|  |
| --- |
| 1. Bit stuffing and unstuffing:   #include <stdio.h>  #include <string.h>  #define MAX\_DATA 50  void bitStuffing(char \*input, char \*output)  {  int inputLength;  int count = 0;  int outputIndex = 0;  inputLength = strlen(input);  for (int i = 0; i < inputLength; i++)  {  if (input[i] == '1')  {  count++;  }  else  {  count = 0;  }  output[outputIndex++] = input[i];  if (count == 5)  {  output[outputIndex++] = '0'; // Corrected line  count = 0;  }  }  output[outputIndex] = '\0';  }  void bitUnstuffing(char \*input, char \*output)  {  int inputLength;  int count = 0;  int outputIndex = 0;  inputLength = strlen(input);  for (int i = 0; i < inputLength; i++)  {  if (input[i] == '1')  {  count++;  }  else  {  count = 0;  }  output[outputIndex++] = input[i];  if (count == 5)  {  output[outputIndex++] = '1'; // Corrected line  count = 0;  }  }  output[outputIndex] = '\0';  }  int main()  {  char input[MAX\_DATA];  char output[MAX\_DATA \* 2];  printf("Enter the stream of bits:");  scanf("%s", input);  if (strlen(input) > MAX\_DATA)  {  printf("Input exceeds maximum data length\n");  return 1;  }  bitStuffing(input, output);  printf("Stuffed bits are:%s\n", output);  bitUnstuffing(output, input);  printf("Unstuffed bits are:%s\n", input);  return 0;  }  Output:     1. Character stuffing or byte stuffing   #include<stdio.h>  #include<string.h>  #define FLAG 0X7E  int main()  {  unsigned char data[]={0x7D,0x55,0x7E,0x42};  int len=sizeof(data);  printf("Original Data:");  for(int i=0;i<len;i++)  {  printf("%02x ",data[i]);  }  printf("\nStuffed data:");  for(int j=0;j<len;j++)  {  if(data[j]==FLAG)  {  printf("%02x %02x ",FLAG,FLAG);  }  else  {  printf("%02x ",data[j]);  }  }  return 0;  }  Output: |