

## **Task 1: Implement the following Data Link Layer framing methods.**

### **NAME OF THE EXPERIMENT:1a). BIT STUFFING.**

**OBJECTIVE:** Implement the data link layer framing method.

**RESOURCE:** Code blocks

#### **PROGRAM LOGIC:**

The new technique allows data frames to contain an arbitrary number of bits and allows character codes with an arbitrary number of bits per character. Each frame begins and ends with special bit pattern, 01111110, called a flag byte. Whenever the sender's data link layer encounters five consecutive ones in the data, it automatically stuffs a 0 bit into the outgoing bit stream. This bit stuffing is analogous to character stuffing, in which a DLE is stuffed into the outgoing character stream before DLE in the data.

#### **1a.) Bit Stuffing code:**

```
#include <stdio.h>
#include<conio.h>
#include<string.h>
void main()
{
    int a[20],b[30],i,j,k,count,n;
    printf("Enter frame length:");
    scanf("%d",&n);
    printf("Enter input frame (0's & 1's only):"); for(i=0;i<n;i++)
    scanf("%d",&a[i]);
    i=0;
    count=1; j=0;
    while(i<n)
    {
        if(a[i]==1)
        {
            b[j]=a[i];
            for(k=i+1;a[k]==1 && k<n &&count<5;k++)
            {
                j++;
                b[j]=a[k]; count++; if(count==5)
                {
                    j++; b[j]=0;
                }
                i=k;
            }
        }
        else
```

```

{
b[j]=a[i];
} i++; j++;
}
printf("After stuffing the frame is:");
for(i=0;i<j;i++)
printf("%d",b[i]);
getch();
}

```

Output:

```

C:\Users\GRIET\Desktop\CodeBlocks\bits\bin\Debug\bits.exe
Enter frame length:10
Enter input frame (0's & 1's only):
1
0
1
0
1
1
1
1
1
1
1
After stuffing the frame is:101011111101_

```

## NAME OF THE EXPERIMENT: 1 b) CHARACTER STUFFING

**OBJECTIVE:** Implement the data link layer framing method.

**RESOURCE:** Codeblocks

### PROGRAM LOGIC:

The framing method gets around the problem of resynchronization after an error by having each frame start with the ASCII character sequence DLE STX and the sequence DLE ETX. If the destination ever loses the track of the frame boundaries all it has to do is look for DLE STX or DLE ETX characters to figure out. The data link layer on the receiving end removes the DLE before the data are given to the network layer. This technique is called character stuffing.

### PROGRAM FOR CHARACTER STUFFING

```

#include <stdio.h>
#include <stdlib.h>
#include<string.h>
#include<process.h>
void main()
{
int i=0,j=0,n,pos;char a[20],b[50],ch;

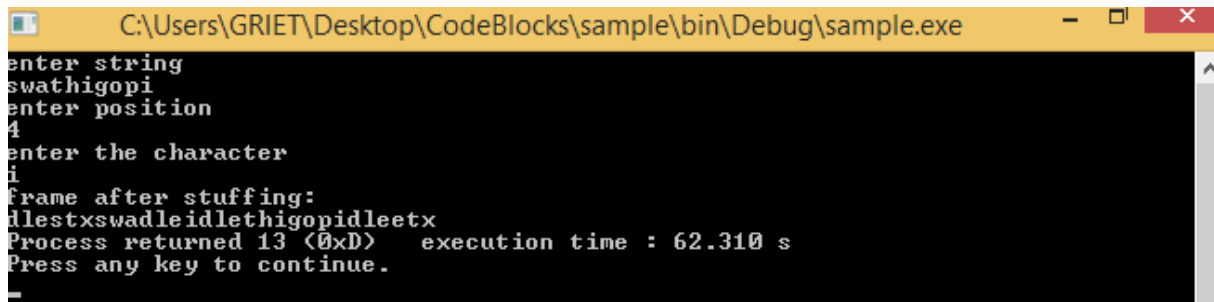
```

```

printf("enter string\n");
scanf("%s",&a);
n=strlen(a);
printf("enter position\n");
scanf("%d",&pos);
if(pos>n)
{
printf("invalid position, Enter again :");
scanf("%d",&pos);
}
printf("enter the character\n");
ch=getche();
b[0]='d';
b[1]='l';
b[2]='e';
b[3]='s';
b[4]='t';
b[5]='x'; j=6;
while(i<n)
{
if(i==pos-1)
{
b[j]='d';
b[j+1]='l';
b[j+2]='e';
b[j+3]=ch; b[j+4]='d';
b[j+5]='l';
b[j+6]='e';
j=j+7;
}
if(a[i]=='d' && a[i+1]=='l' && a[i+2]=='e')
{
b[j]='d';
b[j+1]='l';
b[j+2]='e';
j=j+3;
}
b[j]=a[i]; i++;
j++;
}
b[j]='d';
b[j+1]='l';
b[j+2]='e';
b[j+3]='e';
b[j+4]='t';
b[j+5]='x';
b[j+6]='\0';
printf("\nframe after stuffing:\n");
printf("%s",b);
getch(); }

