VIT®
Vellore Institute of Technology
(Deemed to be University under section 3 of UGC Act, 1956)

LAB DA – 3

Name : Gabani Vandit Sureshbhai

Reg.No : 20BCI0090

Subject : CSE 1004 – Network and Communication

Faculty : SALEEM DURAI M.A

Topic : Flow Control Mechanism

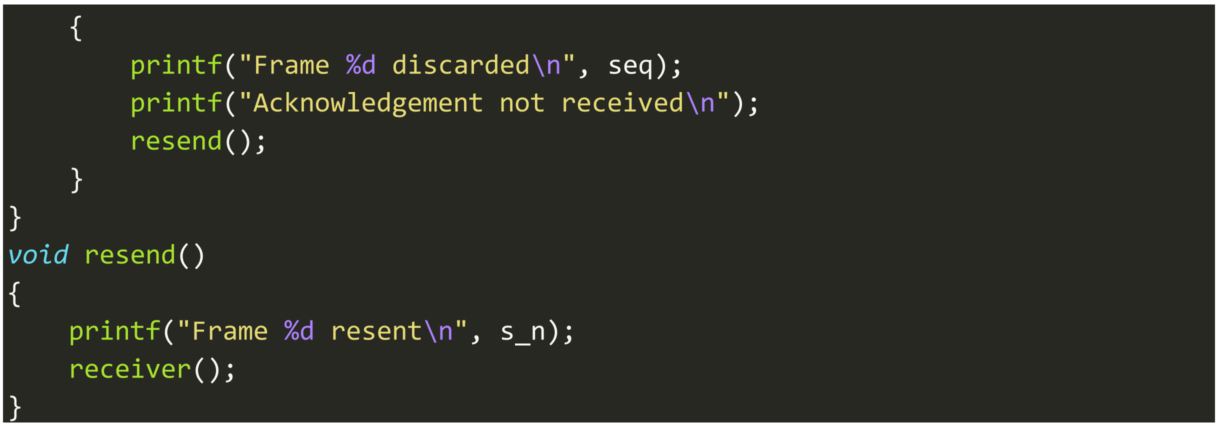
1. Stop and Wait ARQ
2. GO-Back-N ARQ
3. Selective Repeat ARQ
4. Piggbacking in Go-Back-N ARQ

Vandit Gabani
20BCI0090

1.Stop and Wait ARQ:

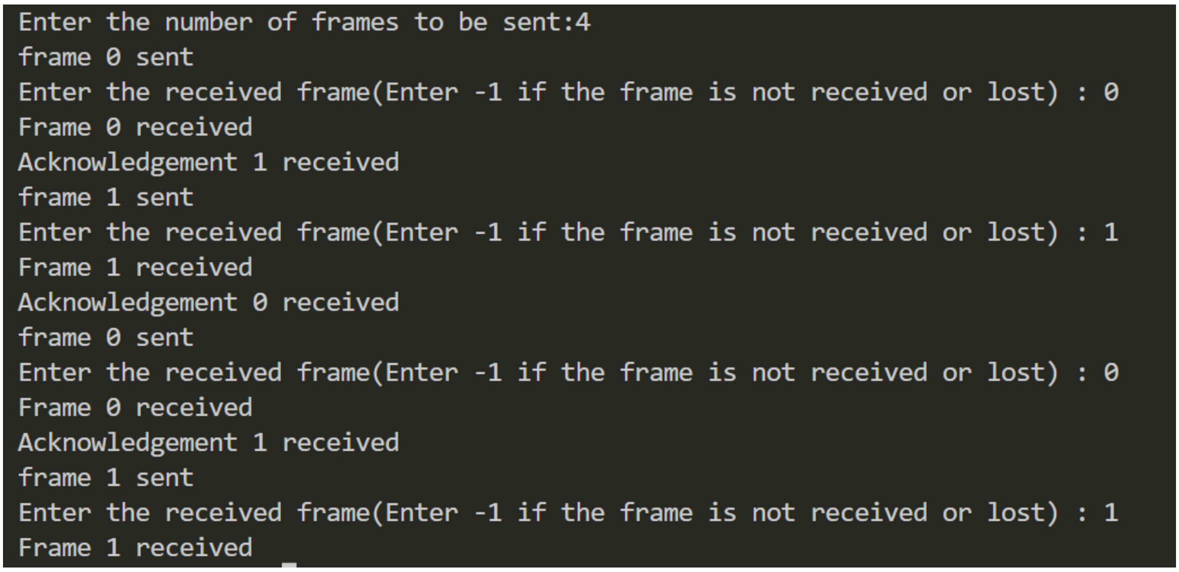
Code:

```
#include <stdio.h>
int n, seq;
int s_n = 0, r_n = 0;
void sender();
void receiver();
void resend();
int main()
{
    printf("Enter the number of frames to be sent:");
    scanf("%d", &n);
    sender();
    return 0;
}
void sender()
{
    while (n != 0)
    {
        printf("frame %d sent\n", s_n);
        receiver();
    }
}
void receiver()
{
    printf("Enter the received frame(Enter -1 if the frame is not received or
lost) : ");
                   scanf("%d", &seq);
    if (seq == r_n)
    {
        printf("Frame %d received\n", seq);
        r_n = (r_n + 1) % 2;
        s_n = (s_n + 1) % 2;
        n--;
        if (n != 0)
        {
            printf("Acknowledgement %d received\n", r_n);
            sender();
        }
    }
    else if (seq == -1)
    {
        printf("Frame %d lost\n", r_n);
        printf("Acknowledgement not received\n");
        resend();
    }
    else
```

