Programming Assignment – 1

Group of –

Abhishek Bahuguna (Enroll.-2019BITE008)

&

Suraj Kannujiya (Enroll.-2019BITE003)

Branch-Information Technology

Subject- Data Communication

Subject Teacher- Dr. Iqra Altaf Gillani

About the Project:

Programming Language Used- C++

Graphics Library Used – SFML (Simple and Fast Multimedia Library)

About C++: C++ is a general-purpose programming language created by Bjarne Stroustrup as an extension of the C programming language, or "C with Classes". The language has expanded significantly over time, and modern C++ now has object-oriented, generic, and functional features in addition to facilities for low-level memory manipulation. It is almost always implemented as a compiled language, and many vendors provide C++ compilers, including the Free Software Foundation, LLVM, Microsoft, Intel, Oracle, and IBM, so it is available on many platforms.[9]

About SFML: SFML provides a simple interface to the various components of your PC, to ease the development of games and multimedia applications. It is composed of five modules: system, window, graphics, audio and network.

About The Assignment:

Topics:

1.Generation of Data Stream

Data stream is generated in an integer format, and stored in vector

For random sequence the length is restricted to 20 bits and for the stream with 4 or 8 consecutive 0’s length is restricted to 30.

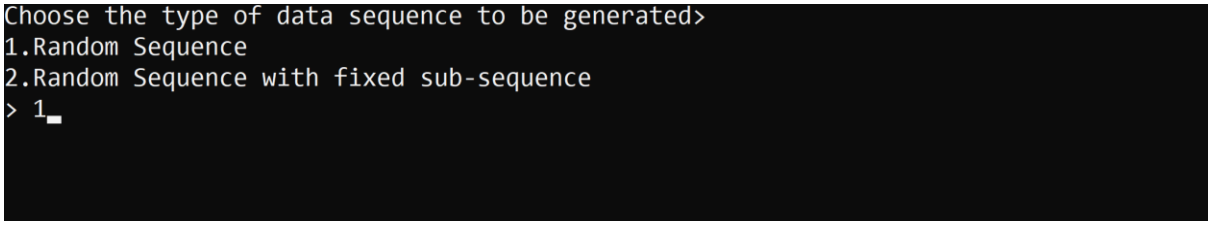
Both of the stream are generated using a simple loop and some if statements.

