

EXPERIMENT 2

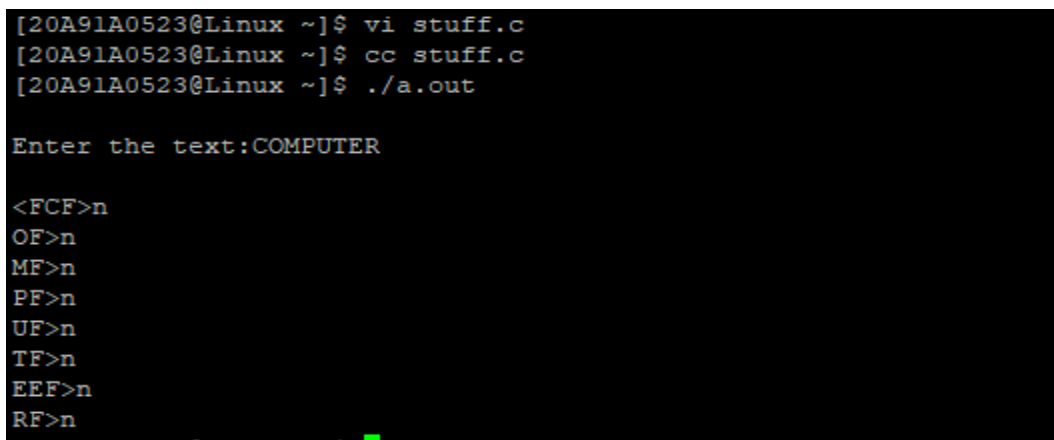
Write a Program to implement the data link layer farming methods such as

i) Character stuffing ii) bit stuffing

i)Character Stuffing

```
#include<stdio.h>
#include<stdlib.h>
#include<string.h>
int main()
{
char txt[30],app;
int position,i;
printf("\nEnter the text:");
scanf("%[^\n]",txt);
printf("\n<F>");
for(i=0;i<strlen(txt);i++)
{
if(txt[i]=='E' || txt[i]=='F')
{
printf("E%c",txt[i]);
}
else
{
printf("%c",txt[i]);
}
printf("F>\n\n");
}
return 0;
}
```

Output:



```
[20A91A0523@Linux ~]$ vi stuff.c
[20A91A0523@Linux ~]$ cc stuff.c
[20A91A0523@Linux ~]$ ./a.out

Enter the text:COMPUTER

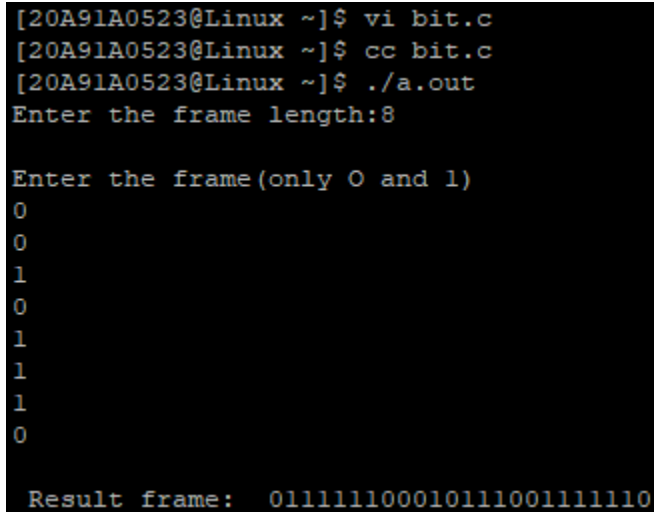
<FCF>\n
OF>\n
MF>\n
PF>\n
UF>\n
TF>\n
EEF>\n
RF>\n
```

ii)Bit Stuffing

```
#include<stdio.h>
#include<stdlib.h>
int main()
{
int frame[30],result[30],len,j,k=0,i,sum=0;
printf("Enter the frame length:");
scanf("%d",&len);
printf("\nEnter the frame(only 0 and 1)\n");
i=0;
while(i<len)
{
scanf("%d",&frame[i]);
i++;
}
i=0;
while(i<len)
{
j=0;
sum=0;
while(j<5&& i<len){
if(frame[i]==0)
{
result[k]=frame[i];
i++;
k++;
break;
}
else
{
sum+=frame[i];
result[k]=frame[i];
i++;
j++;
k++;
}
}
if(sum==5)
{
result[k]=0;
k++;
}
}
printf("\n Result frame:");
```

```
printf("\t01111110");  
for(i=0;i<k;i++)  
{  
printf("%d",result[i]);  
}  
printf("01111110\n");  
return 0;  
}
```

Output:



```
[20A91A0523@Linux ~]$ vi bit.c  
[20A91A0523@Linux ~]$ cc bit.c  
[20A91A0523@Linux ~]$ ./a.out  
Enter the frame length:8  
  
Enter the frame (only 0 and 1)  
0  
0  
1  
0  
1  
1  
1  
1  
1  
0  
  
Result frame: 011111100010111001111110
```

EXPERIMENT-3

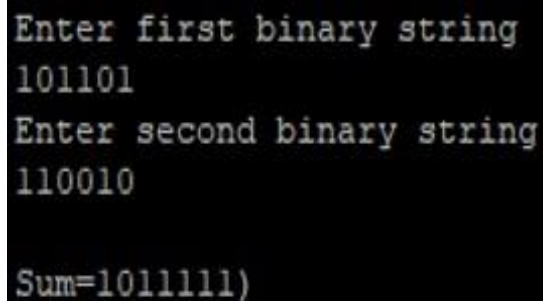
Write a Program to implement data link layer farming method checksum.