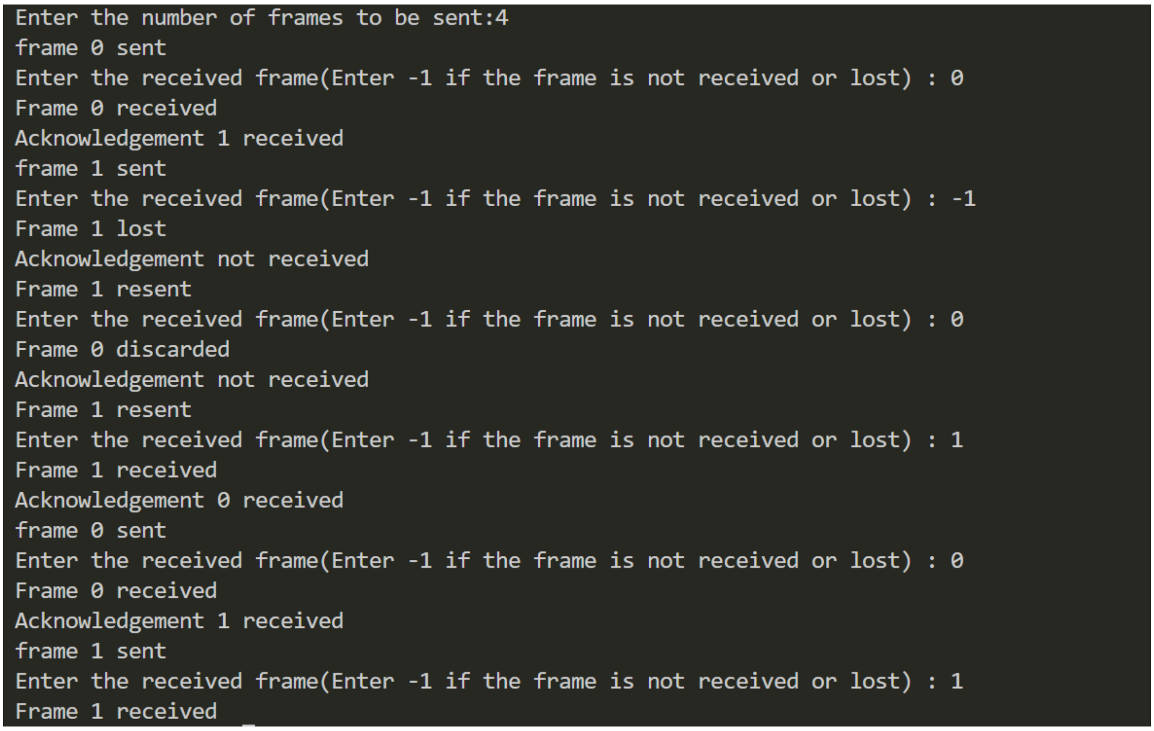


```
    {
        printf("Frame %d discarded\n", seq);
        printf("Acknowledgement not received\n");
        resend();
    }
}
void resend()
{
    printf("Frame %d resent\n", s_n);
    receiver();
}
```

Input – Output



```
Enter the number of frames to be sent:4
frame 0 sent
Enter the received frame(Enter -1 if the frame is not received or lost) : 0
Frame 0 received
Acknowledgement 1 received
frame 1 sent
Enter the received frame(Enter -1 if the frame is not received or lost) : 1
Frame 1 received
Acknowledgement 0 received
frame 0 sent
Enter the received frame(Enter -1 if the frame is not received or lost) : 0
Frame 0 received
Acknowledgement 1 received
frame 1 sent
Enter the received frame(Enter -1 if the frame is not received or lost) : 1
Frame 1 received
```



