COMPUTER NETWORKS LAB (CSL5403)

Name: Lakhan Kumawat

Roll: 1906055

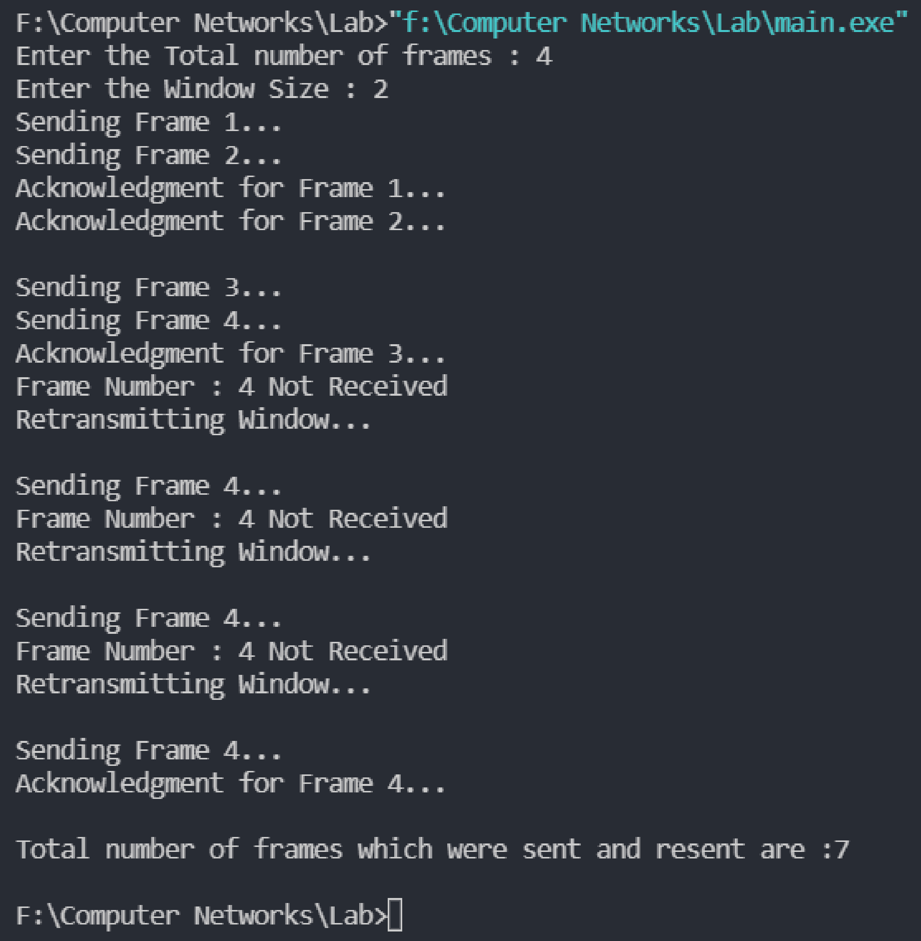
Program: B.Tech CSE
(5th Sem JUL-DEC 2021)

Assignment - 4

Name : Lakhan Kumawat Roll No : 1906055 Course : CSL5403

Go Back N ARQ:

```
#include <iostream>
#include <cstdlib>
#include <ctime>
using namespace std;
void transmission(int &i, int &N, int &tf, int &tt)
{
    while (i <= tf)
    {
        int z = 0;
        for (int k = i; k < i + N && k <= tf; k++)
        {
            cout << "Sending Frame " << k << "..." << endl;
            tt++;
        }
        for (int k = i; k < i + N && k <= tf; k++)
        {
            int f = rand() % 2;
            if (!f)
            {
                cout << "Acknowledgment for Frame " << k << "..