Program:

```
#include<stdio.h>
#include<string.h>
int main()
{
    char a[20],b[20];
    char sum[20],complement[20];
    int i,length;
    printf("Enter first binary string\n");
    scanf("%s",&a);
    printf("Enter second binary string\n");
    scanf("%s",&b);
    if(strlen(a)==strlen(b))
    {
        length = strlen(a);
        char carry='0';
        for(i=length-1;i>=0;i--)
        {
            if(a[i]=='0' && b[i]=='0' && carry=='0') {
                sum[i]='0';
                carry='0';
            }
            else if(a[i]=='0' && b[i]=='0' && carry=='1')
            {
                sum[i]='1';
                carry='0';
            }
            else if(a[i]=='0' && b[i]=='1' && carry=='0')
            {
                sum[i]='1';
                carry='0';
            }
            else if(a[i]=='0' && b[i]=='1' && carry=='1')
            {
                sum[i]='0';
                carry='1';
            }
            else if(a[i]=='1' && b[i]=='0' && carry=='0')
            {
                sum[i]='1';
                carry='0';
            }
            else if(a[i]=='1' && b[i]=='0' && carry=='1')
            {
                sum[i]='0';
                carry='1';
            }
        }
    }
}
```

```
}  
else if(a[i]=='1' && b[i]=='0' && carry=='1') {  
    sum[i]='0';  
    carry='1';  
}  
else if(a[i]=='1' && b[i]=='1' && carry=='0') {  
    sum[i]='0';  
    carry='1';  
}  
else if(a[i]=='1' && b[i]=='1' && carry=='1') {  
    sum[i]='1';  
    carry='1';  
}  
else  
    break;  
}  
printf("\nSum=%c%s",carry,sum);  
for(i=0;i<length;i++) {  
    if(sum[i]=='0')  
        complement[i]='1';  
    else  
        complement[i]='0';  
}  
if(carry=='1')  
    carry='0';  
else  
    carry='1';  
printf("\nChecksum=%c%s",carry,complement);  
}  
else {  
    printf("\nWrong input strings");  
}  
}}
```

Output:



```
Enter first binary string  
101101  
Enter second binary string  
110010  
  
Sum=1011111)
```

EXPERIMENT-4

Write a program for Hamming Code generation for error detection and correction.

Program:

```
#include<stdio.h>
void main() {
    int data[10],dataatrec[10],c,c1,c2,c3,i;
    printf("Enter 4 bits of data one by one\n");
    scanf("%d%d%d%d",&data[0], &data[1], &data[2], &data[4]);
    data[6]=data[0]^data[2]^data[4];
    data[5]=data[0]^data[1]^data[4];
    data[3]=data[0]^data[1]^data[2];
    printf("The encoded data is \n");
    for(i=0;i<7;i++)
        printf("%d",data[i]);
    printf("\n\nEnter received data bits one by one\n");
    for(i=0;i<7;i++)
        scanf("%d",&dataatrec[i]);
    c1=dataatrec[6]^dataatrec[4]^dataatrec[2]^dataatrec[0];
    c2=dataatrec[5]^dataatrec[4]^dataatrec[1]^dataatrec[0];
    c3=dataatrec[3]^dataatrec[2]^dataatrec[1]^dataatrec[0]; c=c3*4+c2*2+c1;
    if(c==0) {
        printf("\nNo error while transmission of data\n");
    }
    else {
        printf("\nError on position %d",c);
        printf("\nData sent: ");
        for(i=0;i<7;i++)
            printf("%d",dataatrec[i]);
        printf("\nCorrect message is \n");
        if(dataatrec[7-c]==0)
            dataatrec[7-c]=1;
        else dataatrec[7-c]=0;
        for(i=0;i<7;i++) {
            printf("%d",dataatrec[i]);
        }
    }
}
```

Output:

```
CA\Users\admin\Desktop\hammingcode.exe
Enter 4 bits of data one by one
1
0
1
0
The encoded data is
1010010

Enter received data bits one by one
1
0
1
0
1
1
1
0

Error on position 3
Data sent: 1010110
Correct message is
1010010
-----
Process exited after 19.51 seconds with return value 1
Press any key to continue . . .
```

EXPERIMENT-5

Write a Program to implement on a data set of characters the three CRC polynomials – CRC 12, CRC 16 and CRC CCIP.