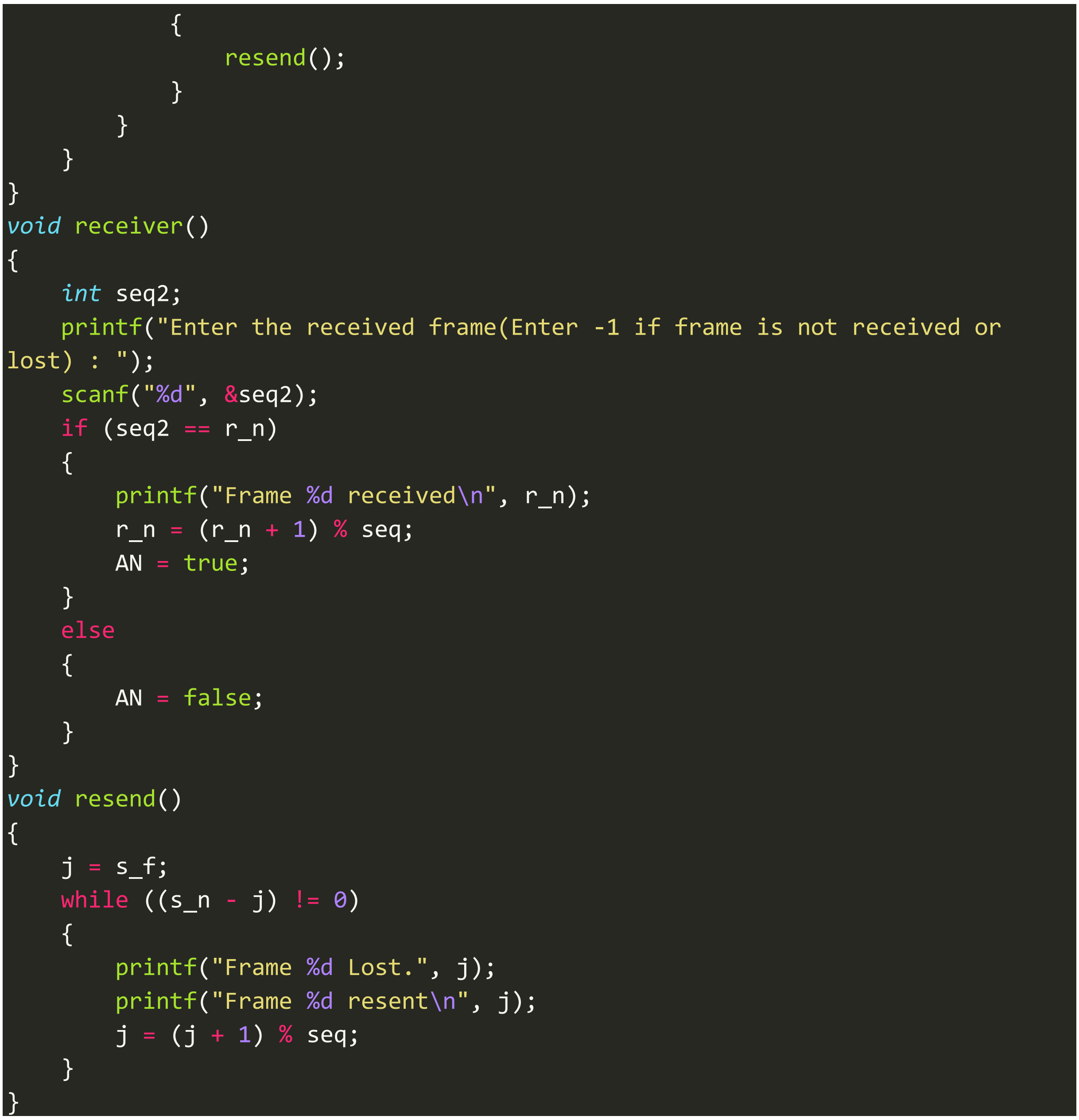
```
Enter the number of frames to be sent:4
frame 0 sent
Enter the received frame(Enter -1 if the frame is not received or lost) : 0
Frame 0 received
Acknowledgement 1 received
frame 1 sent
Enter the received frame(Enter -1 if the frame is not received or lost) : -1
Frame 1 lost
Acknowledgement not received
Frame 1 resent
Enter the received frame(Enter -1 if the frame is not received or lost) : 0
Frame 0 discarded
Acknowledgement not received
Frame 1 resent
Enter the received frame(Enter -1 if the frame is not received or lost) : 1
Frame 1 received
Acknowledgement 0 received
frame 0 sent
Enter the received frame(Enter -1 if the frame is not received or lost) : 0
Frame 0 received
Acknowledgement 1 received
frame 1 sent
Enter the received frame(Enter -1 if the frame is not received or lost) : 1
Frame 1 received
```

Vandit Gabani
20BCI0090

2.Go-Back-N ARQ:

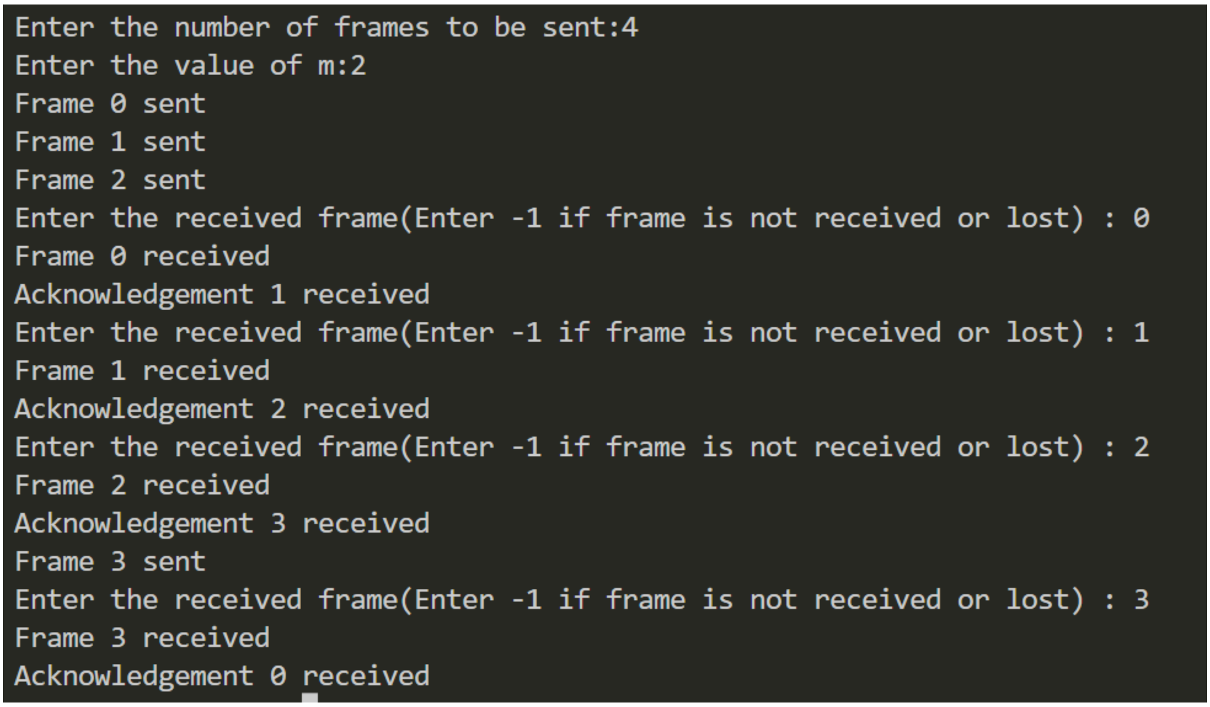
Code:

```
#include <stdio.h>
#include <stdbool.h>
#include <math.h>
int j, n, m, seq, w_size, s_n = 0, s_f = 0, r_n = 0;
bool AN = false;
void sender();
void receiver();
void resend();
int main()
{
    printf("Enter the number of frames to be sent:");
    scanf("%d", &n);
    printf("Enter the value of m:");
    scanf("%d", &m);
    seq = (int)pow(2, m);
    w_size = (int)pow(2, m) - 1;
    sender();
}
void sender()
{
    while (n != 0)
    {
        int i = 0;
        while (i < w_size)
        {
            printf("Frame %d sent\n", s_n);
            s_n = (s_n + 1) % seq;
            n--;
            i++;
            if (n == 0)
            {
                break;
            }
        }
        while ((s_n - s_f) != 0)
        {
            receiver();
            if (AN)
            {
                printf("Acknowledgement %d received\n", r_n);
                AN = false;
                s_f++;
                s_f = s_f % seq;
            }
            else
```

