

### **EXPERIMENT-3**

**Write a Program to implement data link layer farming method checksum.**

The Checksum is an error detection method that detected errors in data/message while it is transmitted from sender to receiver. This method is used by the higher layer protocols and makes use of the Checksum Generator on the Sender side and Checksum Checker on the Receiver side.

**Examples:**

**Input:** sent\_message = "10101111", rec\_message = "10101101", block\_size = 8

**Output:** Error

**Explanation:** Since the 7th bit in the sent\_message and the rec\_message is different, the final checksum value is not equal to zero denoting that some error has occurred during transmission.

**Input:** sent\_message = "10000101011000111001010011101101",  
rec\_message = "10000101011000111001010011101101"  
block\_size = 8

**Output:** No Error

**Program**

```
#include <iostream.h>
#include <conio.h>
#include <string.h>
void onescomplement(char data[], int n)
{
    int i;
    for (i = 0; i < n; i++)
    {
        if(data[i] == '0')
            data[i] = '1';
        else
            data[i] = '0';
    }
}
void add(char a[], char b[], char result[], int n)
{
    int i, carry = 0, sum;
    for (i = n - 1; i >= 0; i--)
    {
```

```

        sum = (a[i] - '0') + (b[i] - '0') + carry;
        result[i] = (sum % 2) + '0';
        carry = sum / 2;
    }
    result[n] = '\0';
    while (carry)
    {
        for (i = n - 1; i >= 0; i--)
        {
            sum = (result[i] - '0') + carry;
            result[i] = (sum % 2) + '0'; carry = sum / 2;
            if (!carry)
                break;
        }
    }
}

void checksum(char d[], int len, int size, char out[])
{
    char sum[32], blk[32]; int i, j;
    for (i = 0; i < size; i++)
        sum[i] = d[i];
    sum[size] = '\0';
    for (i = size; i < len; i += size)
    {
        for (j = 0; j < size; j++)
            blk[j] = d[i + j];
        blk[size] = '\0';
        add(sum, blk, sum, size);
    }
    onescomplement(sum, size);
    strcpy(out, sum);
}

int isValid(char sender[], char receiver[], int size)
{
    char chk[32], tmp[256], res[32];

```

```

checksum(sender, strlen(sender), size, chk);
strcpy(tmp, receiver); strcat(tmp, chk);
checksum(tmp, strlen(tmp), size, res);
for (int i = 0; i < size; i++)
    if (res[i] != '0')
        return 0;
    return 1;
}
void main()
{
    clrscr();
    char sender[256], receiver[256], checksum[32];
    int size;
    cout << "Enter binary sender data (e.g. 10101010...): ";
    cin >> sender;
    cout << "Enter block size (e.g. 8): ";
    cin >> size;
    strcpy(receiver, sender); // receiver is same as sender
    if (isValid(sender, receiver, size))
        cout << "\nNo Error (Valid Data)";
    else
        cout << "\nError Detected!";
    getch();
}

```

### **Output:**

Enter binary sender data (e.g. 10101010...):

11001100101010101111000011000011

Enter block size (e.g. 8): 8

No Error (Valid Data)