Program:

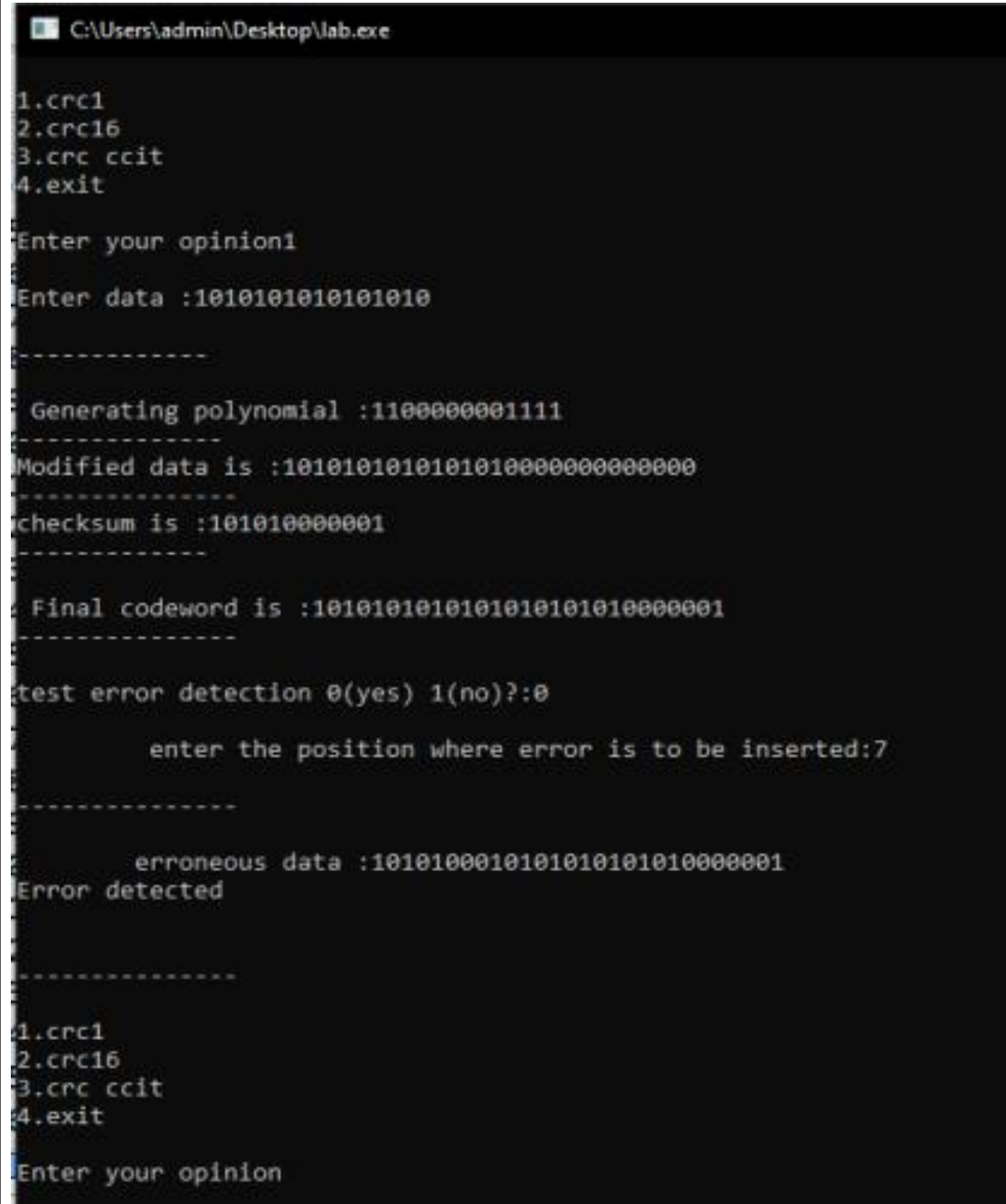
```
#include<stdio.h>
#include<string.h>
#define N strlen(g)
char t[50],cs[50],g[50];
int a,e,c,b;
void xor()
{
for(c=1;c<N;c++)
cs[c]=((cs[c]==g[c])?'0':'1');
}
void crc()
{
for(e=0;e<N;e++)
cs[e]=t[e];
do
{
if(cs[0]=='1')
xor();
for(c=0;c<N-1;c++)
cs[c]=cs[c+1];
cs[c]=t[e++];
}while(e<=a+N-1);
}
int main()
{
int flag=0;
do
{
printf("\n1.crc1\n2.crc16\n3.crc ccit\n4.exit\n\nEnter your opinion");
scanf("%d",&b);
switch(b)
{
case 1:strcpy(g,"1100000001111");
break;
```



```
case 2:strcpy(g,"110000000000000101");
break;
case 3:strcpy(g,"100010000000100001");
break;
case 4:return 0;
}
printf("\nEnter data :");
scanf("%s",t);
printf("\n-----\n");
printf("\n Generating polynomial :%s",g);
a=strlen(t);
for(e=a;e<a+N-1;e++)
t[e]='0';
printf("\n-----\n");
printf("Modified data is :%s",t);
printf("\n-----\n");
crc();
printf("checksum is :%s",cs);
for(e=a;e<a+N-1;e++)
t[e]=cs[e-a];
printf("\n-----\n");
printf("\n Final codeword is :%s",t);
printf("\n-----\n");
printf("\ntest error detection 0(yes) 1(no)?:"");
scanf("%d",&e);
if(e==0){
do
{
printf("\n\t enter the position where error is to be inserted:");
scanf("%d",&e);
}
while(e==0 | e>a+N-1);
t[e-1]=(t[e-1]=='0')?'1':'0';
printf("\n-----\n");
printf("\n\t erroneous data :%s\n",t);
}
crc();
for(e=0;(e<N-1)&&(cs[e]!='1');e++);
```

```
if(e<N-1)
printf("Error detected\n\n");
else
printf("No error detected \n\n");
printf("\n-----\n");
}while(flag!=1);
}
```

