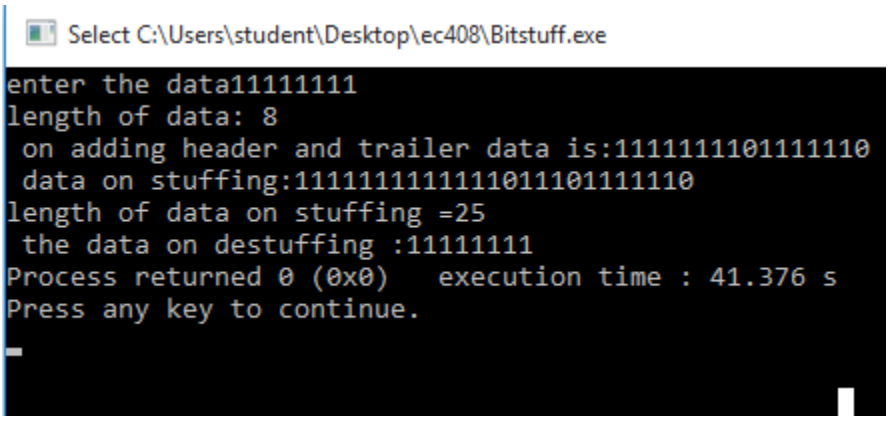
1.Bitstuffing

```
// C PROGRAM DEMONSTRATING STUFFING AND DESTUFFING OF DATA.
#include<stdio.h>
#include<string.h>
#include<stdlib.h>
int main()
{
   char data[100],data2[100],stuff[100],destuff[100];
   char h_t2[100],h_t[]={"01111110"};
   int i,a,n,n1,count=0,cnt=0;
   printf("enter the data");
   scanf("%s",data);
   n=strlen(data);
   printf("length of data: %d",n);
   strcpy(data2,data);
   strcpy(h_t2,h_t);
   strcat(data2,h_t2);
   strcat(h_t2,data2);
   //strcpy(data2,h_t2);
   printf("\n on adding header and trailer data is:%s",data2);

   //STUFFING DATA
   for(i=0,a=8;data2[i]!='\0';i++,a++)
   {
      if(data[i]=='1')
```

```
        count++;
      else
        count=0;

      if(count==5)
      {
        data2[a]=data[i];
        a++;
        data2[a]='0';
        count=0;
      }
      else
        data2[a]=data[i];
    }

    data2[a]='\0';
    strcat(data2,h_t);
    printf("\n data on stuffing:%s",data2);
    n1=strlen(data2);
    printf("\nlength of data on stuffing =%d",n1);
    strcpy(stuff,data2);
  //DESTUFFING
    for(i=0,a=8;a<n1-8;i++,a++)
    {
      if(stuff[a]=='1')
```

```
        cnt++;
      else
        cnt=0;
      if(cnt==5)
      {
        destuff[i]=stuff[a];
        a++;
        cnt=0;
      }
      else
        destuff[i]=stuff[a];
   }
   destuff[i]='\0';
   printf("\n the data on destuffing :%s",destuff);

}
```

