**EXPERIMENT 2**

## AIM:

## Write a program to implement byte stuffing.

**THEORY:**

* In variable-size framing at the data link layer, we need to define a way to separate one frame from the next. Byte stuffing is employed to accomplish the task.
* In byte stuffing an 8-bit flag (‘F’) is added at the beginning and at the end of the frame, thereby distinguishing one frame from the next.
* Therefore, every time a flag sequence (‘F’) is encountered, it signifies the beginning or end of a frame. This, ingenious scheme, however would give rise to a discrepancy, if the flag pattern (‘F’) would occur within the data carried by the frame itself.
* Byte stuffing comes to the rescue here, by stuffing the original data with an extra 8-bit escape sequence (‘E’) before the flag pattern, whenever it occurred within the data carried by a frame.
* The receiver would then have to de-stuff the escape sequence, in order to obtain the original data.

**CODE:**

#include<stdio.h>

#include<string.h>

main()

{

char a[30], fs[50] = " ", t[3], sd, ed, x[3], s[3], d[3], y[3];

int i, j, p = 0, q = 0;

printf("Enter characters to be stuffed:");

scanf("%s", a);

printf("\nEnter a character that represents starting delimiter:");

scanf(" %c", &sd);

printf("\nEnter a character that represents ending delimiter:");

scanf(" %c", &ed);

x[0] = s[0] = s[1] = sd;

x[1] = s[2] = '\0';

y[0] = d[0] = d[1] = ed;

d[2] = y[1] = '\0';

strcat(fs, x);

for(i = 0; i < strlen(a); i++)

{

t[0] = a[i];

t[1] = '\0';

if(t[0] == sd)

strcat(fs, s);

else if(t[0] == ed)

strcat(fs, d);

else

strcat(fs, t);

}

strcat(fs, y);

printf("\n After stuffing:%s", fs);

return 1;

}

**OUTPUT:**