Output:



```
C:\Users\admin\Desktop\lab.exe

1.crc1
2.crc16
3.crc ccit
4.exit

Enter your opinion1
Enter data :1010101010101010

-----

Generating polynomial :1100000001111
-----
Modified data is :101010101010101000000000000000
-----
checksum is :101010000001
-----

Final codeword is :1010101010101010101010000001
-----

test error detection 0(yes) 1(no)?:0

        enter the position where error is to be inserted:7
-----

        erroneous data :1010100010101010101010000001
Error detected

-----

1.crc1
2.crc16
3.crc ccit
4.exit

Enter your opinion
```

EXPERIMENT-6

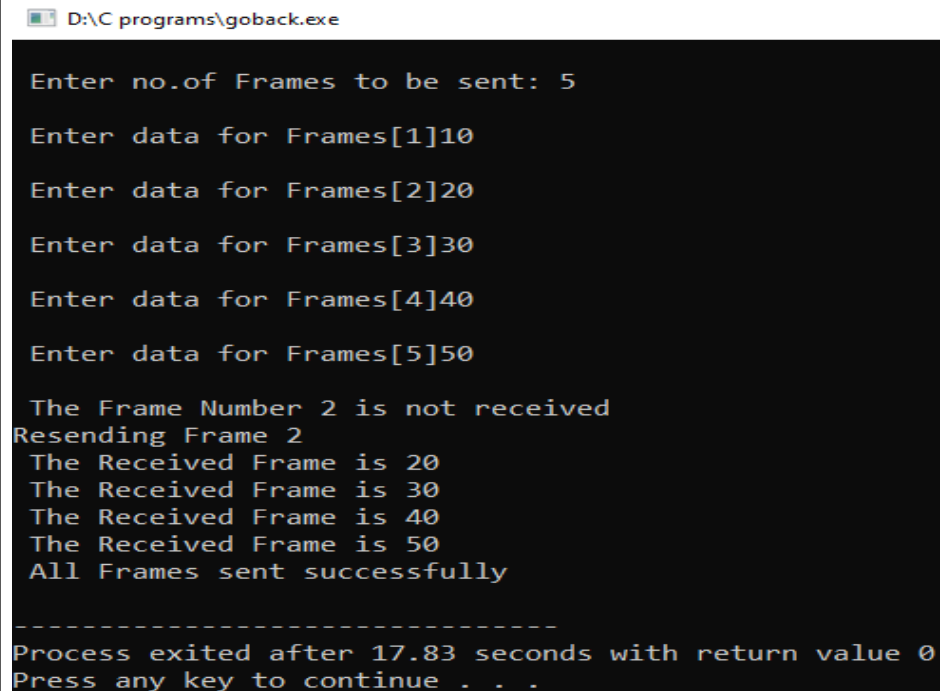
Write a Program to implement Sliding window protocol for Goback N.

Program:

```
#include<stdio.h>
#include<stdlib.h>
#include<math.h>
#include<unistd.h>
int n,r;
struct frame
{
    char ack;
    int data;
}frm[10];
int sender(void);
void recvack(void);
void resend_gb(void);
int main()
{
    int c;
    sender();
    recvack();
    resend_gb();
    printf("\n All Frames sent successfully\n");
}
int sender()
{
    int i;
    printf("\n Enter no.of Frames to be sent: ");
    scanf("%d",&n);
    for(i=1;i<=n;i++)
    {
        printf("\n Enter data for Frames[%d]",i);
        scanf("%d",&frm[i].data);
        frm[i].ack='y';
    }
    return 0;
}
void recvack()
```

```
{
    int i;
    rand();
    r=rand()%n;
    frm[r].ack='n';
    for(i=1;i<=n;i++)
    {
        if(frm[i].ack=='n')
            printf("\n The Frame Number %d is not received",r);
    }
}
void resend_gb()
{
    int i;
    printf("\nResending Frame %d",r);
    for(i=r;i<=n;i++)
    {
        sleep(2);
        frm[i].ack='y';
        printf("\n The Received Frame is %d",frm[i].data);
    }
}
```

Output:



```
D:\C programs\goback.exe
Enter no.of Frames to be sent: 5
Enter data for Frames[1]10
Enter data for Frames[2]20
Enter data for Frames[3]30
Enter data for Frames[4]40
Enter data for Frames[5]50
The Frame Number 2 is not received
Resending Frame 2
The Received Frame is 20
The Received Frame is 30
The Received Frame is 40
The Received Frame is 50
All Frames sent successfully
-----
Process exited after 17.83 seconds with return value 0
Press any key to continue . . .
```

EXPERIMENT-7

Write a Program to implement Sliding window protocol for Selective repeat.