." << endl;
                z++;
            }
            else
            {
                cout << "Frame Number : " << k << " Not Receive
d" << endl;
                cout << "Retransmitting Window..." << endl;
                break;
            }
        }
        cout << "\n";
        i = i + z;
    }
}
int main()
{
    int tf, N, tt = 0;
    srand(time(NULL));
    cout << "Enter the Total number of frames : ";
    cin >> tf;
    cout << "Enter the Window Size : ";
    cin >> N;
    int i = 1;
    transmission(i, N, tf, tt);
    cout << "Total number of frames which were sent and resent
are :"<<tt<<endl;
        return 0;
}
```



```
F:\Computer Networks\Lab>"f:\Computer Networks\Lab\main.exe"
Enter the Total number of frames : 4
Enter the Window Size : 2
Sending Frame 1...
Sending Frame 2...
Acknowledgment for Frame 1...
Acknowledgment for Frame 2...

Sending Frame 3...
Sending Frame 4...
Acknowledgment for Frame 3...
Frame Number : 4 Not Received
Retransmitting Window...

Sending Frame 4...
Frame Number : 4 Not Received
Retransmitting Window...

Sending Frame 4...
Frame Number : 4 Not Received
Retransmitting Window...

Sending Frame 4...
Acknowledgment for Frame 4...

Total number of frames which were sent and resent are :7

F:\Computer Networks\Lab>
```

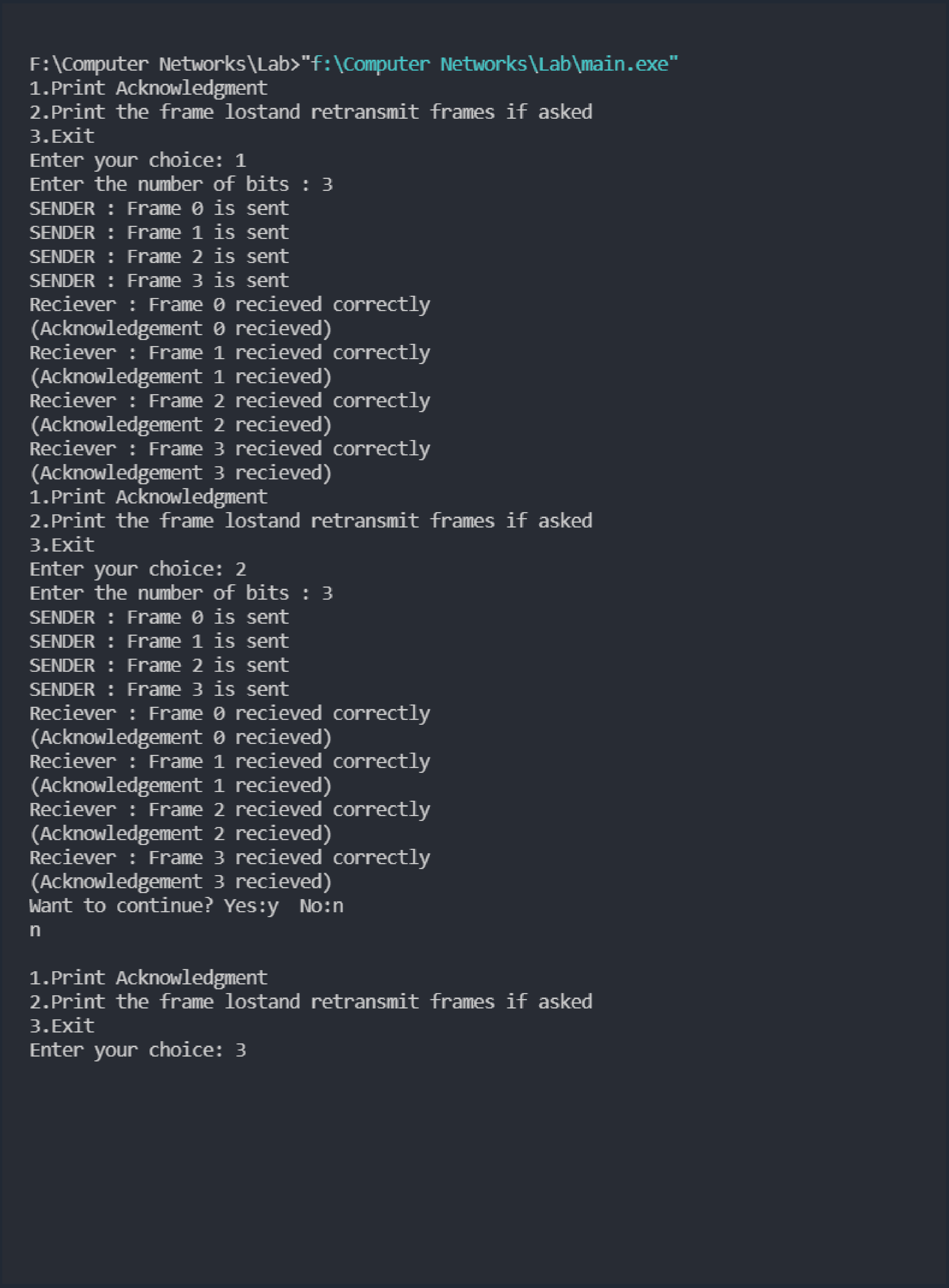
Selective ARQ:

```
#include <iostream>
#include <conio.h>
#include <stdlib.h>
#include <time.h>
#include <math.h>
using namespace std;
#define TOT_FRAMES 50
#define FRAMES_SEND 10
class sel_repeat
{
private:
    int fr_send_at_instance;
    int arr[TOT_FRAMES];
    int send[FRAMES_SEND];
    int rcvd[FRAMES_SEND];
    char rcvd_ack[FRAMES_SEND];
    int sw;
    int rw;

public:
    void input();
    void sender(int);
    void receiver(int);
    void case1();
    void case2();
};
void sel_repeat::input()
{
    int n;
    int m;
    int i;
    cout << "Enter the number of bits : ";
    cin >> n;
    m = pow(2, n);
    int t = 0;
    fr_send_at_instance = (m / 2);
    for (i = 0; i < TOT_FRAMES; i++)
    {
        arr[i] = t;
        t = (t + 1) % m;
    }
    for (i = 0; i < fr_send_at_instance; i++)
    {
        send[i] = arr[i];
        rcvd[i] = arr[i];
        rcvd_ack[i] = 'n';
    }
    rw = sw = fr_send_at_instance;
    sender(m);
}
void sel_repeat::sender(int m)
{
    for (int i = 0; i < fr_send_at_instance; i++)
    {
        if (rcvd_ack[i] == 'n')
            cout << "SENDER : Frame " << send[i] << " is sent\n
";
    }
    receiver(m);
}
void sel_repeat::receiver(int m)
{
    time_t t;
    int f;
    int j;
    int f1;
    int a1;
    char ch;
    srand((unsigned)time(&t));
