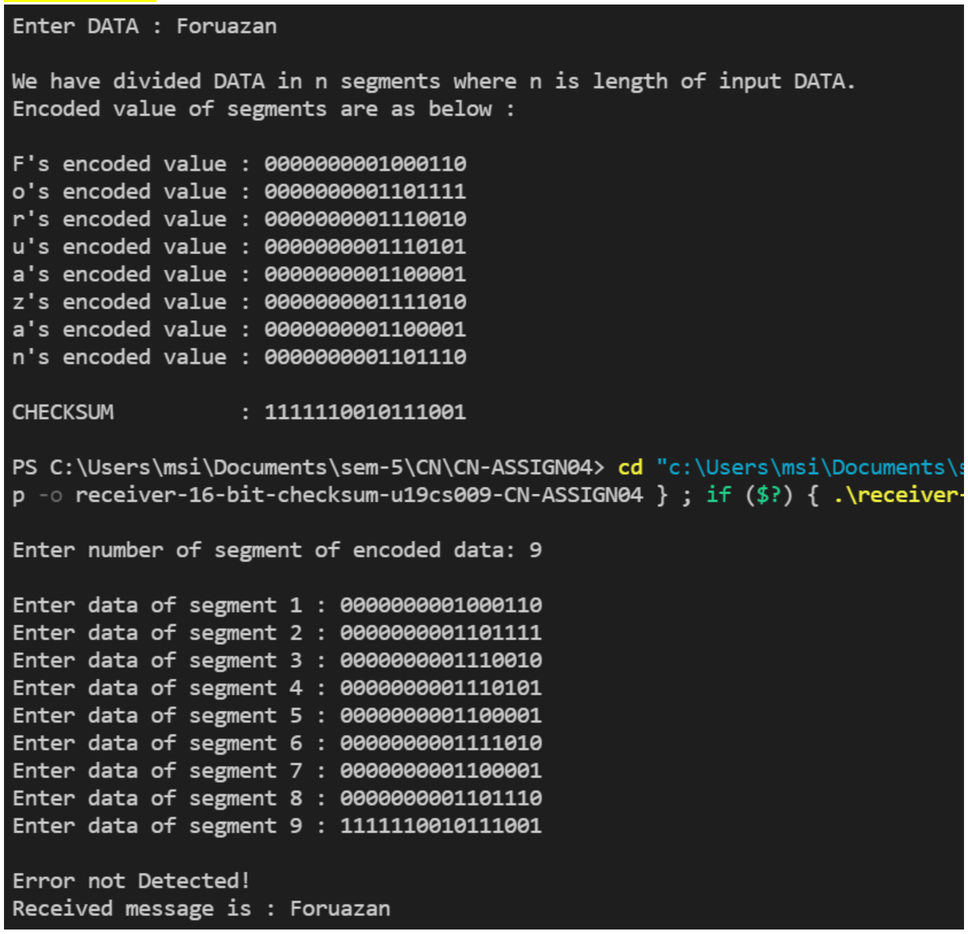
```cpp
bool decode(vector<int> v)
{
    for (int i = 0 ; i < v.size() ; i++)
    {
        if (v[i] == 1)
            return false;
    }


    return true;
}


int decodeToChar(vector<int> v)
{
    reverse(v.begin(), v.end());
    int val = 0;


    for (int i = 0; i < v.size(); i++)
        val += v[i] * (1 << i);

    return val;
}


int main()
{
    //input DATA
    int n;
    cout << endl << "Enter number of segment of encoded data: ";
    cin >> n;
    cout <<  endl;


    //2-D array/vector to store and manipulate DATA's decoding
    vector<vector<int>> v(n);


    for (int i = 0; i < n; i++)
    {
        string s;
        cout << "Enter data of segment " << i + 1 << " : ";
        cin >> s;

        vector<int> segment(s.length());


        for (int i = 0 ; i < s.length() ; i++)
            segment[i] = s[i] - '0';

        v[i] = segment;
    }


    //check sum bits
    vector<int> checksum(16, 0);


    for (int i = 0 ; i < n ; i++)
        checksum = sum(v[i], checksum);

    complement(checksum);


    if (decode(checksum))
    {
        cout << endl << "Error not Detected!" << endl;
        cout << "Received message is : ";


        for (int i = 0; i < n - 1; i++)
        {
            int n = decodeToChar(v[i]);
            cout << (char)n;
        }


        cout << endl;
    }
    else
        cout << "Error Detected!" << endl;

    cout << endl;
    return 0;
}
```

