```
--------------------------------------------------------------------------------
--------------------------------------------------------------------------
/**List of Experiments **/
--------------------------------------------------------------------------------
--------------------------------------------------------------------------
1) Implement the data link layer framing methods such as character, character-
stuffing and bit
stuffing.

#include<stdio.h>
#include<stdlib.h>
#include<string.h>
#define N 256
char f[]="flag";
char e[]="esc";

char* bitStuffing(char data[],char *sd,int *n){
    int count=0,j=0,i=0,s=*n;
    while(i<s)
    {
        if(data[i]=='1')
            count++;
        else
          count=0;
        sd[j]=data[i];
        j++,i++;
        if(count==5)
        {
            count=0;
            n++;
            sd[j]='0';
            j++;
        }
    }
    sd[j]='\0';
    printf("stuffed data=%s",sd);
    return sd;
}
void bitUnstuffing(char sd[],char ud[],int n){
    int count=0,j=0,i=0;
    while(i<=n+1)
    {
        if(sd[i]=='1')
            count++;
        else
          count=0;
        ud[j]=sd[i];
        j++,i++;
        if(count==5)
        {
            count=0;
            i++;
        }
    }
    ud[j]='\0';
    printf("\nUnstuffed data=%s",ud);
}
void charStuffing(char d[],char s[])
{
```

```
    int i=0,j=0;
    while(i<strlen(d))
    {
         int c=0;
         if(d[i]=='f')
         {
             int k=i;
             for(int i2=0;i2<strlen(f);i2++)
                if(d[k]==f[i2])
                    k++,c++;
                else
                    break;
             if(c==strlen(f))
                for(int i2=0;i2<strlen(e);i2++)
                    s[j]=e[i2],j++;
         }
         else if(d[i]=='e')
         {
             int k=i;
             for(int i2=0;i2<strlen(e);i2++)
                if(d[k]==e[i2])
                    k++,c++;
                else
                     break;
             if(c==strlen(e))
                for(int i2=0;i2<strlen(e);i2++)
                    s[j]=e[i2],j++;
         }
         s[j]=d[i],i++,j++;
    }
    s[j]='\0';
     printf("stuffed data=%s",s);

}
void charUnstuffing(char s[],char u[])
{
    int i=0,j=0;
    while(i<strlen(s))
    {
         int c=0;
         if(s[i]=='e')
         {
             int k=i;
             for(int i2=0;i2<strlen(e);i2++)
             {
                  if(s[k]==e[i2])
                       k++,c++;
                  else
                     break;
             }
             if(c==strlen(e))
                i+=strlen(e);

         }
         u[j]=s[i],i++,j++;
    }
    u[j]='\0';
     printf("unstuffed data=%s",u);
```

```
}
int main(){
    int n;
    printf("Enter the size of data=");
    scanf("%d",&n);
    char data[n];

    printf("Enter the binary data=");
    scanf("%s",data);
    char *sd=(char*)calloc((n+n/5),sizeof(char)),ud[n];
    bitStuffing(data,sd,&n);
    bitUnstuffing(sd,ud,n);
    getchar();

   char d[N],s[N],u[N];
   printf("\nEnter the charecter data=");
   fgets(d,N,stdin);

   charStuffing(d,s);
   charUnstuffing(s,u);

}
```

---OUTPUT:

```
Enter the size of data=7
Enter the binary data=1111101
stuffed data=11111001
Unstuffed data=1111101
Enter the charecter data=flag escaped data flag
stuffed data=escflag escescaped data escflag
unstuffed data=flag escaped data flag
```