1.Results



2.Bytestuffing

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

int check_control_characters(char data[])
{
  char ack[]={"ACK"},nak[]={"NAK"},stx[]={"STX"},etx[]={"ETX"},esc[]={"ESC"},dle[]={"DLE"},syn[]={"SYN"};
  if(strcmp(data,ack)==0)
      return 1;
  else if(strcmp(data,nak)==0)
      return 1;
  else if(strcmp(data,stx)==0)
      return 1;
  else if(strcmp(data,etx)==0)
      return 1;
  else if(strcmp(data,esc)==0)
      return 1;
  else if(strcmp(data,dle)==0)
      return 1;
  else if(strcmp(data,syn)==0)
      return 1;
  else
     return 0;
}
```

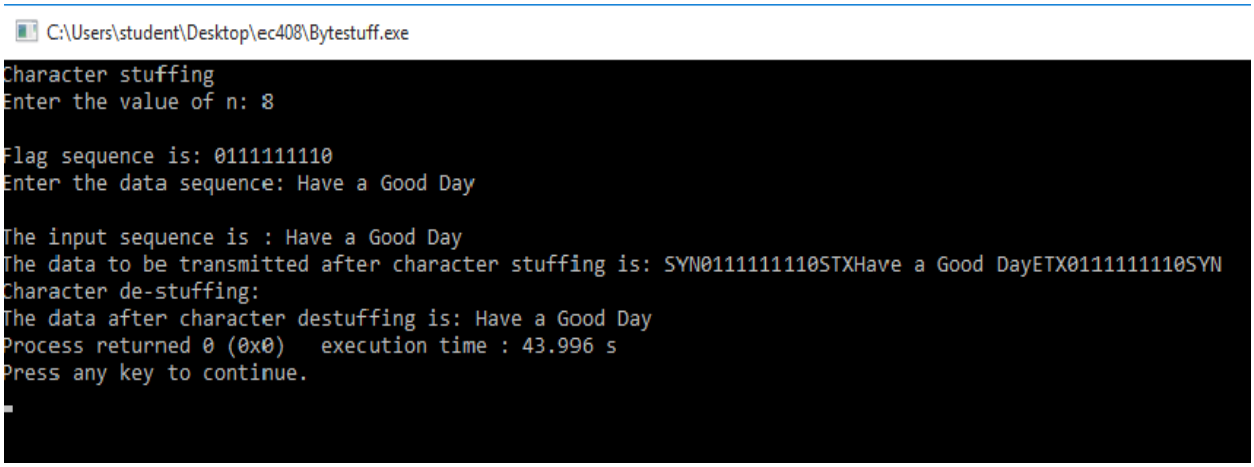
```
int main()
{
   char syn[]={"SYN"};
   char stx[]={"STX"};
   char etx[]={"ETX"};
   char del[]={"DLE"};
   char read_data[100],transmit_data[1000],three_characters[6],destuff_data[1000];
   char flag[100];
   char ch;
   int n,i,j,k,len_data,len_flag;
   printf("Character stuffing");
   printf("\nEnter the value of n: ");
   scanf("%d",&n);
   flag[0]=flag[n+1]='0';
   for(i=1;i<=n;i++)
      flag[i]='1';
   flag[n+2]='\0';
   len_flag=n+2;
   printf("\nFlag sequence is: %s",flag);
   fflush(stdin);
   printf("\nEnter the data sequence: ");
   i=0;
   do
   {
```

```
    ch=getchar();
    if(ch!='\n')
    {
        read_data[i++]=ch;
    }
}while(ch!='\n');
read_data[i]='\0';
printf("\nThe input sequence is : %s",read_data);
//Adding the SYN, header and STX values before the data
for(i=0;syn[i]!='\0';i++)
    transmit_data[i]=syn[i];
for(j=0;flag[j]!='\0';j++)
    transmit_data[i++]=flag[j];
for(j=0;stx[j]!='\0';j++)
    transmit_data[i++]=stx[j];
for(j=0;read_data[j]!='\0';j++)
{
    k=0;
    three_characters[k++]=read_data[j];
    three_characters[k++]=read_data[j+1];
    three_characters[k++]=read_data[j+2];
    three_characters[k]='\0';
    if(check_control_characters((three_characters)))
    {
        for(k=0;del[k]!='\0';k++)
```

```
            transmit_data[i++]=del[k];
        }
        transmit_data[i++]=read_data[j];
    }
    for(j=0;etx[j]!='\0';j++)
        transmit_data[i++]=etx[j];
    for(j=0;flag[j]!='\0';j++)
        transmit_data[i++]=flag[j];
    for(j=0;syn[j]!='\0';j++)
        transmit_data[i++]=syn[j];
    transmit_data[i]='\0';
    printf("\nThe data to be transmitted after character stuffing is: %s",transmit_data);
    //Character de-stuffing.
    printf("\nCharacter de-stuffing: ");
    len_data=strlen(transmit_data);
    for(j=0,i=6+(n+2);i<len_data-(6+n+2);j++,i++)
    {
        k=0;
        three_characters[k++]=transmit_data[i];
        three_characters[k++]=transmit_data[i+1];
        three_characters[k++]=transmit_data[i+2];
        three_characters[k]='\0';
        if(check_control_characters((three_characters)))
        {
            i=i+3;
```