    for (int i = 0; i < fr_send_at_instance; i++)
    {
        if (rcvd_ack[i] == 'n')
        {
            f = rand() % 10;
            if (f != 5)
            {
                for (int j = 0; j < fr_send_at_instance; j++)
                    if (rcvd[j] == send[i])
                    {
                        cout << "Reciever : Frame "
                             << rcvd[j] << " recieved correctly
\n";
                        rcvd[j] = arr[rw];
                        rw = (rw + 1) % m;
                        break;
                    }
                int j;
                if (j == fr_send_at_instance)
                    cout << "Reciever : Duplicate frame "
                         << send[i] << " discarded\n";
                a1 = rand() % 5;
                if (a1 == 3)
                {
                    cout << "(Acknowledgement " << send[i] << "
 lost)\n";
                    cout << "(Sender timeouts --
> Resend the frame)\n";
                    rcvd_ack[i] = 'n';
                }
                else
                {
                    cout << "(Acknowledgement " << send[i] << "
 recieved)\n";
                    rcvd_ack[i] = 'p';
                }
            }
            else
            {
                int ld = rand() % 2;
                if (ld == 0)
                {
                    cout << "RECEIVER : Frame " << send[i]
                         << " is damaged\n";
                    cout << "RECEIVER : Negative Acknowledgemen
t "
                         << send[i] << " sent\n";
                }
                else
                {
                    cout << "RECEIVER : Frame " << send[i] << "
 is lost\n";
                    cout << "(Sender timeouts --
> Resend the frame)\n";
                }
                rcvd_ack[i] = 'n';
            }
        }
    }
    for (int j = 0; j < fr_send_at_instance; j++)
    {
        if (rcvd_ack[j] == 'n')
            break;
    }
    int i = 0;
    for (int k = j; k < fr_send_at_instance; k++)
    {
        send[i] = send[k];
        if (rcvd_ack[k] == 'n')
            rcvd_ack[i] = 'n';
        else
            rcvd_ack[i] = 'p';
        i++;
    }
    if (i != fr_send_at_instance)
    {
        for (int k = i; k < fr_send_at_instance; k++)
        {
            send[k] = arr[sw];
            sw = (sw + 1) % m;
            rcvd_ack[k] = 'n';
        }
    }
    cout << "Want to continue? Yes:y\t No:n\n";
    cin >> ch;
    cout << "\n";
    if (ch == 'y')
        sender(m);
}
void sel_repeat::case1()
{
    int n, m, i;
    cout << "Enter the number of bits : ";
    cin >> n;
    m = pow(2, n);
    int t = 0;