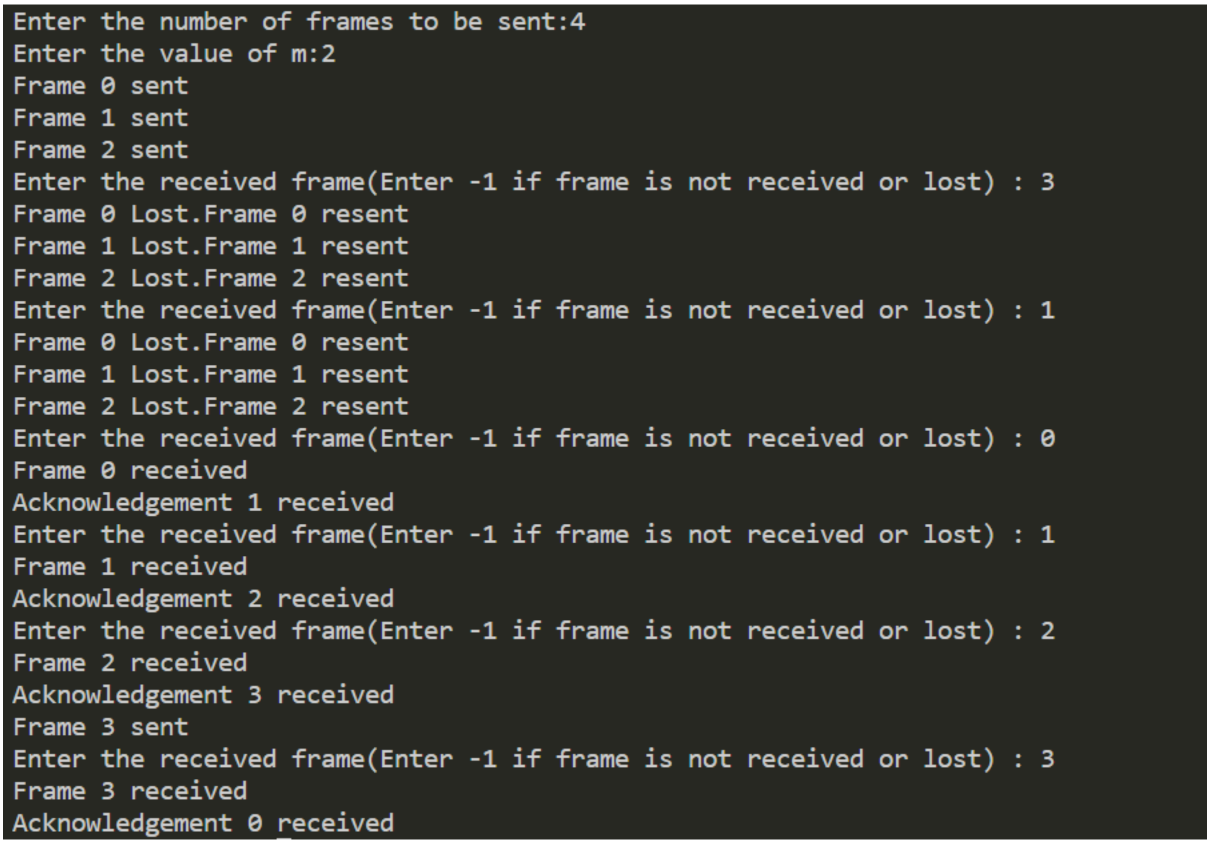


```
            {
                resend();
            }
        }
    }
}
void receiver()
{
    int seq2;
    printf("Enter the received frame(Enter -1 if frame is not received or
lost) : ");
    scanf("%d", &seq2);
    if (seq2 == r_n)
    {
        printf("Frame %d received\n", r_n);
        r_n = (r_n + 1) % seq;
        AN = true;
    }
    else
    {
        AN = false;
    }
}
void resend()
{
    j = s_f;
    while ((s_n - j) != 0)
    {
        printf("Frame %d Lost.", j);
        printf("Frame %d resent\n", j);
        j = (j + 1) % seq;
    }
}
```

Input – Output:



```
Enter the number of frames to be sent:4
Enter the value of m:2
Frame 0 sent
Frame 1 sent
Frame 2 sent
Enter the received frame(Enter -1 if frame is not received or lost) : 0
Frame 0 received
Acknowledgement 1 received
Enter the received frame(Enter -1 if frame is not received or lost) : 1
Frame 1 received
Acknowledgement 2 received
Enter the received frame(Enter -1 if frame is not received or lost) : 2
Frame 2 received
Acknowledgement 3 received
Frame 3 sent
Enter the received frame(Enter -1 if frame is not received or lost) : 3
Frame 3 received
Acknowledgement 0 received
```