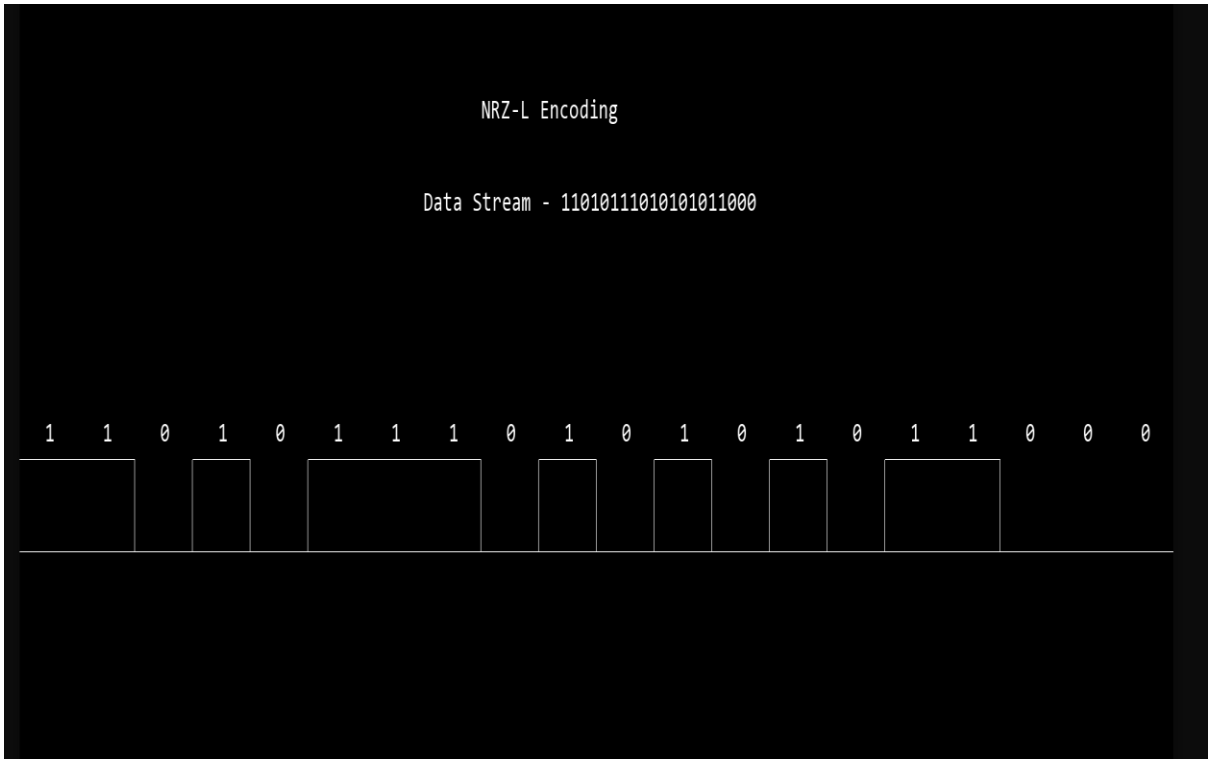
2.Encoding

1.NRZ-L Encoding

In NRZ-L encoding, the polarity of the signal changes only when the incoming signal changes from a 1 to a 0 or from a 0 to a 1. NRZ-L method looks just like the NRZ method, except for the first input one data bit. This is because NRZ does not consider the first data bit to be a polarity change, where NRZ-L does.

Output




NRZ-L Encoding
Data Stream - 11010111010101011000
1 1 0 1 0 1 1 1 0 1 0 1 0 1 0 1 1 0 0 0

2.NRZ-I Encoding

Transition at the beginning of bit interval = bit 1 and No Transition at beginning of bit interval = bit 0 or vicecersa. This technique is known as differential encoding.