    fr_send_at_instance = (m / 2);
    for (i = 0; i < TOT_FRAMES; i++)
    {
        arr[i] = t;
        t = (t + 1) % m;
    }
    for (i = 0; i < fr_send_at_instance; i++)
    {
        send[i] = arr[i];
        rcvd[i] = arr[i];
        rcvd_ack[i] = 'n';
    }
    rw = sw = fr_send_at_instance;
    for (int i = 0; i < fr_send_at_instance; i++)
    {
        if (rcvd_ack[i] == 'n')
            cout << "SENDER : Frame " << send[i] << " is sent\n
";
    }
    time_t ti;
    int f, j, f1, a1;
    char ch;
    srand((unsigned)time(&ti));
    for (int i = 0; i < fr_send_at_instance; i++)
    {
        if (rcvd_ack[i] == 'n')
        {
            f = rand() % 10;
            for (int j = 0; j < fr_send_at_instance; j++)
                if (rcvd[j] == send[i])
                {
                    cout << "Reciever : Frame " << rcvd[j]
                         << " recieved correctly\n";
                    rcvd[j] = arr[rw];
                    rw = (rw + 1) % m;
                    break;
                }
            int j;
            if (j == fr_send_at_instance)
                cout << "Reciever : Duplicate frame "
                      << send[i] << " discarded\n";
            cout << "(Acknowledgement " << send[i] << " recieve
d)\n";
            rcvd_ack[i] = 'p';
        }
    }
}
void sel_repeat::case2()
```

```
{
    int n, m, i;
    cout << "Enter the number of bits : ";
    cin >> n;
    m = pow(2, n);
    int t = 0;
    fr_send_at_instance = (m / 2);
    for (i = 0; i < TOT_FRAMES; i++)
    {
        arr[i] = t;
        t = (t + 1) % m;
    }
    for (i = 0; i < fr_send_at_instance; i++)
    {
        send[i] = arr[i];
        rcvd[i] = arr[i];
        rcvd_ack[i] = 'n';
    }
    rw = sw = fr_send_at_instance;
    for (int i = 0; i < fr_send_at_instance; i++)
    {
        if (rcvd_ack[i] == 'n')
            cout << "SENDER : Frame " << send[i] << " is sent\n
";
    }
    time_t ti;
    int f, j, f1, a1;
    char ch;
    srand((unsigned)time(&ti));
    for (int i = 0; i < fr_send_at_instance; i++)
    {
        if (rcvd_ack[i] == 'n')
        {
            f = rand() % 10;
            if (f != 5)
            {
                for (int j = 0; j < fr_send_at_instance; j++)
                    if (rcvd[j] == send[i])
                    {
                        cout << "Reciever : Frame "
                             << rcvd[j] << " recieved correctly
\n";
                        rcvd[j] = arr[rw];
                        rw = (rw + 1) % m;
                        break;
                    }
                int j;
                if (j == fr_send_at_instance)