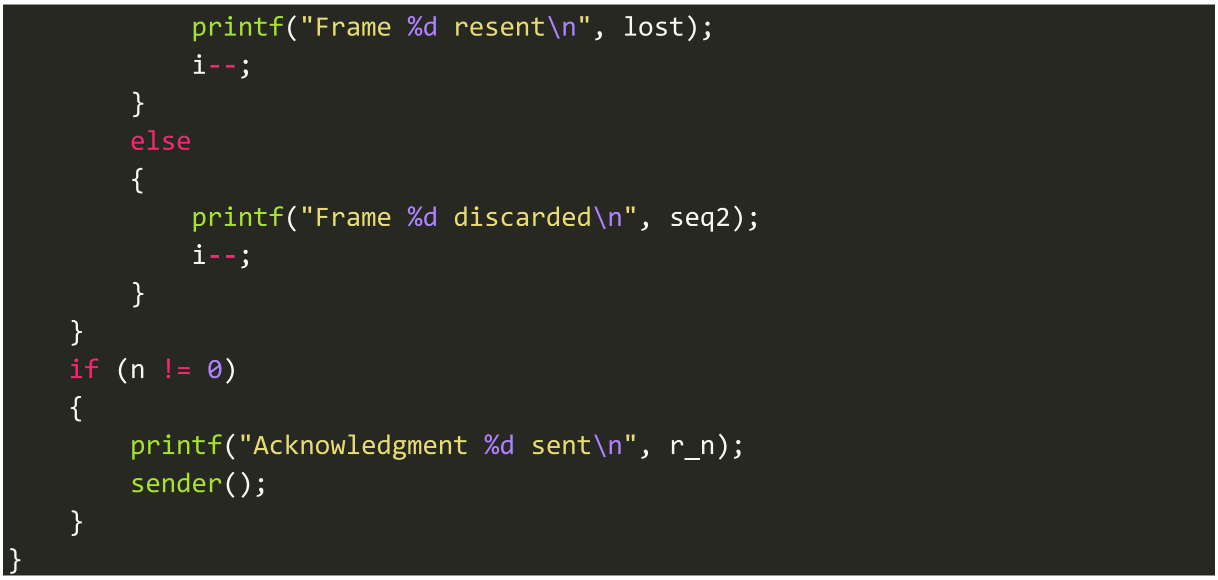


```
Enter the number of frames to be sent:4
Enter the value of m:2
Frame 0 sent
Frame 1 sent
Frame 2 sent
Enter the received frame(Enter -1 if frame is not received or lost) : 3
Frame 0 Lost.Frame 0 resent
Frame 1 Lost.Frame 1 resent
Frame 2 Lost.Frame 2 resent
Enter the received frame(Enter -1 if frame is not received or lost) : 1
Frame 0 Lost.Frame 0 resent
Frame 1 Lost.Frame 1 resent
Frame 2 Lost.Frame 2 resent
Enter the received frame(Enter -1 if frame is not received or lost) : 0
Frame 0 received
Acknowledgement 1 received
Enter the received frame(Enter -1 if frame is not received or lost) : 1
Frame 1 received
Acknowledgement 2 received
Enter the received frame(Enter -1 if frame is not received or lost) : 2
Frame 2 received
Acknowledgement 3 received
Frame 3 sent
Enter the received frame(Enter -1 if frame is not received or lost) : 3
Frame 3 received
Acknowledgement 0 received
```

3.Selective Repeat ARQ:

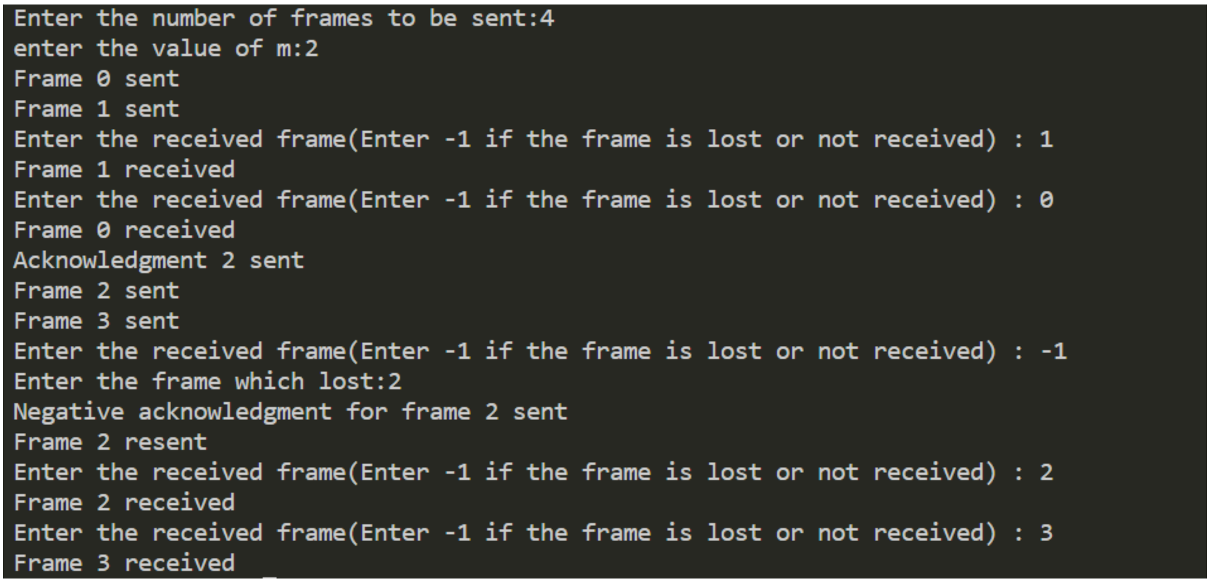
Code:

```c
#include <stdio.h>
#include <math.h>
int n, m, seq, w_size, ack, nak, s_n = 0, r_n = 0, seq2, lost, j;
void sender();
void receiver();
int main()
{
    printf("Enter the number of frames to be sent:");
    scanf("%d", &n);
    printf("enter the value of m:");
    scanf("%d", &m);
    seq = (int)pow(2, m);
    w_size = (int)pow(2, m - 1);
    sender();
    return 0;
}
void sender()
{
    for (j = 0; j < w_size && n != 0; j++)
    {
        printf("Frame %d sent\n", s_n);
        s_n = (s_n + 1) % seq;
        n--;
    }
    receiver();
}
void receiver()
{
    int i, k;
    k = r_n;
    for (i = 0; i < j; i++)
    {
        printf("Enter the received frame(Enter -1 if the frame is lost or not received) : ");
        scanf("%d", &seq2);
        if (seq2 >= k && seq2 < k + j)
        {
            printf("Frame %d received\n", seq2);
            r_n = (r_n + 1) % seq;
        }
        else if (seq2 == -1)
        {
            printf("Enter the frame which lost:");
            scanf("%d", &lost);
            printf("Negative acknowledgment for frame %d sent\n", lost);
```