Output

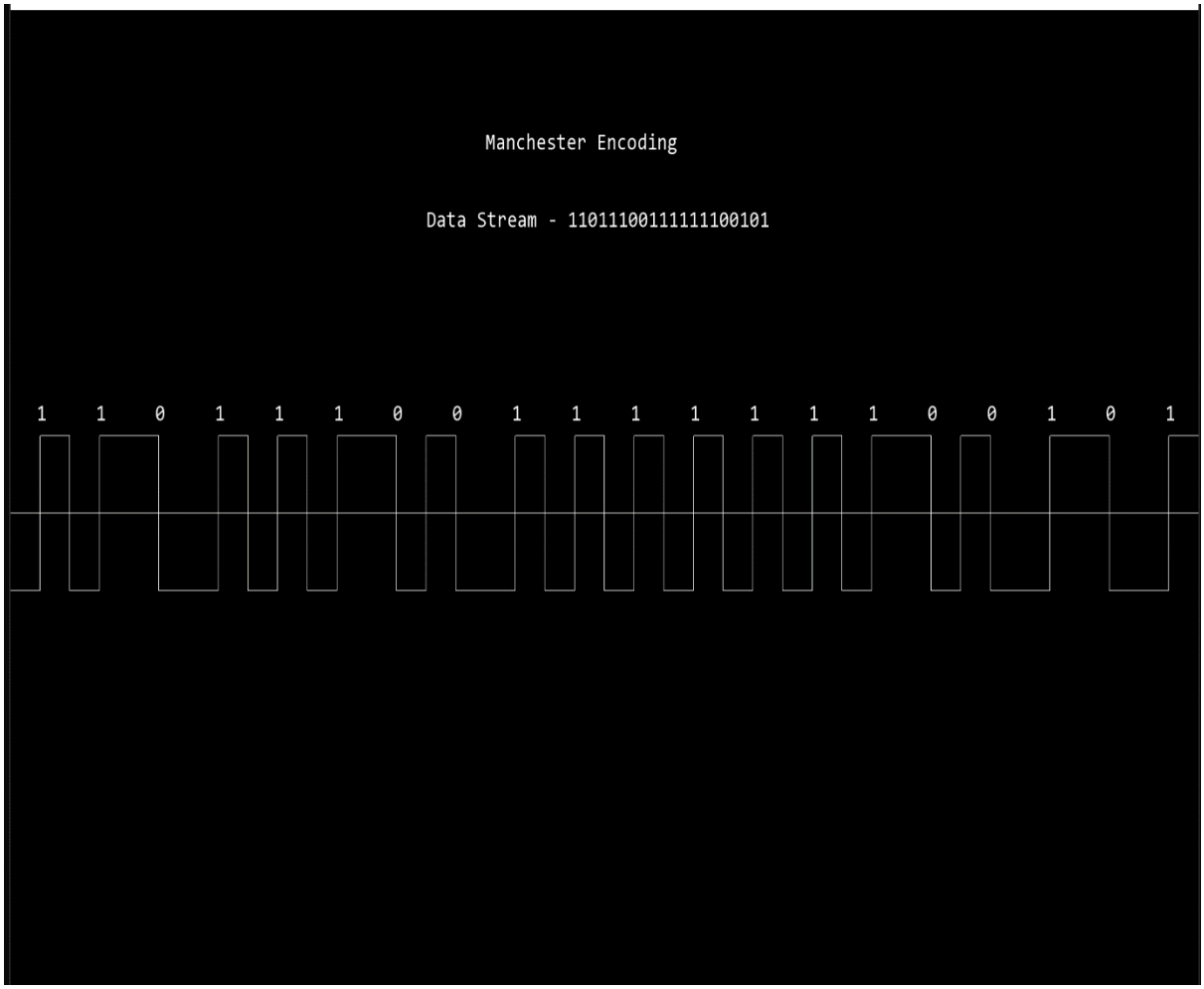
```
Generated Data Sequence>
0 0 0 0 0 1 0 0 0 1 1 0 0 0 0 0 0 0 1 0
Choose the type of Encoding>
1.NRZ-L
2.NRZ-I
3.Manchester
4.Differential Manchester
5.AMI
> 2
```

3.Manchester Encoding

Manchester encoding is a synchronous clock encoding technique used by the physical layer of the Open System Interconnection [OSI] to encode the clock and data of a synchronous bit stream.

