

NETWORK AND COMMUNICATION

LAB 3

AIM: Implement a parity generator and checker for the string input using a. Even parity b. Odd Parity

EVEN PARITY**Code**

```
#include<iostream>
using namespace std;

int main()
{
    cout << "Data sent (Five - 5 bit strings) : " << endl;
    int input[6][6], temp;
    for(int i = 0; i < 5; i++)
    {
        int rowOnes = 0;
        for(int j = 0; j < 5; j++)
        {
            cin >> temp;
            input[i][j] = temp;
            if(temp == 1)
                rowOnes += 1;
        }
        if(rowOnes%2 == 0)
            input[i][5] = 0;
        else
            input[i][5] = 1;
    }
    input[5][5] = 0;
    cout << endl;
    cout << "The parity bits will now be generated. (EVEN PARITY)\n" << endl;
    for(int i = 0; i < 5; i++)
    {
        int colOnes = 0;
        for(int j = 0; j < 5; j++)
        {
```

```

        if(input[j][i] == 1)
            colOnes += 1;
    }
    if(colOnes%2 == 0)
        input[5][i] = 0;
    else
        input[5][i] = 1;
}
cout << "Transmitted data (with even parity bits: Six - 6 bit strings) : \n";
for(int i = 0; i < 6; i++)
{
    for(int j = 0; j < 6; j++)
    {
        cout << input[i][j] << " ";
    }
    cout << endl;
}
cout << endl << endl;
cout << "Received data (Six - 6 bit strings) : " << endl;
int checker[6][6];
for(int i = 0; i < 6; i++)
{
    for(int j = 0; j < 6; j++)
    {
        cin >> temp;
        checker[i][j] = temp;
    }
}
cout << endl;
cout << "Receiver system knows that the last column and last row are the parity bits.\n\n";
int counter = 0;
for(int i = 0; i < 5; i++)
{
    for(int j = 0; j < 5; j++)
    {
        if(input[i][j] != checker[i][j])
            counter++;
    }
}

```

```

    }
}
cout << "Received data contains " << counter << " errors.\n" << endl;
}

```

OUTPUT

```

dhruv@dhruv-Inspiron-5559:~$ g++ parityCheck.cpp
dhruv@dhruv-Inspiron-5559:~$ ./a.out
Data sent (Five - 5 bit strings) :
1 0 1 1 0
0 0 1 0 0
1 1 1 0 0
1 1 0 0 0
1 0 0 0 0

The parity bits will now be generated. (EVEN PARITY)

Transmitted data (with even parity bits: Six - 6 bit strings) :
1 0 1 1 0 1
0 0 1 0 0 1
1 1 1 0 0 1
1 1 0 0 0 0
1 0 0 0 0 1
0 0 1 1 0 0

Received data (Six - 6 bit strings) :
1 0 1 1 0 1
0 0 1 0 1 1
1 1 1 0 0 1
1 1 0 1 1 0
1 1 0 0 0 1
0 0 1 1 0 0

Receiver system knows that the last column and last row are the parity bits.
Received data contains 4 errors.

```

ODD PARITY

Code

```

#include<iostream>

using namespace std;

int main()
{
    cout << "Data sent (Five - 5 bit strings) : " << endl;
    int input[6][6], temp;
    for(int i = 0; i < 5; i++)

```

```

{
    int rowOnes = 0;
    for(int j = 0; j < 5; j++)
    {
        cin >> temp;
        input[i][j] = temp;
        if(temp == 1)
            rowOnes += 1;
    }
    if(rowOnes%2 == 0)
        input[i][5] = 1;
    else
        input[i][5] = 0;
}
input[5][5] = 0;
cout << endl;
cout << "The parity bits will now be generated. (ODD PARITY)\n" << endl;
for(int i = 0; i < 5; i++)
{
    int colOnes = 0;
    for(int j = 0; j < 5; j++)
    {
        if(input[j][i] == 1)
            colOnes += 1;
    }
    if(colOnes%2 == 0)
        input[5][i] = 1;
    else
        input[5][i] = 0;
}
cout << "Transmitted data (with odd parity bits: Six - 6 bit strings) : \n";
for(int i = 0; i < 6; i++) {
    for(int j = 0; j < 6; j++)
        cout << input[i][j] << " ";
    cout << endl;
}
cout << endl;

```

```

cout << "Received data (Six - 6 bit strings) : " << endl;

int checker[6][6];

for(int i = 0; i < 6; i++)
{
    for(int j = 0; j < 6; j++)
    {
        cin >> temp;
        checker[i][j] = temp;
    }
} cout << endl;

cout << "Receiver system knows that the last column and last row are the parity bits.\n\n";

int counter = 0;

for(int i = 0; i < 5; i++) {
    for(int j = 0; j < 5; j++) {
        if(input[i][j] != checker[i][j])
            counter++;
    }
}

cout << "Received data contains " << counter << " errors.\n" << endl;
}

```

```

dhruv@dhruv-Inspiron-5559:~$ g++ parityCheck.cpp
dhruv@dhruv-Inspiron-5559:~$ ./a.out
Data sent (Five - 5 bit strings) :
0 0 1 0 0
1 0 1 1 0
0 1 0 0 1
1 1 0 1 1
1 0 1 0 0

The parity bits will now be generated. (ODD PARITY)

Transmitted data (with odd parity bits: Six - 6 bit strings) :
0 0 1 0 0 0
1 0 1 1 0 0
0 1 0 0 1 1
1 1 0 1 1 1
1 0 1 0 0 1
0 1 0 1 1 0

Received data (Six - 6 bit strings) :
1 0 1 0 0 0
1 0 1 1 0 0
0 1 0 0 1 1
1 0 0 1 1 1
1 0 0 0 0 1
0 1 0 1 1 0

Receiver system knows that the last column and last row are the parity bits.
Received data contains 3 errors.

```