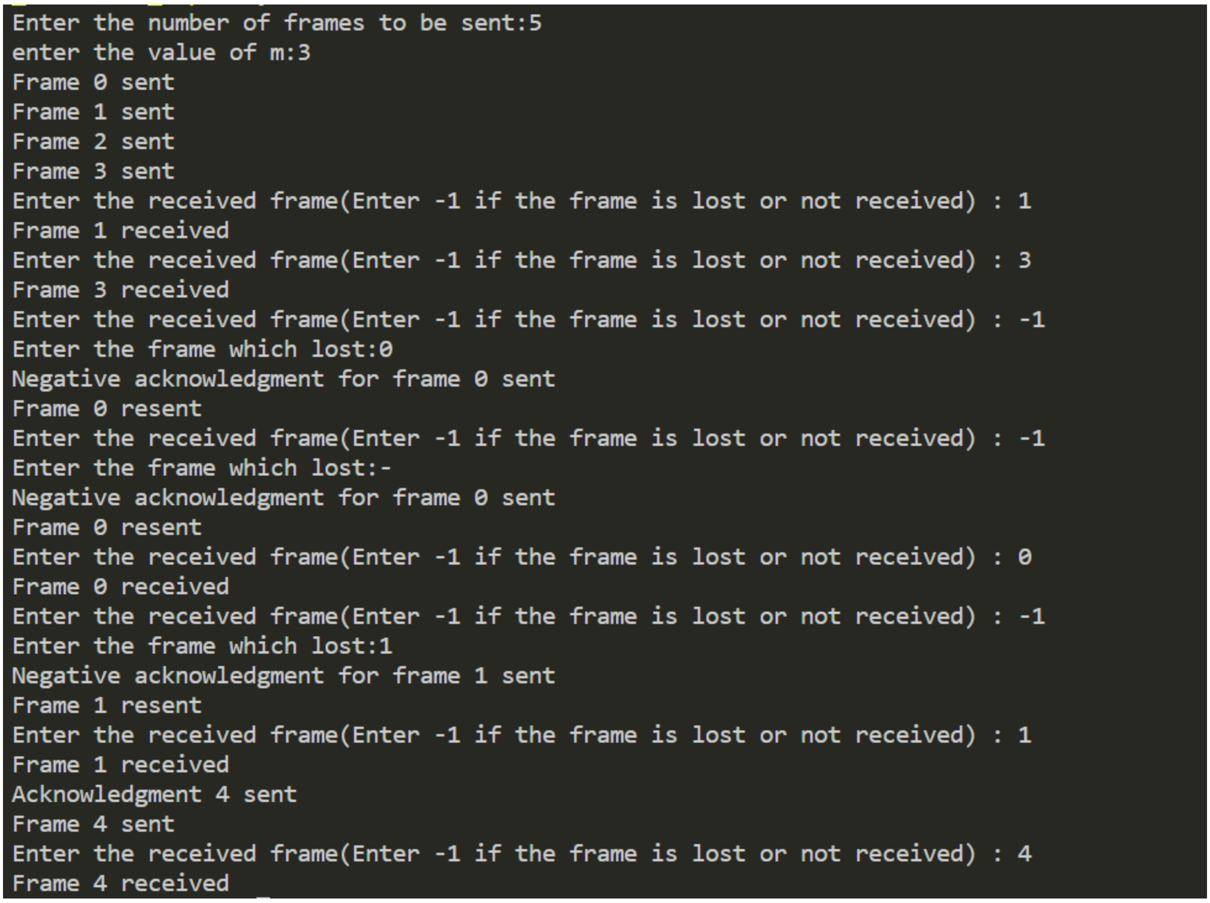


```
            printf("Frame %d resent\n", lost);
            i--;
        }
        else
        {
            printf("Frame %d discarded\n", seq2);
            i--;
        }
    }
    if (n != 0)
    {
        printf("Acknowledgment %d sent\n", r_n);
        sender();
    }
}
```

Input – Output:



```
Enter the number of frames to be sent:4
enter the value of m:2
Frame 0 sent
Frame 1 sent
Enter the received frame(Enter -1 if the frame is lost or not received) : 1
Frame 1 received
Enter the received frame(Enter -1 if the frame is lost or not received) : 0
Frame 0 received
Acknowledgment 2 sent
Frame 2 sent
Frame 3 sent
Enter the received frame(Enter -1 if the frame is lost or not received) : -1
Enter the frame which lost:2
Negative acknowledgment for frame 2 sent
Frame 2 resent
Enter the received frame(Enter -1 if the frame is lost or not received) : 2
Frame 2 received
Enter the received frame(Enter -1 if the frame is lost or not received) : 3
Frame 3 received
```

