NAME : MANVI SHARMA USN: 1BM22CS149

CN LAB 9

<u>AIM:</u> IMPLEMENTATION OF CRC ALGORITHM

OBSERVATION:

Date: Page:	Will Burgling a British Burgling
Implementation of CRC Algorithm (Polynomial)	- string sender string dataward, string divisor)
CRC = 8 claterword = x8 + x3 + x1 +1	otting poolded Dota - dotaword + steing (divisorlay)
Chica o summero a x 1 x 12 11	-1,'0');
10.1.1.1.1.1.1.1.1.1.1.1.1.1.1.1.1.1.1.	String nem - dividata (padded Data, divicor);
# indudexiostneam >	notium dateword + nem;
thinclude (othering)	AND AND WET STAND
wing namepace std;	4 Act major con
11-from to perform ROR	> bool receiver (strong received Data, steing divisor) &
> String nonoperation (string dividend, whong divisor) &	string yem = divido Date (recinad Date, divion);
string rent = "	uitum rem find(11) = otring: npa;
for (int i=1; 1 divisor.length (); i++)4	Yell Franker Structure of the second
Meant += (dividend [1] == divisor[i] ?'o'; 's');	
A Some to monda (1) = Townstar(1) , o , 2).	Pelniam tri
1.4	string dakward, distribur,
Return yesult;	Cout « "order datemord";
raintin, transfel with	cin>> datemord
boundary of study	cout < enter divisor
Strong divide Date (strong date, Strong divisor) }	cin >> divisor
string temps data substitue, divisor lengthe);	coud « in "transmitted date " " TD exert!
fortint i= divisor.length(); i<=datelength(); i++)\$	courter received data;
/ lemp[0] == '1')}	ciny string RD;
temp = xorrepression (temp, divisor)	Jn >> RD:
3 olve f	1 receiver (RO, divisor))
temp > Konopenation (temp, alterny divisor-length), (0)	cout «" Valid (No oxnox)."
3 (along anson on In), b)	else à
: Cichota de Marso	cout << " Error " << end!
of (icdata.teny th(1))	4
temp to date 07;	· o neutron
3	ofpo Enter datemared: 110)
110	Enter divisor: 1011
noturn temp: substa (i);	Treammitted Data: - 110101
a de la contraction de la cont	Enter received date > 110101
Jon Short Shirt Sh	Received Dota to Valid (No exercised)
the land to the land	- Concreted)

PROGRAM:

```
#include <iostream>
#include <string>
using namespace std;
// Function to perform XOR operation
string xorOperation(string dividend, string divisor) {
  string result = "";
  for (int i = 1; i < divisor.length(); i++) {
     result += (dividend[i] == divisor[i] ? '0' : '1');
  }
  return result;
}
// Function to perform binary division
string divideData(string data, string divisor) {
  string temp = data.substr(0, divisor.length());
  for (int i = divisor.length(); i <= data.length(); i++) {
     if (temp[0] == '1') {
        temp = xorOperation(temp, divisor);
     } else {
        temp = xorOperation(temp, string(divisor.length(), '0'));
     if (i < data.length()) {
        temp += data[i];
  }
  return temp.substr(1); // Return the remainder
}
// Sender side: Generate transmitted data
string sender(string dataword, string divisor) {
  string paddedData = dataword + string(divisor.length() - 1, '0'); // Append zeros equal to
(degree of divisor - 1)
  string remainder = divideData(paddedData, divisor);
                                                                 // Calculate CRC
  return dataword + remainder;
                                                         // Concatenate dataword and CRC
remainder
```

```
}
// Receiver side: Verify received data
bool receiver(string receivedData, string divisor) {
  string remainder = divideData(receivedData, divisor);
  return remainder.find('1') == string::npos; // If remainder is all zeros, data is valid
}
int main() {
  string dataword, divisor;
  // Input dataword and divisor in binary form
  cout << "Enter the dataword (binary polynomial, e.g., 1101 for x^3 + x^2 + 1): ";
  cin >> dataword;
  cout << "Enter the divisor (binary polynomial, e.g., 1011 for x^3 + x + 1): ";
  cin >> divisor;
  // Sender Side
  string transmittedData = sender(dataword, divisor);
  cout << "\nTransmitted Data (Dataword + CRC): " << transmittedData << endl;
  // Receiver Side
  cout << "Enter the received data (binary polynomial): ";
  string receivedData;
  cin >> receivedData;
  if (receiver(receivedData, divisor)) {
     cout << "Received data is valid (No error detected)." << endl;
  } else {
     cout << "Received data contains errors." << endl;
  }
  return 0;
}
```

OUTPUT:

CASE 1 -> NO ERROR

Enter the dataword (binary polynomial, e.g., 1101 for $x^3 + x^2 + 1$): 1101 Enter the divisor (binary polynomial, e.g., 1011 for $x^3 + x + 1$): 1011

Transmitted Data (Dataword + CRC): 110101 Enter the received data (binary polynomial): 110101 Received data is valid (No error detected).

=== Code Execution Successful === === Session Ended. Please Run the code again ===

CASE 2 -> ERROR

Enter the dataword (binary polynomial, e.g., 1101 for $x^3 + x^2 + 1$): 1101 Enter the divisor (binary polynomial, e.g., 1011 for $x^3 + x + 1$): 1011

Transmitted Data (Dataword + CRC): 110101 Enter the received data (binary polynomial): 110100 Received data contains errors.

=== Code Execution Successful ===