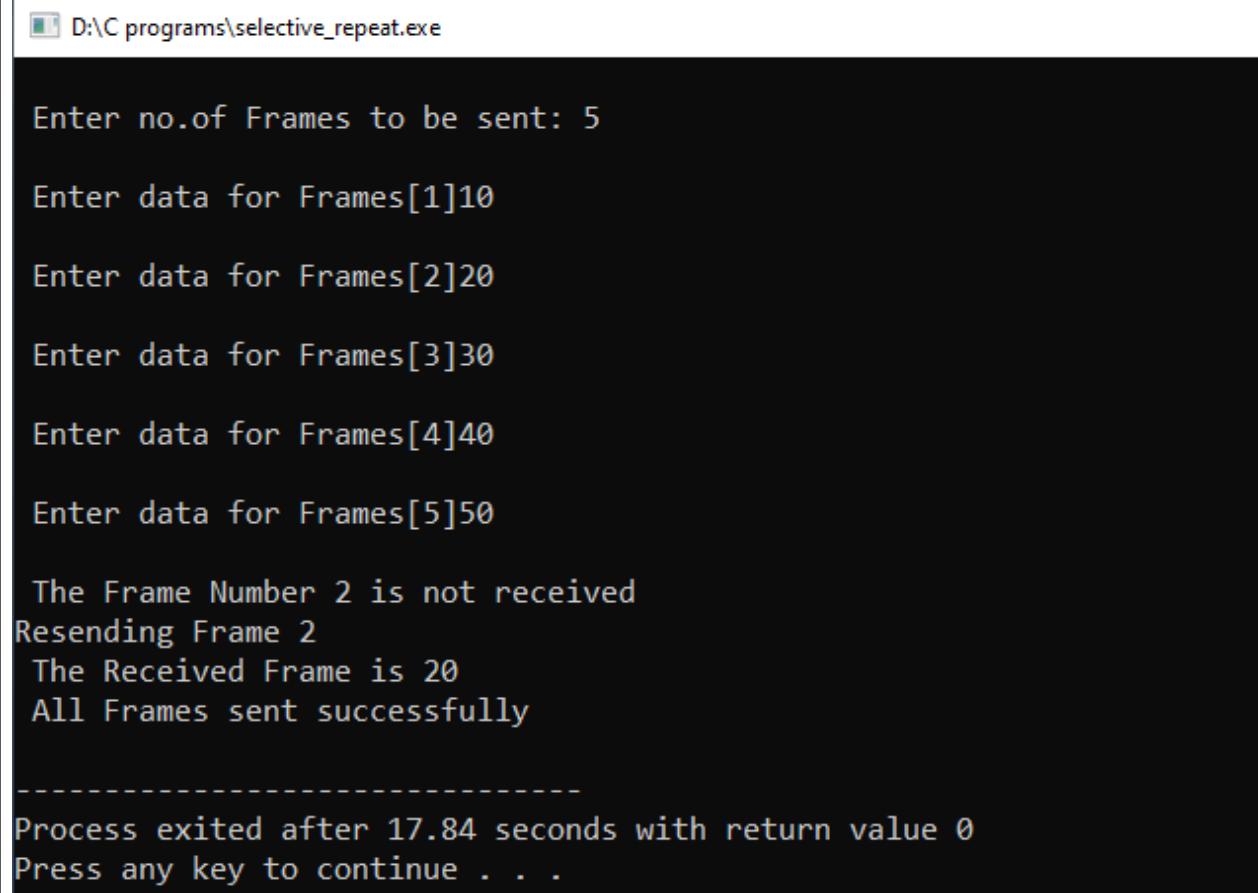
Program:

```
#include<stdio.h>
#include<stdlib.h>
#include<math.h>
#include<unistd.h>
int n,r;
struct frame
{
    char ack;
    int data;
}frm[10];
int sender(void);
void recvack(void);
void resend_sr(void);
int main()
{
    int c;
    sender();
    recvack();
    resend_sr();
    printf("\n All Frames sent successfully\n");
}
int sender()
{
    int i;
    printf("\n Enter no.of Frames to be sent: ");
    scanf("%d",&n);
    for(i=1;i<=n;i++)
    {
        printf("\n Enter data for Frames[%d]",i);
        scanf("%d",&frm[i].data);
        frm[i].ack='y';
    }
    return 0;
}
void recvack()
```

```
{
    int i;
    rand();
    r=rand()%n;
    frm[r].ack='n';
    for(i=1;i<=n;i++)
    {
        if(frm[i].ack=='n')
            printf("\n The Frame Number %d is not received",r);
    }
}

void resend_sr()
{
    printf("\nResending Frame %d",r);
    sleep(2);
    frm[r].ack='y';
    printf("\n The Received Frame is %d",frm[r].data);
}
```

Output:



```
D:\C programs\selective_repeat.exe

Enter no.of Frames to be sent: 5

Enter data for Frames[1]10

Enter data for Frames[2]20

Enter data for Frames[3]30

Enter data for Frames[4]40

Enter data for Frames[5]50

The Frame Number 2 is not received
Resending Frame 2
The Received Frame is 20
All Frames sent successfully

-----
Process exited after 17.84 seconds with return value 0
Press any key to continue . . .
```

EXPERIMENT-8

Write a Program to implement Stop and Wait Protocol.