                    cout << "Reciever : Duplicate frame "
                         << send[i] << " discarded\n";
                a1 = rand() % 5;
                if (a1 == 3)
                {
                    cout << "(Acknowledgement "
                         << send[i] << " lost)\n";
                    cout << "(Sender timeouts --
> Resend the frame)\n";
                    rcvd_ack[i] = 'n';
                }
                else
                {
                    cout << "(Acknowledgement "
                         << send[i] << " recieved)\n";
                    rcvd_ack[i] = 'p';
                }
            }
            else
            {
                int ld = rand() % 2;
                if (ld == 0)
                {
                    cout << "RECEIVER : Frame "
```

```
                         << send[i] << " is damaged\n";
                    cout << "RECEIVER : Negative Acknowledgemen
t "
                         << send[i] << " sent\n";
                }
                else
                {
                    cout << "RECEIVER : Frame "
                         << send[i] << " is lost\n";
                    cout << "(Sender timeouts --
> Resend the frame)\n";
                }
                rcvd_ack[i] = 'n';
            }
        }
    }
    for (int j = 0; j < fr_send_at_instance; j++)
    {
        if (rcvd_ack[j] == 'n')
            break;
    }
    i = 0;
    for (int k = j; k < fr_send_at_instance; k++)
    {
        send[i] = send[k];
        if (rcvd_ack[k] == 'n')
            rcvd_ack[i] = 'n';
        else
            rcvd_ack[i] = 'p';
        i++;
    }
    if (i != fr_send_at_instance)
    {
        for (int k = i; k < fr_send_at_instance; k++)
        {
            send[k] = arr[sw];
            sw = (sw + 1) % m;
            rcvd_ack[k] = 'n';
        }
    }
    cout << "Want to continue? Yes:y\t No:n\n";
    cin >> ch;
    cout << "\n";
    if (ch == 'y')
        sender(m);
}
int main()
{
    sel_repeat sr;
    int a = 1;
    while (a <= 2)
    {
        cout << "1.Print Acknowledgment" << endl;
        cout << "2.Print the frame lost"
             << "and retransmit frames if asked"
             << endl;
        cout << "3.Exit" << endl;
        cout << "Enter your choice: ";
        cin >> a;
        switch (a)
        {
        case 1:
            sr.case1();
            break;
        case 2:
            sr.case2();
            break;
        default:
            break;
        }
    }
}
```



```
F:\Computer Networks\Lab>"f:\Computer Networks\Lab\main.exe"
1.Print Acknowledgment
2.Print the frame lostand retransmit frames if asked
3.Exit
Enter your choice: 1