Output

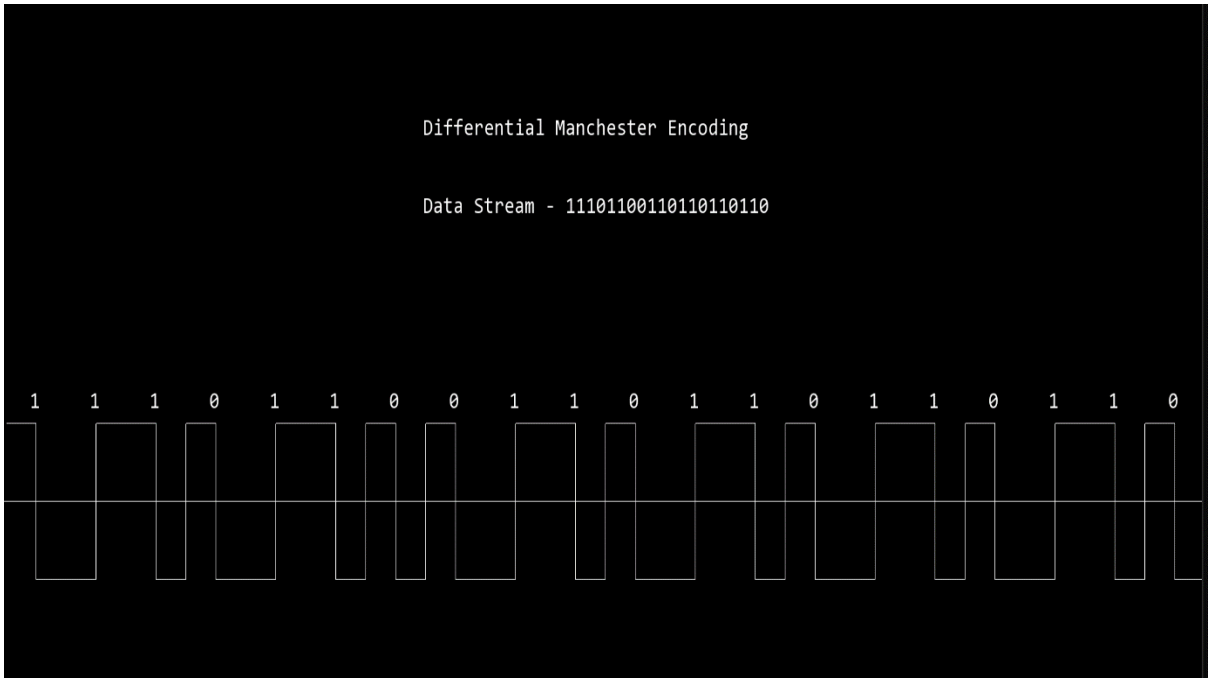
```
Generated Data Sequence>
1 1 0 1 1 1 0 0 1 1 1 1 1 1 1 0 0 1 0 1
Choose the type of Encoding>
1.NRZ-L
2.NRZ-I
3.Manchester
4.Differential Manchester
5.AMI
> 3
```



4.Differential Manchester Encoding

Differential Manchester encoding is a differential encoding technology, using the presence or absence of transitions to indicate logical value. An improvement to Manchester coding which is a special case of binary phase-shift keying, it is not necessary to know the initial polarity of the transmitted message signal, because the information is not represented by the absolute voltage levels but by their transitions.

Output



5.AMI

1.B8ZS Scrambling

Bipolar with 8-zero substitution (B8ZS) is commonly used in North America. In this technique, eight consecutive zero-level voltages are replaced by the sequence OOOVBOVB.

The V in the sequence denotes violation; this is a nonzero voltage that breaks an AMI rule of encoding (opposite polarity from the previous). The B in the sequence denotes bipolar which means a nonzero level voltage in accordance with the AMI rule.