Program:

```
#include<stdio.h>
#include<stdlib.h>
#include<math.h>
#include<unistd.h>
int main()
{
int i,j,packet[30];
int fsize=(rand()%16)+1;
printf("\n\t Frame Size : %d \n",fsize);
printf("\n\t -----Data Log ----- \n");
printf("\n FRAME\t \tDATA\t \tWAITING\t \tACKNOW\t \tRESEND\t");
for(i=0;i<fsize;i++)
{
packet[i]=rand()%1000;
printf("\n %d \t\t %d",i+1,packet[i]);
while(j=0 || rand()%2==0)
{
sleep(1);
printf("\t1 ");
sleep(1);
for(j=2;rand()%2==0&& j<4;j++)
{
printf("%d ",j);
sleep(1);
}
if(j==4)
printf("\t\t\t NO \tRESENDING...\n %d \t %d",i+1,packet[i]);
else
break;
}
if(j==0)
{
sleep(1);
printf("\t0 ");
}
}
```

```
printf("\t\t\t YES \t\t NO");  
}  
  
printf("\n\n-----ALL DATA PACKETS SEND -----\\n");  
return 0;  
}
```

Output:



```
C:\Users\admin\Desktop\stop_and_wait.exe  
  
Frame Size : 10  
  
-----Data Log -----  
  
FRAME      DATA      WAITING      ACKNOW      RESEND  
1           467      1 2          YES         NO  
2           724      1 2 3        NO          RESENDING...  
2           724      0            YES         NO  
3           145      0            YES         NO  
4           827      0            YES         NO  
5           491      0            YES         NO  
6           942      0            YES         NO  
7           436      0            YES         NO  
8           604      1            YES         NO  
9           292      1            YES         NO  
10          716      1            YES         NO  
  
-----ALL DATA PACKETS SEND -----  
  
-----  
Process exited after 19.29 seconds with return value 0  
Press any key to continue . . .
```


EXPERIMENT-9

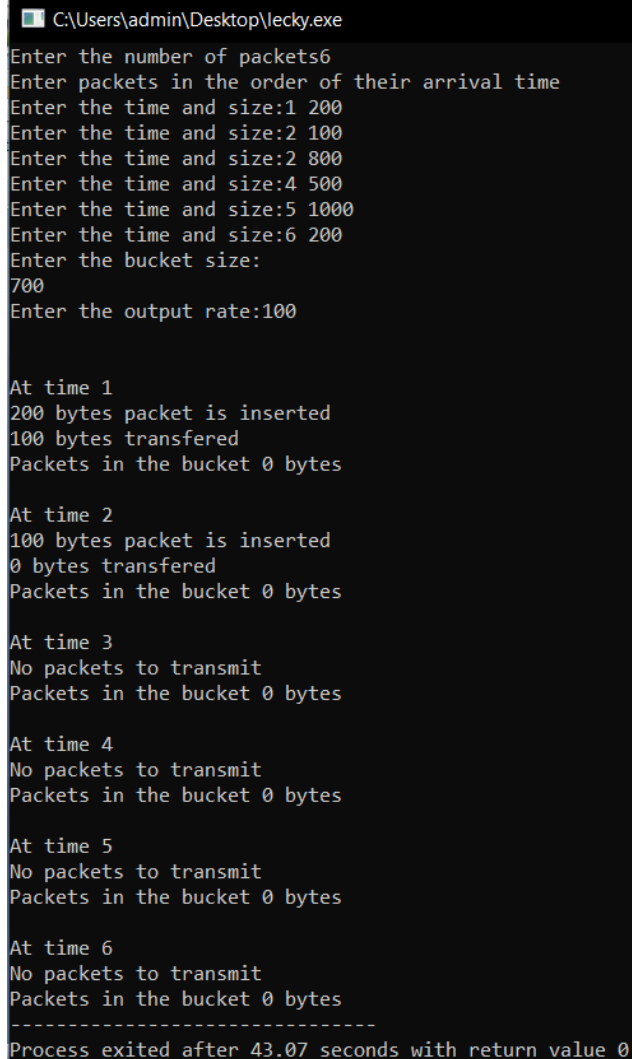
Write a program for congestion control using leaky bucket algorithm

Program:

```
#include<stdio.h>
#include<stdlib.h>
struct packet{
int time;
int size;
}p[50];
int main(){
int i,n,m,k=0;
int bsize,bfilled,outrate;
printf("Enter the number of packets");
scanf("%d",&n);
printf("Enter packets in the order of their arrival time\n");
for(i=0;i<n;i++){
printf("Enter the time and size:");
scanf("%d%d",&p[i].time,&p[i].size);
}
printf("Enter the bucket size:");
scanf("%d",&bsize);
printf("Enter the output rate:");
scanf("%d",&outrate);
m=p[n-1].time;
i=1;
k=0;
bfilled=0;
while(i<=m || bfilled!=0){
printf("\n\nAt time %d",i);
if(p[k].time==i){
if(bsize>=bfilled + p[k].size){
bfilled=bfilled+p[k].size;
printf("\n%d bytes packet is inserted",p[k].size);
k=k+1;
}
else{
printf("\n%d bytes packet is discarded",p[k].size);
k=k+1;
}
```

```
}  
}  
if(bfilled==0){  
printf("\nNo packets to transmit");  
}  
else if(bfilled>=outrate){  
bfilled=bfilled-outrate;  
printf("\n%d bytes transfered",bfilled);  
bfilled=0;  
}  
printf("\nPackets in the bucket %d bytes",bfilled);  
i++;  
}}
```