---

2) Write a program to compute CRC code for the polynomials CRC-12, CRC-16 and CRC CCIP

```
#include<stdio.h>
#include<stdlib.h>
void readInput(int a[],int n)
{
     for(int i=0;i<n;i++)
        scanf("%d",&a[i]);
}
void sender(int f[],int g[],int c[],int n,int k)
{
    int s=(n-n/k)-1,i=0;

    for(int j=0;j<n && i<s;j++)
    {
         int p=j;
         if(f[j]==1)
         {
             for(int w=0;w<k;w++)
             {
                f[p]=f[p]^g[w];
                p++;
             }
          }
```

```
          else if(f[j]==0)
          {
              for(int w=0;w<k;w++)
              {
                  f[p]=f[p]^0;
                  p++;
              }
          }
          i++;
     }
     for(int i=n-1,rev=k-2;rev>=0;rev--)
     {
            c[rev]=f[i];
            i--;
     }
}
int receiver(int f[],int g[],int n,int k)
{
    int s=(n-n/k)-1,i=0;

    for(int j=0;j<n & i<s;j++)
    {
         int p=j;
         if(f[j]==1)
         {
             for(int w=0;w<k;w++)
             {
                 f[p]=f[p]^g[w];
                 p++;
             }
          }
          else if(f[j]==0)
          {
              for(int w=0;w<k;w++)
              {
                  f[p]=f[p]^0;
                  p++;
              }
          }
          i++;
     }
     int count=0;
     for(int i=n-1;i>n-k-1;i--)
     {
            if(f[i]==0)
              count++;
     }
     if(count==k)
        return 1;
     else
        return 0;

}
void AddCrcToDataBit(int data[],int crc[],int m,int n)
{
      int  j=m-2;
       for(int i=n-1;i>n-m;i--)
       {
            data[i]=data[i]|crc[j];
```

```
            j--;
        }
}
void printOut(int a[], int n)
{
    for(int i=0;i<n;i++)
       printf("%d",a[i]);
    printf("\n");
}
int main()
{
  int s1,s2;
  printf("Enter the frame size=");
  scanf("%d",&s1);
  printf("\nEnter the generator size=");
  scanf("%d",&s2);

  int
*f1=(int*)calloc(s1+s2,sizeof(int)),*f2=(int*)calloc(s1+s2,sizeof(int)),g[s2];
  printf("\nEnter the frame=");
  readInput(f1,s1);
  for(int i=0;i<s1;i++)
     f2[i]=f1[i];
  printf("\nEnter the generator=");
  readInput(g,s2);

  int crc[s2-1];
  sender(f2,g,crc,s1,s2);

  printf("\nYour crc remender:");
  printOut(crc,s2-1);

  int n=s1+s2-1,flag=0;
  printf("Before adding redundancy:");
  printOut(f1,n);
  AddCrcToDataBit(f1,crc,s2,n);
  printf("After adding redundancy:");
  printOut(f1,n);

  flag=receiver(f1,g,n,s2);
  (flag==1)?printf("\nRecevied CORRECT"):printf("\nReceived ERROR");

}
```

```
OUTPUT:1
Enter the frame size=8

Enter the generator size=5

Enter the frame=1 1 1 0 0 0 1 1

Enter the generator=1 1 0 0 1

Your crc remender:0000
Before adding redundancy:111000110000
After adding redundancy :111000110000
```

```
Recevied CORRECT

OUTPUT:2
Enter the generator size=3

Enter the frame=1 1 0 0 1

Enter the generator=1 0 1

Your crc remender:10
Before adding redundancy:1100100
After adding redundancy :1100110

Recevied CORRECT
```

---

3) Develop a simple data link layer that performs the flow control using the sliding window protocol,
and loss recovery using the Go-Back-N mechanism.

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

int f=0,r=0;
void frameSeq(int F[],int m)
{
   for(int i=0;i<m;i++)
        F[i]=i;
}
void enqueue(int F[],int s[],int n)
{
    if(f==r)
    {
        for(int i=0;i<n;i++)
        {
            s[i]=F[i];
            f++;
        }
    }
    else
    {
        for(int i=0;i<n-1;i++)
           s[i]=s[i+1];
        s[n-1]=F[f];
        f++;
    }
}
void dequeue(int R[],int s[],int m,int *i)
{
      if(f==r){
         printf("frame is Empty.");
      }
      else if(f<m)
      {
         R[r]=s[*i];
         r++;
      }
      else
      {
```

```
         R[r]=s[*i];
         r++;
         (*i)++;
      }
}
void receiveSeq(int R[],int m)
{
   for(int i=0;i<m;i++)
      printf("%d\t",R[i]);
    printf("\n");
}
int main()
{
    int n,m;
    printf("enter total no of frame=");
    scanf("%d",&m);
    printf("enter the window size=");
    scanf("%d",&n);
    int F[m],s[n],R[m];
    frameSeq(F,m);
    int i=0;
    printf("sliding window:\n");
    while(r<m)
    {
        if(f<m)
           enqueue(F,s,n);  //sender slide
        receiveSeq(s,n);
        dequeue(R,s,m,&i);  //receiver slide
    }
    (f==r)?printf("Data received\n"):printf("Data Interrupted\n");
    if(f==r)receiveSeq(R,m);
}
```

OUTPUT:

```
enter total no of frame=10
enter the window size=3
sliding window:
0       1       2
1       2       3
2       3       4
3       4       5
4       5       6
5       6       7
6       7       8
7       8       9
7       8       9
7       8       9
Data received
0       1       2       3       4       5       6       7       8       9
```

---

4) Implement Dijsktra’s algorithm to compute the shortest path through a network

```
#include<stdio.h>
#include<conio.h>
#define INFINITY 9999
```

```
void readAdjMat(int n,int G[][n])
{
   for(int i=0;i<n;i++)
     for(int j=0;j<n;j++)
        scanf("%d",&G[i][j]);
}
void createCostMat(int n,int G[][n],int C[][n])
{
    for(int i=0;i<n;i++)
      for(int j=0;j<n;j++)
         if(G[i][j]==-1)
            C[i][j]=INFINITY;
         else
            C[i][j]=G[i][j];
}
void dijkstra(int n,int G[][n],int start)
{

   int cost[n][n],distance[n],pred[n];
   int visited[n],count,min,next,i,j;
   //pred[] stores the predecessor of each node
   //count gives the number of nodes seen so far
   createCostMat(n,G,cost);
   //initialize pred[],distance[] and visited[]
   for(i=0;i<n;i++)
   {
     distance[i]=cost[start][i];
     pred[i]=start;
     visited[i]=0;
   }
   distance[start]=0;
   visited[start]=1;
   count=1;
   while(count<n-1)
   {
     min=INFINITY;
     //nextnode gives the node at minimum distance
     for(i=0;i<n;i++)
        if(distance[i]<min&&!visited[i])
        {
            min=distance[i];
            next=i;
        }
     //check if a better path exists through nextnode
     visited[next]=1;
     for(i=0;i<n;i++)
        if(!visited[i])
            if(min+cost[next][i]<distance[i])
            {
                distance[i]=min+cost[next][i];
                pred[i]=next;
            }
      count++;
    }
    //print the path and distance of each node
    for(i=0;i<n;i++)
        if(i!=start)
        {
            printf("\nDistance of node%d=%d",i,distance[i]);
```

```
            printf("\nPath=%d",i);
            j=i;
            do
            {
                j=pred[j];
                printf("<-%d",j);
            }while(j!=start);
        }
    printf("\n");
    for(i=0;i<n;i++)
        printf("%d\t",pred[i]);
    printf("\n");
    for(i=0;i<n;i++)
        printf("%d\t",visited[i]);
     printf("\n");
     for(i=0;i<n;i++)
        printf("%d\t",distance[i]);
    printf("\n");
}
int main()
{
    int n;
    printf("Enter no. of vertices:");
    scanf("%d",&n);
    int G[n][n],s;
    printf("\nEnter the adjacency matrix:\n");
    readAdjMat(n, G);
    start:
        printf("\nEnter the starting node:");
        scanf("%d",&s);
        dijkstra(n,G,s);
    goto start;
    return 0;
}
```

---OUTPUT-1:

```
Enter no. of vertices:4

Enter the adjacency matrix:
0 3 1 -1
3 0 5 2
1 5 0 -1
-1 2 -1 0

Enter the starting node:0

Distance of node1=3
Path=1<-0
Distance of node2=1
Path=2<-0
Distance of node3=5
Path=3<-1<-0
0       0       0       1 (perdecessor)
1       1       1       0 (visited)
0       3       1       5 (distance)

Enter the starting node:1
```

```
Distance of node0=3
Path=0<-1
Distance of node2=4
Path=2<-0<-1
Distance of node3=2
Path=3<-1
1       1       0       1
1       1       0       1
3       0       4       2

Enter the starting node:2

Distance of node0=1
Path=0<-2
Distance of node1=4
Path=1<-0<-2
Distance of node3=6
Path=3<-1<-0<-2
2       0       2       1
1       1       1       0
1       4       0       6

Enter the starting node:3

Distance of node0=5
Path=0<-1<-3
Distance of node1=2
Path=1<-3
Distance of node2=6
Path=2<-0<-1<-3
1       3       0       3
1       1       0       1
5       2       6       0

OUTPUT-2:

Enter no. of vertices:6

Enter the adjacency matrix:
0 3 2 5 -1 -1
3 0 -1 1 4 -1
2 1 0 2 -1 1
5 1 2 0 3 -1
-1 4 -1 3 0 2
-1 -1 1 -1 2 0