Enter the number of bits : 3
SENDER : Frame 0 is sent
SENDER : Frame 1 is sent
SENDER : Frame 2 is sent
SENDER : Frame 3 is sent
Reciever : Frame 0 recieved correctly
(Acknowledgement 0 recieved)
Reciever : Frame 1 recieved correctly
(Acknowledgement 1 recieved)
Reciever : Frame 2 recieved correctly
(Acknowledgement 2 recieved)
Reciever : Frame 3 recieved correctly
(Acknowledgement 3 recieved)
1.Print Acknowledgment
2.Print the frame lostand retransmit frames if asked
3.Exit
Enter your choice: 2
Enter the number of bits : 3
SENDER : Frame 0 is sent
SENDER : Frame 1 is sent
SENDER : Frame 2 is sent
SENDER : Frame 3 is sent
Reciever : Frame 0 recieved correctly
(Acknowledgement 0 recieved)
Reciever : Frame 1 recieved correctly
(Acknowledgement 1 recieved)
Reciever : Frame 2 recieved correctly
(Acknowledgement 2 recieved)
Reciever : Frame 3 recieved correctly
(Acknowledgement 3 recieved)
Want to continue? Yes:y  No:n
n

1.Print Acknowledgment
2.Print the frame lostand retransmit frames if asked
3.Exit
Enter your choice: 3
```

Name : Lakhan Kumawat Roll No : 1906055 Course : CSL5403

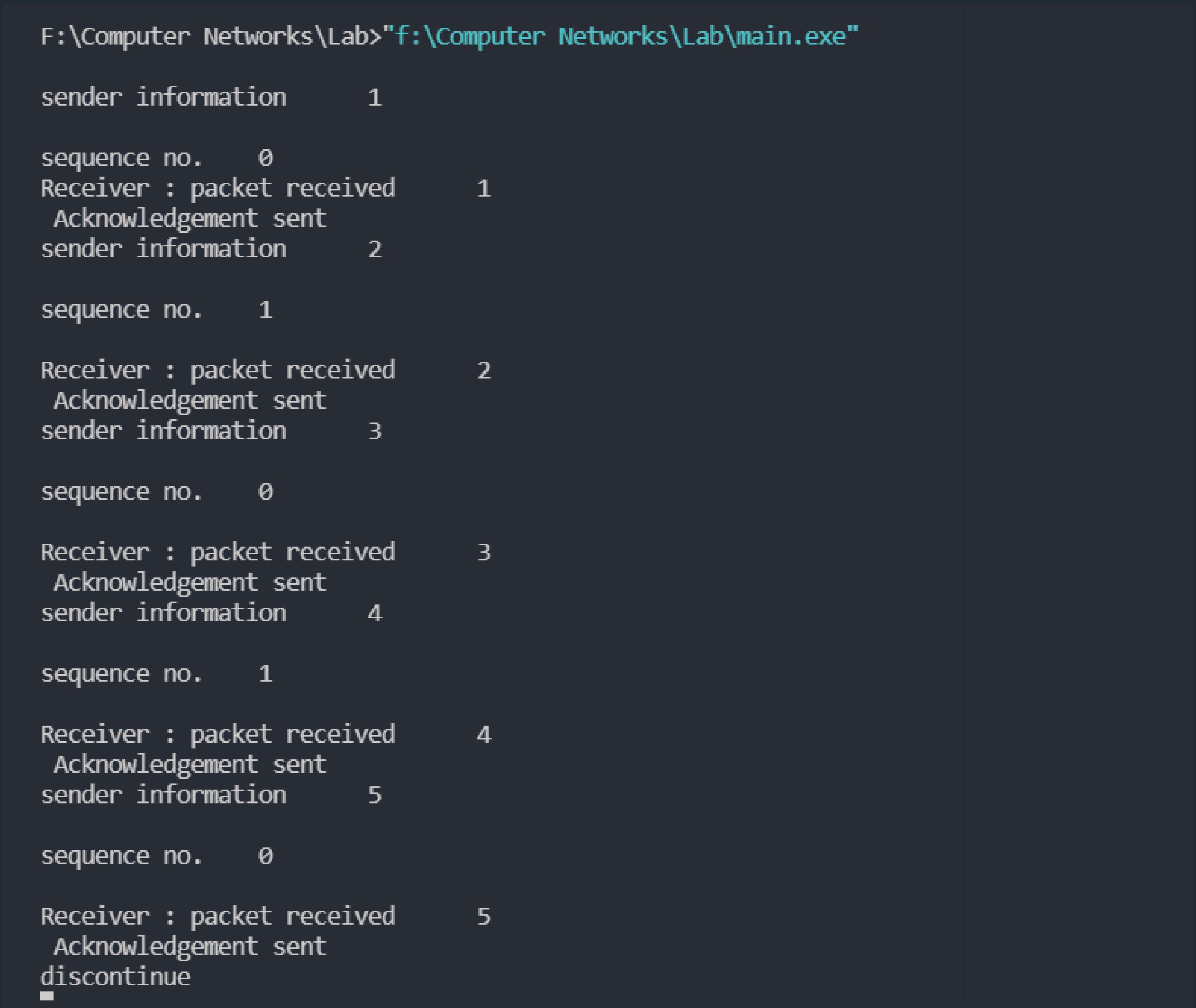
Stop & Wait ARQ:

```
#include <iostream>
#include <stdio.h>
#include <stdlib.h>
#include <conio.h>
#include <dos.h>
using namespace std;
#define time 5
#define max_seq 1
#define tot_pack 5
int randn(int n)
{
    return rand() % n + 1;
}
typedef struct
{
    int data;
} packet;
typedef struct
{
    int kind;
    int seq;
    int ack;
    packet info;
} frame;
typedef enum
{
    frame_arrival,
    error,
    time_out
} event_type;
frame data1;
//creating prototype
void from_network_layer(packet *);
```

```c
void to_physical_layer(frame *);
void to_network_layer(packet *);
void from_physical_layer(frame *);
void sender();
void receiver();
void wait_for_event_sender(event_type *);
void wait_for_event_receiver(event_type *);
//end
#define inc(k)       \
    if (k < max_seq) \
        k++;         \
    else             \
        k = 0;
int i = 1;
char turn;
int disc = 0;
int main()
{
    while (!disc)
    {
        sender();
        // delay(400);
        receiver();
    }
    getchar();
}
void sender()
{
    static int frame_to_send = 0;
    static frame s;
    packet buffer;
    event_type event;
    static int flag = 0; //first place
    if (flag == 0)
    {
        from_network_layer(&buffer);
        s.info = buffer;
        s.seq = frame_to_send;
        cout << "\nsender information \t" << s.info.data << "\n
";
        cout << "\nsequence no. \t" << s.seq;
        turn = 'r';
        to_physical_layer(&s);
        flag = 1;
    }
    wait_for_event_sender(&event);