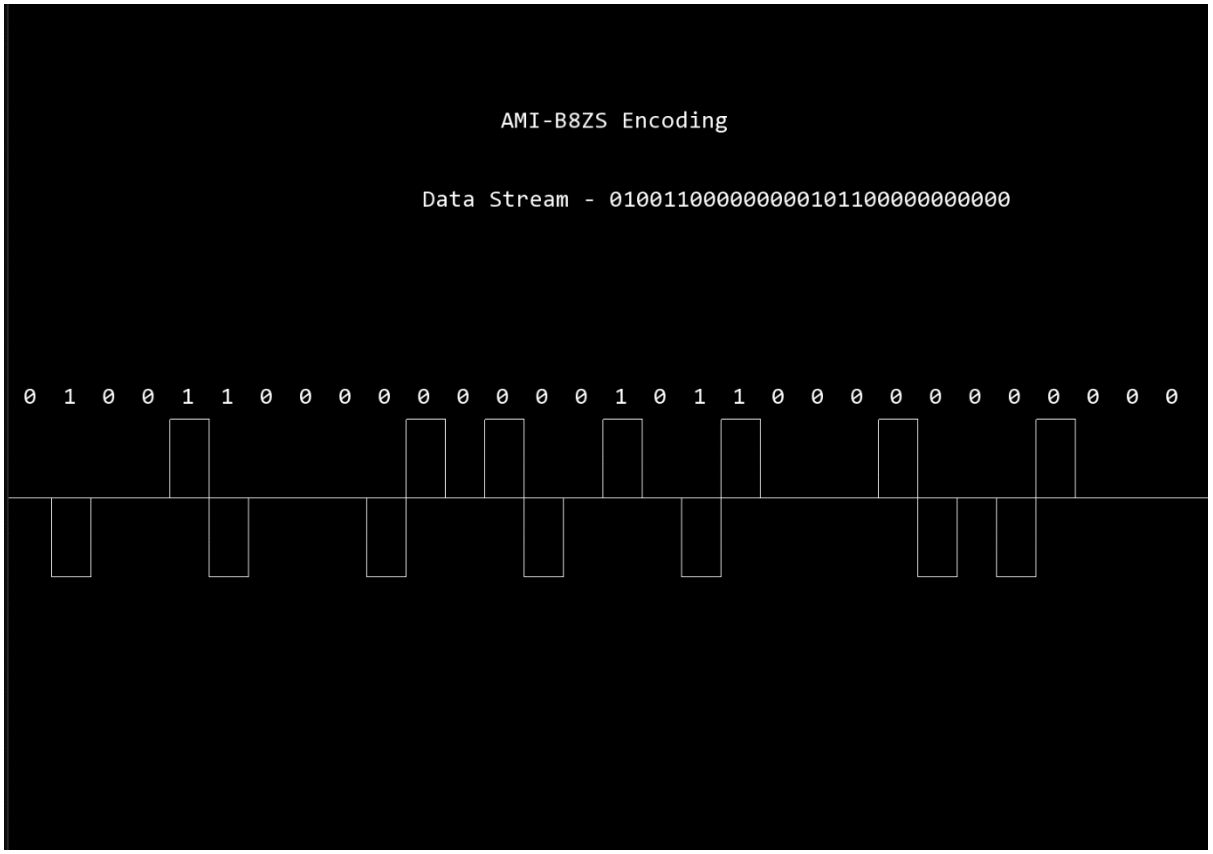
Output

```
Choose the type of data sequence to be generated>
1.Random Sequence
2.Random Sequence with fixed sub-sequence
> 2
```

```
Choose a sub-sequence>
1. 8 consecutive zeroes (B8ZS and HDB3)
2. 4 consecutive zeroes (HDB3)
> 1
```

```
Generated Data Sequence>
0 1 0 0 1 1 0 0 0 0 0 0 0 0 1 0 1 1 0 0 0 0 0 0 0 0 0 0
Choose the type of Encoding>
1.NRZ-L
2.NRZ-I
3.Manchester
4.Differential Manchester
5.AMI
> 5
```

```
Choose the type of scrambling>
1.B8ZS
2.HDB3
> 1
```



2.HDB3 Scrambling

High-density bipolar 3-zero (HDB3) is commonly used outside of North America. In this technique, which is more conservative than B8ZS, four consecutive zero-level voltages are replaced with a sequence of OOOV or BOOV. The reason for two different substitutions is to maintain the even number of nonzero pulses after each substitution.

The two rules can be stated as follows:

1. If the number of nonzero pulses after the last substitution is odd, the substitution pattern will be OOOV, which makes the total number of nonzero pulses even.

