**Lab Practical #15:**

Implementation of parity bit check Using C/Java language with example.

**Practical Assignment #15:**

**C/Java Program: Implementation of Bit stuffing Using C/Java language.**

1. **Enter the binary data: 011111101111110**

**Bit-stuffed data: 01111101011111010**

1. **Enter the binary data: 111110111111**

**Bit-stuffed data: 1 1 1 1 1 0 0 1 1 1 1 1 0 1**

1. Parity **Bit Check**

#include <stdio.h>

#include <string.h>

int main() {

char data[100];

int count = 0, i;

char parityType;

printf("Enter the binary data: ");

scanf("%s", data);

printf("Enter parity type (E for Even / O for Odd): ");

scanf(" %c", &parityType);

// Count number of 1s

for (i = 0; i < strlen(data); i++) {

if (data[i] == '1')

count++;

}

if (parityType == 'E' || parityType == 'e') {

if (count % 2 == 0)

printf("Parity Bit: 0 (Data already even)\n");

else

printf("Parity Bit: 1 (Added to make even)\n");

}

else if (parityType == 'O' || parityType == 'o') {

if (count % 2 == 0)

printf("Parity Bit: 1 (Added to make odd)\n");

else

printf("Parity Bit: 0 (Data already odd)\n");

}

else {

printf("Invalid parity type!\n");

}

return 0;

}

**2.Bit Stuffing**

#include <stdio.h>

#include <string.h>

int main() {

char data[100], stuffed[200];

int i, j = 0, count = 0;

printf("Enter the binary data: ");

scanf("%s", data);

for (i = 0; i < strlen(data); i++) {

stuffed[j++] = data[i];

if (data[i] == '1') {

count++;

if (count == 5) {

stuffed[j++] = '0'; // Stuff a 0 after five 1s

count = 0;

}

}

else {

count = 0;

}

}

stuffed[j] = '\0';

printf("Bit-stuffed data: %s\n", stuffed);

return 0;

}