    if (turn == 's')
    {
        if (event == frame_arrival)
        {
            from_network_layer(&buffer);
            inc(frame_to_send);
            s.info = buffer;
            s.seq = frame_to_send;
            cout << "\nsender information \t" << s.info.data <<
 "\n";
            cout << "\nsequence no. \t" << s.seq << "\n";
            getch();
            turn = 'r';
            to_physical_layer(&s);
        }
    }
} //end of sender function
void from_network_layer(packet *buffer)
{
    (*buffer).data = i;
    i++;
} //end of from network layer function
void to_physical_layer(frame *s)
{
    data1 = *s;
} //end of to physical layer function
```

```
void wait_for_event_sender(event_type *e)
{
    static int timer = 0;
    if (turn == 's')
    {
        timer++;
        //timer=0;
        return;
    }
    else //event is frame arrival
    {
        timer = 0;
        *e = frame_arrival;
    }
} //end of wait for event function
void receiver()
{
    static int frame_expected = 0;
    frame s, r;
    event_type event;
    wait_for_event_receiver(&event);
    if (turn == 'r')
    {
        if (event == frame_arrival)
        {
            from_physical_layer(&r);
            if (r.seq == frame_expected)
            {
                to_network_layer(&r.info);
                inc(frame_expected);
            }
            else
                cout << "\nReceiver :Acknowledgement resent \n"
;
            getch();
            turn = 's';
            to_physical_layer(&s);
        }
    }
} //end of receiver function
void wait_for_event_receiver(event_type *e)
{
    if (turn == 'r')
    {
        *e = frame_arrival;
    }
}
void from_physical_layer(frame *buffer)
{
    *buffer = data1;
}
void to_network_layer(packet *buffer)
{
    cout << "\nReceiver : packet received \t" << i - 1;
    cout << "\n Acknowledgement sent \t";
    getch();
    if (i > tot_pack)
    {
        disc = 1;
        cout << "\ndiscontinue\n";
    }
}
```

Name : Lakhan Kumawat Roll No : 1906055 Course : CSL5403



```
F:\Computer Networks\Lab>"f:\Computer Networks\Lab\main.exe"

sender information      1

sequence no.    0
Receiver : packet received      1
 Acknowledgement sent
sender information      2

sequence no.    1

Receiver : packet received      2
 Acknowledgement sent
sender information      3

sequence no.    0

Receiver : packet received      3
 Acknowledgement sent
sender information      4

sequence no.    1

Receiver : packet received      4
 Acknowledgement sent
sender information      5

sequence no.    0

Receiver : packet received      5
 Acknowledgement sent
discontinue
```

End Of Assignment