OUTPUT 1: **Corresponding to output 1**



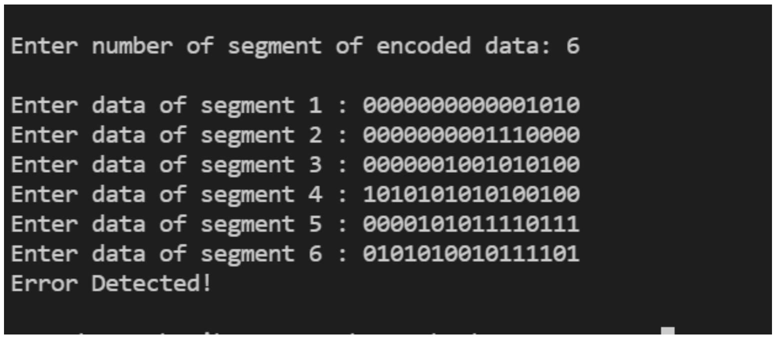
```
Enter DATA : brijesh

We have divided DATA in n segments where n is length of input DATA.
Encoded value of segments are as below :

b's encoded value : 0000000001100010
r's encoded value : 0000000001110010
i's encoded value : 0000000001101001
j's encoded value : 0000000001101010
e's encoded value : 0000000001100101
h's encoded value : 0000000001101000

CHECKSUM          : 1111110100011000

PS C:\Users\msi\Documents\sem-5\CN\CN-ASSIGN04> cd "c:\Users\msi\Document
p -o receiver-16-bit-checksum-u19cs009-CN-ASSIGN04 } ; if ($?) { .\receiv

Enter number of segment of encoded data: 8

Enter data of segment 1 : 0000000001100010
Enter data of segment 2 : 0000000001110010
Enter data of segment 3 : 0000000001101001
Enter data of segment 4 : 0000000001101010
Enter data of segment 5 : 0000000001100101
Enter data of segment 6 : 0000000001110011
Enter data of segment 7 : 0000000001101000
Enter data of segment 8 : 1111110100011000

Error not Detected!
Received message is : brijesh
```

OUTPUT 2: **Corresponding to output 2**



```
Enter DATA : Foruazan

We have divided DATA in n segments where n is length of input DATA.
Encoded value of segments are as below :

F's encoded value : 0000000001000110
o's encoded value : 0000000001101111
r's encoded value : 0000000001110010
u's encoded value : 0000000001110101
a's encoded value : 0000000001100001
z's encoded value : 0000000001111010
a's encoded value : 0000000001100001
n's encoded value : 0000000001101110

CHECKSUM          : 1111110010111001

PS C:\Users\msi\Documents\sem-5\CN\CN-ASSIGN04> cd "c:\Users\msi\Documents\
p -o receiver-16-bit-checksum-u19cs009-CN-ASSIGN04 } ; if ($?) { .\receiver-

Enter number of segment of encoded data: 9

Enter data of segment 1 : 0000000001000110
Enter data of segment 2 : 0000000001101111
Enter data of segment 3 : 0000000001110010
Enter data of segment 4 : 0000000001110101
Enter data of segment 5 : 0000000001100001
Enter data of segment 6 : 0000000001111010
Enter data of segment 7 : 0000000001100001
Enter data of segment 8 : 0000000001101110
Enter data of segment 9 : 1111110010111001

Error not Detected!
Received message is : Foruazan
```

INVALID OUTPUT :



```
Enter number of segment of encoded data: 6

Enter data of segment 1 : 0000000000001010
Enter data of segment 2 : 0000000001110000
Enter data of segment 3 : 0000001001010100
Enter data of segment 4 : 1010101010100100
Enter data of segment 5 : 0000101011110111
Enter data of segment 6 : 0101010010111101
Error Detected!
```