Output



```
C:\Users\admin\Desktop\lecky.exe  
Enter the number of packets:6  
Enter packets in the order of their arrival time  
Enter the time and size:1 200  
Enter the time and size:2 100  
Enter the time and size:2 800  
Enter the time and size:4 500  
Enter the time and size:5 1000  
Enter the time and size:6 200  
Enter the bucket size:  
700  
Enter the output rate:100  
  
At time 1  
200 bytes packet is inserted  
100 bytes transfered  
Packets in the bucket 0 bytes  
  
At time 2  
100 bytes packet is inserted  
0 bytes transfered  
Packets in the bucket 0 bytes  
  
At time 3  
No packets to transmit  
Packets in the bucket 0 bytes  
  
At time 4  
No packets to transmit  
Packets in the bucket 0 bytes  
  
At time 5  
No packets to transmit  
Packets in the bucket 0 bytes  
  
At time 6  
No packets to transmit  
Packets in the bucket 0 bytes  
-----  
Process exited after 43.07 seconds with return value 0
```

EXPERIMENT-10

Write a Program to implement Dijkstra's algorithm to compute the Shortest path through a graph.

Program:

```
#include<stdio.h>
#include<stdlib.h>
#define MAX 30
#define LARGE 1000
struct state{
int len;
int pre;
int label;
};
struct state states[MAX];
int a[MAX][MAX];
int main(){
int i,j,s,d,n,tem,min,mini;
printf("Enter no.of vertices:");
scanf("%d",&n);
printf("\nEnter adjacency matrix\n");
/*Reading the adjacency matrix*/
for(i=0;i<n;i++)
for(j=0;j<n;j++){
printf("a[%d][%d]=",i,j);
scanf("%d",&a[i][j]);
}
/*Initially marking all nodes in the graph as temporary*/
for(i=0;i<n;i++){
states[i].len=LARGE;
states[i].label=0;
states[i].pre=-1;
}
printf("\nEnter source vertex:");
scanf("%d",&s);
printf("\nEnter destination vertex:");
scanf("%d",&d);
states[d].len=0;
states[d].label=1;
```

```
tem=d;
while(tem!=s){
for(i=0;i<n;i++){
if(a[tem][i]!=0 && states[tem].len+a[tem][i]<states[i].len &&
states[i].label==0){
states[i].len=states[tem].len+a[tem][i];
states[i].pre=tem;
}
}
min=LARGE;
mini=0;
/*Now find the vertex with the smallest length from all the nodes that are
temporary in the graph*/
for(i=0;i<n;i++){
if(states[i].len<LARGE && states[i].label==0){
min=states[i].len;
mini=i;
}
}
states[mini].label=1;
tem=mini;
}
tem=s;
printf("\nPath length:%d",states[s].len);
printf("\nPath\n");
printf("%d",tem);
do{
tem=states[tem].pre;
printf("%d",tem);
}while(tem!=d);
return 0;
}
```

Output

```
enter no.of vertices:6

enter adjacency matrix
a[0][0]=0
a[0][1]=7
a[0][2]=9
a[0][3]=0
a[0][4]=0
a[0][5]=14
a[1][0]=7
a[1][1]=0
a[1][2]=10
a[1][3]=15
a[1][4]=0
a[1][5]=0
a[2][0]=9
a[2][1]=10
a[2][2]=0
a[2][3]=11
a[2][4]=0
a[2][5]=2
a[3][0]=0
a[3][1]=0
a[3][2]=15
a[3][3]=11
a[3][4]=0
a[3][5]=6
a[4][0]=0
a[4][1]=0
a[4][2]=0
a[4][3]=0
a[4][4]=0
a[4][5]=9
a[5][0]=14
a[5][1]=0
a[5][2]=2
a[5][3]=0
a[5][4]=9
a[5][5]=0

enter source vertex:0

enter destination vertex:4

path length:21
path
0254
```

EXPERIMENT-11

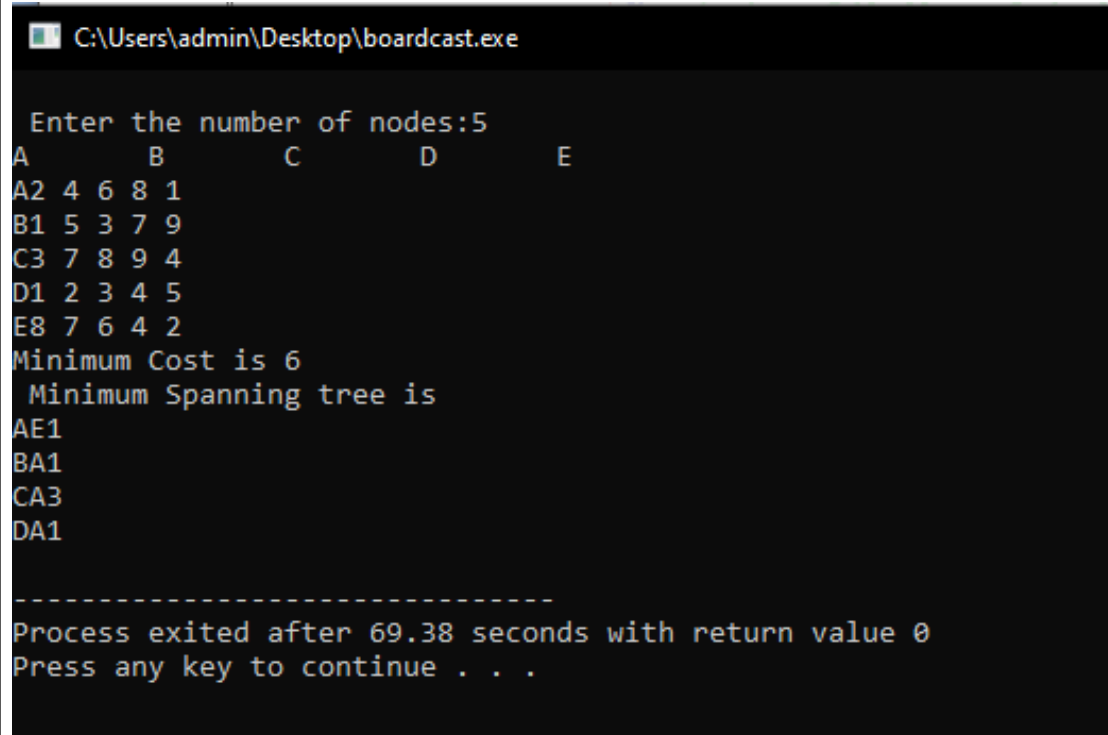
Write a Program to implement Distance vector routing algorithm by obtaining routing table at each node (Take an example subnet graph with weights indicating delay between nodes).

