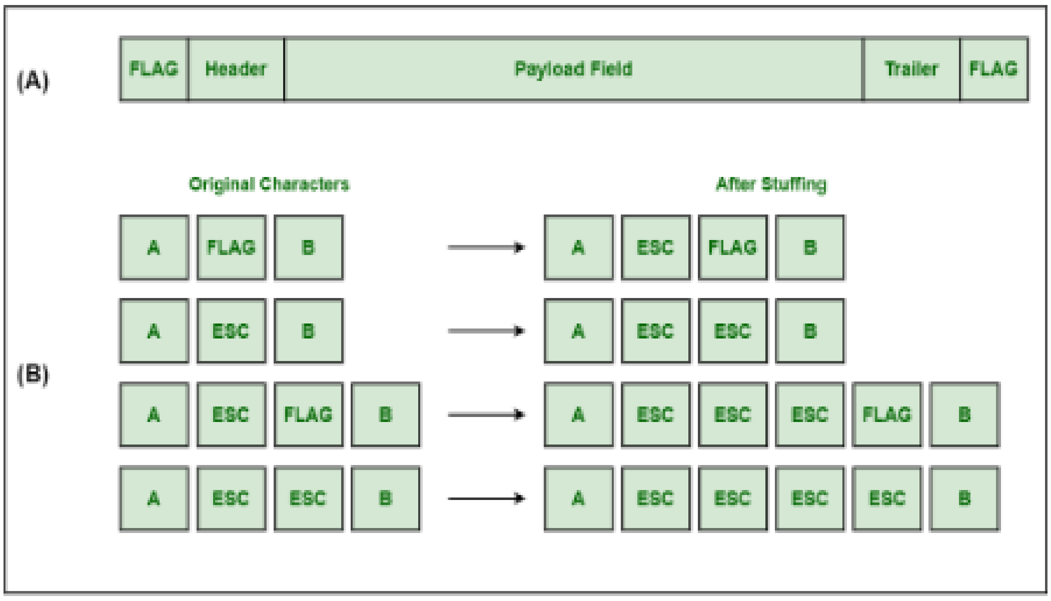
EXPERIMENT 2

2. i) Write a Program to implement the data link layer framing methods such as --> character stuffing.

Character Stuffing :

Character stuffing is also known as byte stuffing or character-oriented framing and is same as that of bit stuffing but byte stuffing actually operates on bytes whereas bit stuffing operates on bits. In byte stuffing, special byte that is basically known as ESC (Escape Character) that has predefined pattern is generally added to data section of the data stream or frame when there is message or character that has same pattern as that of flag byte.

But receiver removes this ESC and keeps data part that causes some problems or issues. In simple words, we can say that character stuffing is addition of 1 additional byte if there is presence of ESC or flag in text



A Character Stuffing
(A) A frame delimited by flag bytes
(B) Four examples of byte sequences before and after byte stuffing

PROGRAM:

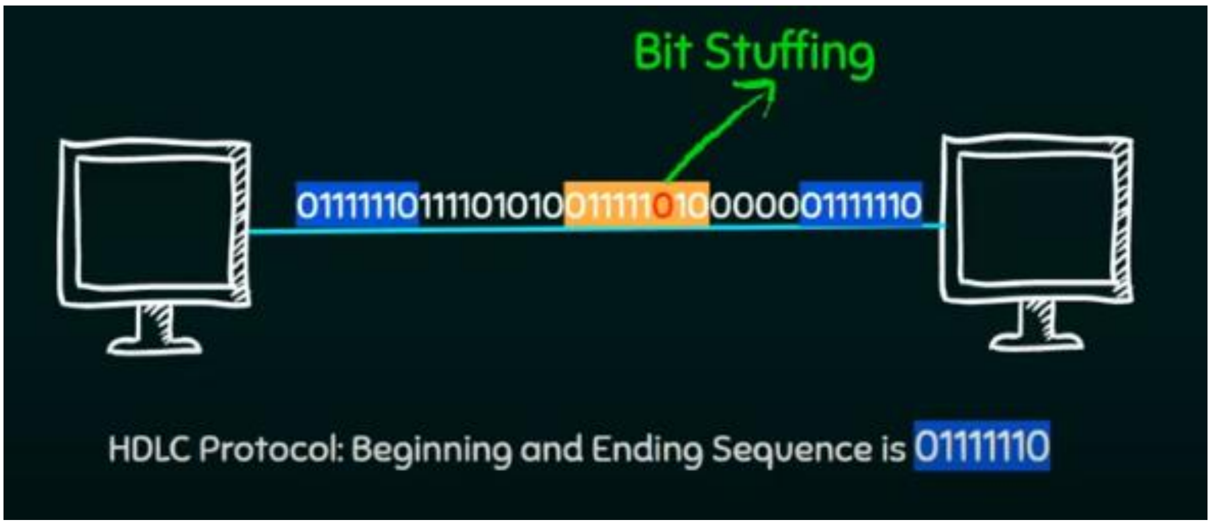
```
head = input("Enter character that represents the starting delimiter: ")
tail = input(" Enter character that represents the ending delimiter: ")
st = input("Enter the characters to be stuffed: ")
res=head
for i in st:
      if i==head or i ==tail:
             res = res + i + i
      else:
             res = res + i
res = res+tail
print("Frame after character stuffing: ", res)
```

OUTPUT:

```
Enter character that represents the starting delimiter: d

Enter character that represents the ending delimiter: g

Enter the characters to be stuffed: goodday

Frame after character stuffing: dggooddddayg
```

2. ii) Write a Program to implement the data link layer framing methods such as --> bit stuffing.

BitStuffing :
Bit stuffing is also known as bit-oriented framing or bit-oriented approach. In bit stuffing, extra bits are being added by network protocol designers to data streams. It is generally insertion or addition of extra bits into transmission unit or message to be transmitted as simple way to provide and give signaling information and data to receiver and to avoid or ignore appearance of unintended or unnecessary control sequences.



PROGRAM:

```
st = input ("Enter the frame: ") count = 0
res = ""
for i in st:
       if i == '1' and count < 5:
          res += '1'
          count += 1
       elif i == ' ':
          pass
       else:
          res += i count = 0
       if count == 5:
          res += '0'
          count= 0
print("Frame after bit stuffing: ", res)
```

OUTPUT:

```
Enter the frame: 01111110

Frame after bit stuffing: 011111010
```