**EXPERIMENT-33**

**IMPLEMENTATION OF BIT STUFFING MECHANISM USING C**

**Aim:**

To implement the bit stuffing mechanism using the C programming language.

**Software/Apparatus required:**

C compiler (e.g., GCC), Code editor (e.g., VS Code, Dev C++).

**Procedure:**

**Step 1: Understand the Bit Stuffing Mechanism**

1. Bit stuffing is a technique used in data communication to ensure that a specific pattern (e.g., five consecutive 1s) is not mistaken for a control signal.
2. If five consecutive 1s are detected, a 0 is stuffed (inserted) after them to differentiate the data from control signals.

**Step 2: Write the C Program**

1. Open a code editor and write the following C program to implement bit stuffing: **Step 3: Compile and Run the Program**

1. Save the program with a .c extension (e.g., bit\_stuffing.c).
2. Compile the program using a C compiler.
3. Run the compiled program
4. **Step 4: Analyze the Output**

The program will output the stuffed bit sequence.

For the input array {1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1}, the output will be:

111110111110

Here, a 0 is stuffed after every five consecutive 1s.

**Program**

#include <stdio.h>

#include <string.h>

// Function for bit stuffing void bitStuffing(int N, int arr[])

{

// Stores the stuffed array

int brr[30];

// Variables to traverse arrays int i, j, k; i = 0; j = 0;

// Loop to traverse in the range [0, N) while (i < N) {

// If the current bit is a set bit if (arr[i] == 1) {

// Stores the count of consecutive ones int count = 1;

// Insert into array brr[]

brr[j] = arr[i];

// Loop to check for

// next 5 bits

for (k = i + 1;

arr[k] == 1 && k < N && count < 5; k++) { j++; brr[j] = arr[k];

count++;

// If 5 consecutive set bits // are found insert a 0 bit if (count == 5) {

j++; brr[j] = 0;

} i = k;

}

}

// Otherwise insert arr[i] into

// the array brr[] else {

brr[j] = arr[i];

} i++; j++;

}

// Print Answer for (i = 0; i < j; i++) printf("%d", brr[i]);

}

// Driver Code int main() { int N = 12;

int arr[] = { 1, 1, 1, 1, 1, 1,1,1,1,1,1,1 };

bitStuffing(N, arr);

return 0;

}

**Output:**

111110111110

**Result:**

Thus, the bit stuffing mechanism was successfully implemented using the C programming language.