```
    }
      destuff_data[j]=transmit_data[i];
  }
  destuff_data[j]='\0';
  printf("\nThe data after character destuffing is: %s",destuff_data);
  return 0;
}
```

2.Result



3.CRC

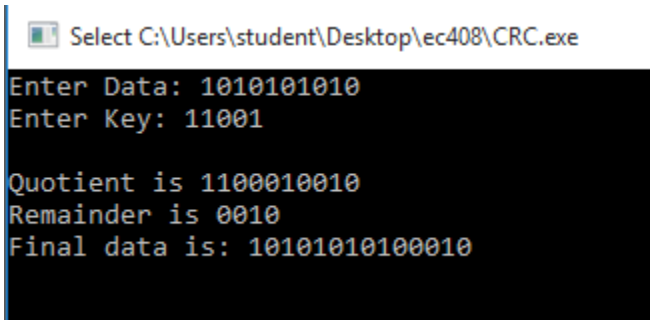
```
#include <stdio.h>
 #include <conio.h>
 #include <string.h>
 void main()
 {
        int i,j,keylen,msglen;
        char input[100], key[30],temp[30],quot[100],rem[30],key1[30];
```

```
printf("Enter Data: ");
gets(input);
printf("Enter Key: ");
gets(key);
keylen=strlen(key);
msglen=strlen(input);
strcpy(key1,key);
for (i=0;i<keylen-1;i++) {
        input[msglen+i]='0';
}
for (i=0;i<keylen;i++)
 temp[i]=input[i];
for (i=0;i<msglen;i++) {
        quot[i]=temp[0];
        if(quot[i]=='0')
         for (j=0;j<keylen;j++)
         key[j]='0'; else
         for (j=0;j<keylen;j++)
         key[j]=key1[j];
        for (j=keylen-1;j>0;j--) {
                if(temp[j]==key[j])
                 rem[j-1]='0'; else
                 rem[j-1]='1';
        }
        rem[keylen-1]=input[i+keylen];
```

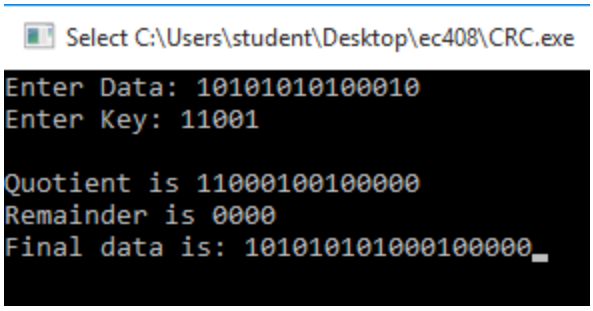
```
                strcpy(temp,rem);
        }
        strcpy(rem,temp);
        printf("\nQuotient is ");
        for (i=0;i<msglen;i++)
         printf("%c",quot[i]);
        printf("\nRemainder is ");
        for (i=0;i<keylen-1;i++)
         printf("%c",rem[i]);
        printf("\nFinal data is: ");
        for (i=0;i<msglen;i++)
         printf("%c",input[i]);
        for (i=0;i<keylen-1;i++)
         printf("%c",rem[i]);
        getch();
}
```

3.Result

Error detection(error is present)



Error correction(no error)


Select C:\Users\student\Desktop\ec408\CRC.exe
Enter Data: 10101010100010
Enter Key: 11001
Quotient is 11000100100000
Remainder is 0000
Final data is: 1010101010001000000