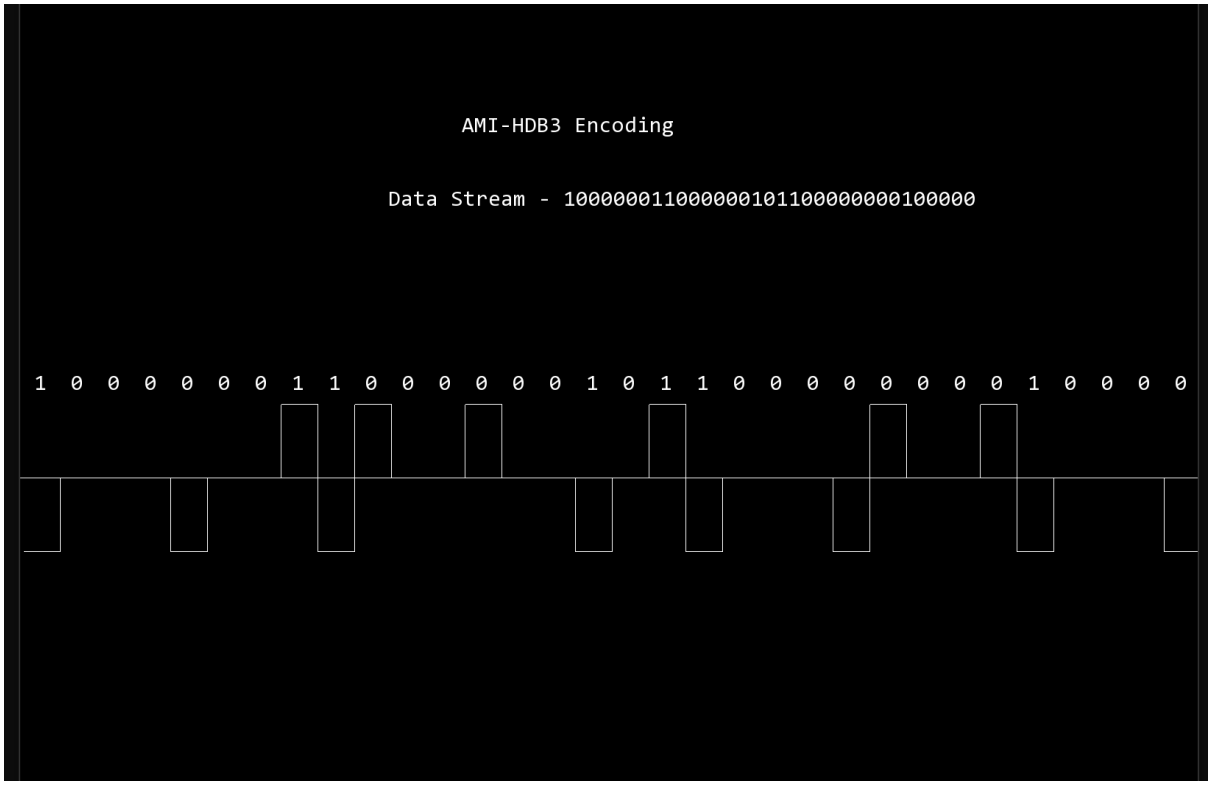
2.If the number of nonzero pulses after the last substitution is even, the substitution pattern will be BOOV, which makes the total number of nonzero pulses even.

Output

```
Choose a sub-sequence>
1. 8 consecutive zeroes (B8ZS and HDB3)
2. 4 consecutive zeroes (HDB3)
> 2
```

```
Generated Data Sequence>
1 0 0 0 0 0 0 1 1 0 0 0 0 0 0 1 0 1 1 0 0 0 0 0 0 0 0 1 0 0 0 0 0
Choose the type of Encoding>
1.NRZ-L
2.NRZ-I
3.Manchester
4.Differential Manchester
5.AMI
> 5
```

```
Choose the type of scrambling>
1.B8ZS
2.HDB3
> 2
```


AMI-HDB3 Encoding
Data Stream - 10000001100000010110000000100000
1 0 0 0 0 0 0 1 1 0 0 0 0 0 0 1 0 1 1 0 0 0 0 0 0 0 0 1 0 0 0 0

3.Longest Palindormic Subsequence

This problem is close to the Longest Common Subsequence (LCS) problem. In fact, we can use LCS as a subroutine to solve this problem. Following is the two step solution that uses LCS.

Algorithm-

- Reverse the given sequence and store the reverse in another array say rev[0..n-1]
- LCS of the given sequence and rev[] will be the longest palindromic sequence.
- Once we have found LCS, we can print the LCS.

```
Original Sequence > 11010111010101011000
Longest Palindromic Subsequence > 001101010101100
```

```
Original Sequence > 00000100011000000010
Longest Palindromic Subsequence > 000000001100000000
```

```
Original Sequence > 11011100111111100101
Longest Palindromic Subsequence > 10100111111100101
```

```
Original Sequence > 11101100110110110110
Longest Palindromic Subsequence > 11011011011011011
```