Program:

```
#include<stdio.h>
struct node{
unsigned dist[20];
unsigned from[20];
}rt[10];
int main(){
int costmat[20][20];
int nodes,i,j,k,count=0;
printf("\nEnter the number of nodes:");
scanf("%d",&nodes);
printf("\nEnter the cost matrix (999 for no link):\n");
for(i=0;i<nodes;i++){
for(j=0;j<nodes;j++){
scanf("%d",&costmat[i][j]);
costmat[i][i]=0;
rt[i].dist[j]=costmat[i][j];
rt[j].from[j]=j;
}
}
do{
count=0;
for(i=0;i<nodes;i++)
for(j=0;j<nodes;j++)
for(k=0;k<nodes;k++){
if(rt[i].dist[j]>costmat[i][k]+rt[k].dist[j]){
rt[i].dist[j]=rt[i].dist[k]+rt[k].dist[j];
rt[i].from[j]=k;
count++;
}
}while(count!=0);
for(i=0;i<nodes;i++){
printf("\n\nFor router %d\n",i+1);
```

```
for(j=0;j<nodes;j++){  
printf("\t\nnode %d via %d Distance %d ",j+1,rt[i].from[j]+1,rt[i].dist[j]);  
}  
}  
printf("\n\n");  
return 0;  
}
```

Output:



```
C:\Users\admin\Desktop\boardcast.exe  
  
Enter the number of nodes:5  
A      B      C      D      E  
A2 4 6 8 1  
B1 5 3 7 9  
C3 7 8 9 4  
D1 2 3 4 5  
E8 7 6 4 2  
Minimum Cost is 6  
Minimum Spanning tree is  
AE1  
BA1  
CA3  
DA1  
  
-----  
Process exited after 69.38 seconds with return value 0  
Press any key to continue . . .
```

EXPERIMENT-12

Write a Program to implement Broadcast tree by taking subnet of hosts.

Program:

```
#include<stdio.h>
int p,q,u,v,n;
int min=99,mincost=0;
int t[50][2],i,j;
int parent[50],edge[50][50];
int find(int r);
void sunion(int a,int b);
int main()
{
printf("\n Enter the number of nodes:");
scanf("%d",&n);
for(i=0;i<n;i++)
{
printf("%c\t",65+i);
parent[i]=-1;
}
printf("\n");
for(i=0;i<n;i++)
{
printf("%c",65+i);
for(j=0;j<n;j++)
scanf("%d",&edge[i][j]);
}
for(i=0;i<n;i++)
{
for(j=0;j<n;j++)
if(edge[i][j]!=99)
if(min>edge[i][j])
{
min=edge[i][j];
u=i;
v=j;
}
}
```

Exp No:

Date: COMPUTER NETWORKS LAB

Aditya Engineering College Page No: Roll No:

```
p=find(u);
q=find(v);
if(p!=q)
{
t[i][0]=u;
t[i][1]=v;
mincost=mincost+edge[u][v];
union(p,q);
}
else
{
t[i][0]=-1;
t[i][1]=-1;
}
min=99;
}
printf("Minimum Cost is %d \n Minimum Spanning tree is \n",mincost);
for(i=0;i<n;i++)
if(t[i][0]!=-1 && t[i][1]!=-1)
{
printf("%c%c%d",65+t[i][0],65+t[i][1],edge[t[i][0]][t[i][1]]);
printf("\n");
}
return 0;
}
int find(int l)
{
if(parent[l]>0)
l=parent[l];
return l;
}
void union(int l,int m)
{
parent[l]=m;
}
```

Output:

```
C:\Users\admin\Desktop\boardcast.exe

Enter the number of nodes:5
A      B      C      D      E
A2 4 6 8 1
B1 5 3 7 9
C3 7 8 9 4
D1 2 3 4 5
E8 7 6 4 2
Minimum Cost is 6
Minimum Spanning tree is
AE1
BA1
CA3
DA1

-----
Process exited after 69.38 seconds with return value 0
Press any key to continue . . .
```