Enter the starting node:0

Distance of node1=3
Path=1<-0
Distance of node2=2
Path=2<-0
Distance of node3=4
Path=3<-2<-0
Distance of node4=5
Path=4<-5<-2<-0
Distance of node5=3
Path=5<-2<-0
0       0       0       2       5       2 (perdecessor)
```

```
1       1       1       1       0       1 (visited)
0       3       2       4       5       3 (distance)

Enter the starting node:1

Distance of node0=3
Path=0<-1
Distance of node2=3
Path=2<-3<-1
Distance of node3=1
Path=3<-1
Distance of node4=4
Path=4<-1
Distance of node5=4
Path=5<-2<-3<-1
1       1       3       1       1       2
1       1       1       1       1       0
3       0       3       1       4       4

Enter the starting node:2

Distance of node0=2
Path=0<-2
Distance of node1=1
Path=1<-2
Distance of node3=2
Path=3<-2
Distance of node4=3
Path=4<-5<-2
Distance of node5=1
Path=5<-2
2       2       2       2       5       2
1       1       1       1       0       1
2       1       0       2       3       1

Enter the starting node:3

Distance of node0=4
Path=0<-1<-3
Distance of node1=1
Path=1<-3
Distance of node2=2
Path=2<-3
Distance of node4=3
Path=4<-3
Distance of node5=3
Path=5<-2<-3
1       3       3       3       3       2
0       1       1       1       1       1
4       1       2       0       3       3

Enter the starting node:4

Distance of node0=5
Path=0<-2<-5<-4
Distance of node1=4
Path=1<-4
Distance of node2=3
Path=2<-5<-4
```

```
Distance of node3=3
Path=3<-4
Distance of node5=2
Path=5<-4
2       4       5       4       4       4
0       1       1       1       1       1
5       4       3       3       0       2

Enter the starting node:5

Distance of node0=3
Path=0<-2<-5
Distance of node1=2
Path=1<-2<-5
Distance of node2=1
Path=2<-5
Distance of node3=3
Path=3<-2<-5
Distance of node4=2
Path=4<-5
2       2       5       2       5       5
1       1       1       0       1       1
3       2       1       3       2       0
--------------------------------------------------------------------------------
------------------------------------------------------------------------
```

5) Take an example subnet of hosts and obtain a broadcast tree for the subnet

```
#include <stdio.h>
#define INFINITY 99999

typedef struct node
{
    int cost, path;
}host;
void readAdjMat(int n,int G[][n])
{
   for(int i=0;i<n;i++)
     for(int j=0;j<n;j++)
        scanf("%d",&G[i][j]);
}
int findMinCostIndex(int n, host H[],int visited[])
{
    int min = INFINITY,index;

    for (int i= 0; i<n;i++)
        if (visited[i]==0 && H[i].cost< min)
        {
            min = H[i].cost;
            index=i;
        }
    return index;
}
int sumOfAllMinCost(int n,host H[])
{
    int sum=0;
    for(int i=0;i<n;i++)
        sum+=H[i].cost;
    return sum;
}
```

```
void printBroadcastTree( int n,host H[])
{
    printf("Edge \tWeight\n");
    for (int i = 0; i < n; i++)
      if(H[i].path>=0)
        printf("%d - %d \t%d \n", H[i].path, i,H[i].cost);
}
void primMST(int n,int s,int G[n][n],host H[])
{
    int visited[n];

    for (int i= 0; i<n;i++)
    {
        H[i].cost =INFINITY;
        visited[i]=0;
    }
    H[s].cost= 0;
    H[s].path=-1;

    for (int i= 0; i < n; i++)
    {
        int u =findMinCostIndex(n,H,visited);
        visited[u] = 1;
        for (int v = 0; v < n; v++)
          if (G[u][v] && visited[v]==0 && G[u][v] < H[v].cost)
                H[v].path= u , H[v].cost= G[u][v];
    }
}
int main()
{

    int n,sum=0;
    printf("Enter no. of vertices:");
    scanf("%d",&n);
    int G[n][n],s;
    printf("\nEnter the adjacency matrix:\n");
    readAdjMat(n,G);
    host H[n];
    label:
      printf("\nEnter the starting node:");
      scanf("%d",&s);
      primMST(n,s,G,H);
      sum=sumOfAllMinCost(n,H);
      printf("\nMinimium Spanning cost:%d\n",sum);
      printBroadcastTree(n,H);
    goto label;
    return 0;
}
```

---OUTPUT:

```
Enter no. of vertices:5

Enter the adjacency matrix:
0 2 0 6 0
2 0 3 8 5
0 3 0 0 7
6 8 0 0 9
```

```
0 5 7 9 0

Enter the starting node:0

Minimium Spanning cost:16
Edge    Weight
0 - 1   2
1 - 2   3
0 - 3   6
1 - 4   5

Enter the starting node:1

Minimium Spanning cost:16
Edge    Weight
1 - 0   2
1 - 2   3
0 - 3   6
1 - 4   5

Enter the starting node:2

Minimium Spanning cost:16
Edge    Weight
1 - 0   2
2 - 1   3
0 - 3   6
1 - 4   5

Enter the starting node:4

Minimium Spanning cost:16
Edge    Weight
1 - 0   2
4 - 1   5
1 - 2   3
0 - 3   6
```

---

6) Implement distance vector routing algorithm for obtaining routing tables at each node.

```
#include<stdio.h>
#define N 64
typedef struct node {
      unsigned dist[N];
      unsigned from[N];
}Router;
void readDistMat(int n,int M[][n],Router R[])
{
   for(int i=0;i<n;i++)
   {
       for(int j=0;j<n;j++)
       {
         scanf("%d",&M[i][j]);
         if(i==j) M[i][j]=0;
         R[i].dist[j]=M[i][j];
         R[i].from[j]=j;
       }
```

```
   }
}
void vectorRouting(int n,int M[][n],Router R[])
{
   int count;
   do{
      count=0;
      for(int i=0;i<n;i++)
        for(int j=0;j<n;j++)
          for(int k=0;k<n;k++)
            if(R[i].dist[j]>M[i][k]+R[k].dist[j])
                {
              R[i].dist[j]=R[i].dist[k]+R[k].dist[j];
              R[i].from[j]=k;
              count++;
            }
      }while(count!=0);
}
void DisplayShrtDist(int n,int m[][n],Router R[])
{
    for(int i=0;i<n;i++)
   {
       printf("Router-%d\n",i+1);
       for(int j=0;j<n;j++)
       {
          printf("Node:%d Via:%d Dist:%d\n",j+1,R[i].from[j]+1,R[i].dist[j]);
       }
       printf("\n");
   }
}
int main()
{
    int n;
    printf("Enter total no.of router=");
    scanf("%d",&n);
    Router rt[n];
    int M[n][n];
    printf("Enter a distance matric:\n");
    readDistMat(n,M,rt);
    vectorRouting(n,M,rt);
    printf("\nState value of each Router:\n");
    DisplayShrtDist(n,M,rt);
}
```

OUTPUT:

```
Enter total no.of router=4
Enter a distance matric:
0 3 5 99
3 0 99 1
5 4 0  2
99 1 2 0

State value of each Router:
Router-1
Node:1 Via:1 Dist:0
Node:2 Via:2 Dist:3
Node:3 Via:3 Dist:5
```

```
Node:4 Via:2 Dist:4

Router-2
Node:1 Via:1 Dist:3
Node:2 Via:2 Dist:0
Node:3 Via:4 Dist:3
Node:4 Via:4 Dist:1

Router-3
Node:1 Via:1 Dist:5
Node:2 Via:4 Dist:3
Node:3 Via:3 Dist:0
Node:4 Via:4 Dist:2

Router-4
Node:1 Via:2 Dist:4
Node:2 Via:2 Dist:1
Node:3 Via:3 Dist:2
Node:4 Via:4 Dist:0
--------------------------------------------------------------------------------
--------------------------------------------------------------------------7)
Implement data encryption and data decryption

#include<stdio.h>
#include<string.h>
#define N 256

void encryptionData(char D[],int key)
{
    int i=0;
    while(i<strlen(D))
    {
        D[i]=D[i]-key;
        i++;
    }
    D[i]='\0';
    printf("Encrypted data=%s",D);

}
void decryptionData(char D[],int key)
{
    int i=0;
    while(i<strlen(D))
    {
        D[i]=D[i]+key;
        i++;
    }
    D[i]='\0';
    printf("\nDecrypted data=%s",D);
}
int main()
{
   int key;
   char D[N];
   printf("\nEnter the key =");
   scanf("%d",&key);
   getchar();

   printf("\nEnter the charecter data=");
```

```
   fgets(D,N,stdin);

   encryptionData(D,key);
   decryptionData(D,key);

}
--------------------------------------------------------------------------------
------------------------------------------------------------------------OUTPUT:

Enter the key =2

Enter the charecter data=jaipal
Encrypted data=h_gn_j
Decrypted data=jaipal

Enter the key size=7

Enter the charecter data=jaipal
Encrypted data=cZbiZe
Decrypted data=jaipal
--------------------------------------------------------------------------------
-------------------------------------------------------------------------
8) Write a program for congestion control using Leaky bucket algorithm

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

int f=0,r=0;

void enqueFullBucket(int B[],int n)
{
   for(int i=0;i<n;i++)
   {
      if(r>n)
        printf("BUCKET OVERFLOW");
      else
      {
        printf("Enter the Data-%d=",i+1);
        scanf("%d",&B[r]);
        r++;
      }
   }
}
int dequeue(int B[])
{
    int data;
    if(f>=r)
        printf("BUCKET UNDERFLOW");
    else
    {
        data=B[f];
        f++;
        return data;
    }
}
int checkValidDataSize(int B[],int n,int s)
{
    int count=0;
    for(int i=0;i<n;i++)
```

```
      if(s>=B[i])
        count++;
    return count;
}
int main()
{
    int n,s;
    printf("enter the Bucket size=");
    scanf("%d",&n);
    printf("enter the data rate size=");
    scanf("%d",&s);
    int B[n];
    enqueFullBucket(B,n);
    int i=1,dataRt=s,packet,flag=1;

    if(n==checkValidDataSize(B,n,s))
    {
        if(flag)
        {
            printf("\nTime\t Packets\tstatus\n");
            flag=0;
        }
        while(f<r)
        {
          if(s>=B[f])
          {
            s=s-B[f];
            packet=dequeue(B);
            printf("%ds\t  %dMbps\tAccepted\n",i,packet);
          }
          else
          {
            i++;
            s=dataRt;
          }
        }
    }
    else
    {
        printf("ERROR OF DATA SIZE");
    }
    (f==r)?printf("Data received\n"):printf("\n3Data Interrupted\n");
}
```

---OUTPUT 1:

```
enter the Bucket size=6
enter the data rate size=1000
Enter the Data-1=200
Enter the Data-2=400
Enter the Data-3=450
Enter the Data-4=500
Enter the Data-5=700
Enter the Data-6=200

Time     Packets        status
1s        200Mbps       Accepted
1s        400Mbps       Accepted
2s        450Mbps       Accepted
```

```
2s        500Mbps       Accepted
3s        700Mbps       Accepted
3s        200Mbps       Accepted
Data received

OUTPUT 2:

enter the Bucket size=5
enter the data rate size=500
Enter the Data-1=500
Enter the Data-2=500
Enter the Data-3=500
Enter the Data-4=500
Enter the Data-5=500

Time     Packets        status
1s        500Mbps       Accepted
2s        500Mbps       Accepted
3s        500Mbps       Accepted
4s        500Mbps       Accepted
5s        500Mbps       Accepted
Data received
--------------------------------------------------------------------------------
--------------------------------------------------------------------------
9)Write a program for frame sorting technique used in buffers

#include <stdio.h>
#include<stdlib.h>
#define N 256

typedef struct node{
   char msg[N];
   int seq;
}Frame;

void readFrameSeq(Frame B[],int n)
{
   for(int i=0;i<n;i++)
   {
      printf("Enter the message-%d=",i+1);
      scanf("%s",&B[i].msg);
      B[i].seq=rand()%n;
   }
}
void printFrameSeq(Frame B[],int n)
{
   for(int i=0;i<n;i++)
      printf("%d\t%s\n",B[i].seq,B[i].msg);
    printf("\n");
}
void sortFrames(Frame B[],int n)
{
    for(int i=0;i<n-1;i++)
      for(int j=0;j<n-i-1;j++)
         if(B[j].seq>B[j+1].seq)
         {
            Frame swap=B[j];
            B[j]=B[j+1];
            B[j+1]=swap;
```

```
        }
}
int main()
{
    int n;
    printf("Enter total no of frame=");
    scanf("%d",&n);
    Frame Buffer[n];
    readFrameSeq(Buffer,n);
    sortFrames(Buffer,n);
    printf("Sorted frames:\n");
    printFrameSeq(Buffer,n);
}
```

OUTPUT:

```
Enter total no of frame=5
Enter the Data-1=debo
Enter the Data-2=jai
Enter the Data-3=yesh
Enter the Data-4=vikas
Enter the Data-5=jntu
Sorted frames:
0       vikas
1       debo
2       jai
4       yesh
4       jntu
```