```

Output:



```
C:\Users\GRIET\Desktop\CodeBlocks\sample\bin\Debug\sample.exe
enter string
swathigopi
enter position
4
enter the character
i
frame after stuffing:
dlestxswadleidlethigopidleetx
Process returned 13 (0xD) execution time : 62.310 s
Press any key to continue.
```

## NAME OF THE EXPERIMENT: 1.c) Character Count

**OBJECTIVE:** Implement the data link layer framing method.

**RESOURCE:** Code blocks

### PROGRAM LOGIC:

The framing method To count the number of characters present in the string, we will iterate through the string and count the characters. In above example, total number of characters present in the string are 19.

### Program for character count:

```
#include <stdio.h>
#include <string.h>

int main() {
    int n;
    printf("enter no of frames: ");
    scanf("%d", &n);
    int frames[n];
    for (int i = 0; i < n; i++)
    {
        printf("\nenter frame %d max_size: ", i+1);
        scanf("%d", &frames[i]);
    }
    char string [n][50];
    printf("enter the content for each frame: \n");
    for (int i = 0; i < n; i++)
    {
        printf("\nenter content for frame %d: ", i+1);
        scanf("%s", &string[i]);
        while (strlen(string[i]) > frames[i]){
            printf("entered string exceeding the maxsize!!\nenter again: ");
            scanf("%s", &string[i]);
        }
    }
    for (int i = 0; i < n; i++)
```

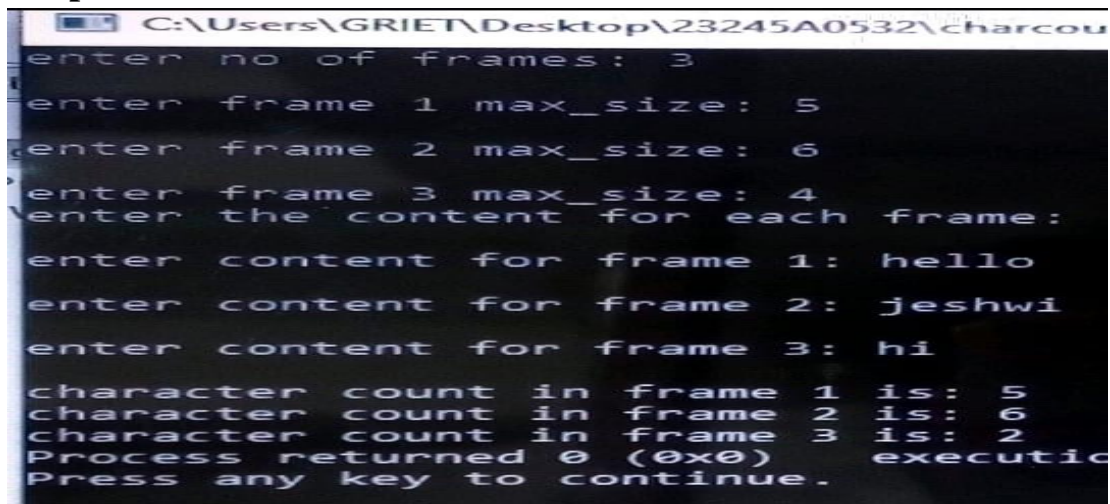
```

{
    int cnt = 0;
    for (int j = 0; j < strlen(string[i]); j++)
    {
        if (string[i][j] != ' ')
            cnt++;
    }
    printf("\ncharacter count in frame %d is: %d", i+1, cnt);
}

return 0;
}

```

## Output:



```

C:\Users\GRIET\Desktop\23245A0532\charcou
enter no of frames: 3
enter frame 1 max_size: 5
enter frame 2 max_size: 6
enter frame 3 max_size: 4
enter the content for each frame:
enter content for frame 1: hello
enter content for frame 2: jeshwi
enter content for frame 3: hi
character count in frame 1 is: 5
character count in frame 2 is: 6
character count in frame 3 is: 2
Process returned 0 (0x0) execution completed.
Press any key to continue.

```