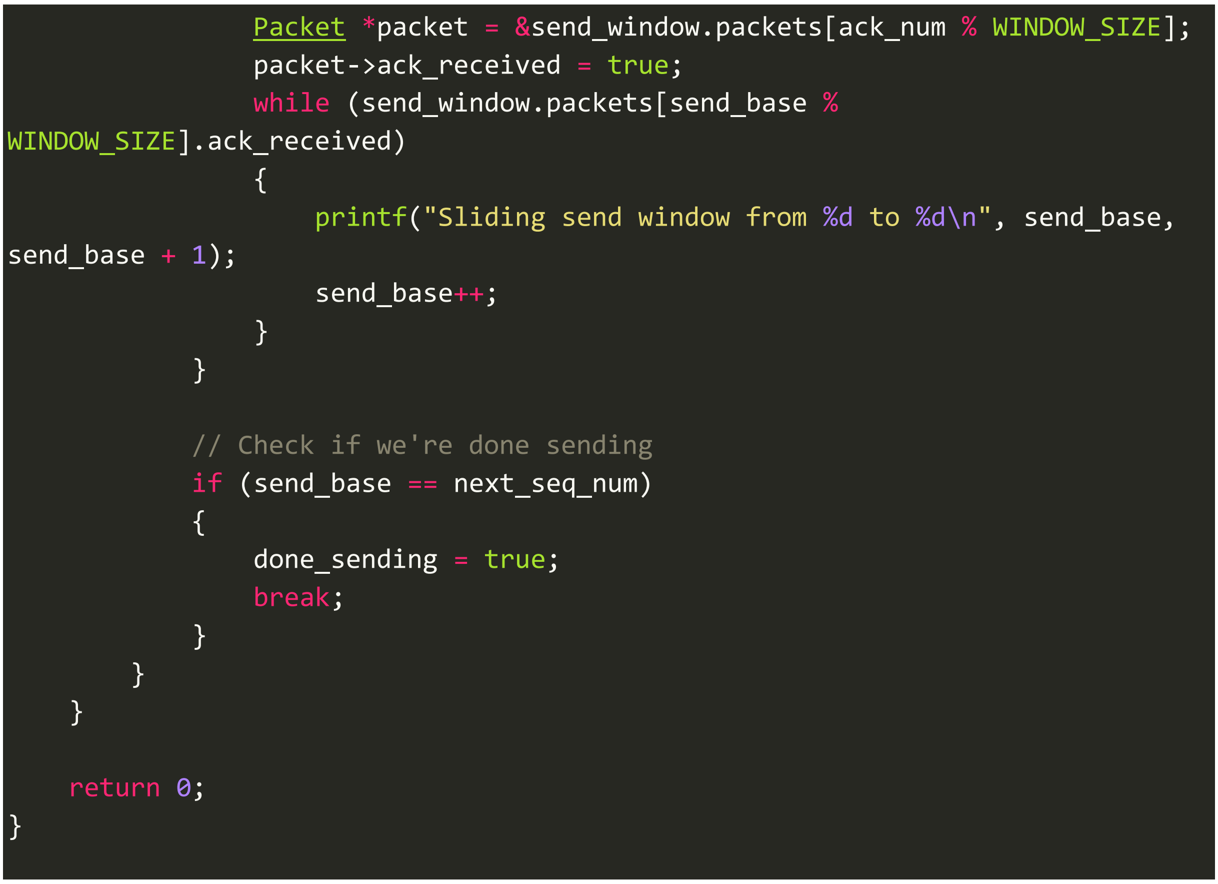


```
Enter the number of frames to be sent:5
enter the value of m:3
Frame 0 sent
Frame 1 sent
Frame 2 sent
Frame 3 sent
Enter the received frame(Enter -1 if the frame is lost or not received) : 1
Frame 1 received
Enter the received frame(Enter -1 if the frame is lost or not received) : 3
Frame 3 received
Enter the received frame(Enter -1 if the frame is lost or not received) : -1
Enter the frame which lost:0
Negative acknowledgment for frame 0 sent
Frame 0 resent
Enter the received frame(Enter -1 if the frame is lost or not received) : -1
Enter the frame which lost:-
Negative acknowledgment for frame 0 sent
Frame 0 resent
Enter the received frame(Enter -1 if the frame is lost or not received) : 0
Frame 0 received
Enter the received frame(Enter -1 if the frame is lost or not received) : -1
Enter the frame which lost:1
Negative acknowledgment for frame 1 sent
Frame 1 resent
Enter the received frame(Enter -1 if the frame is lost or not received) : 1
Frame 1 received
Acknowledgment 4 sent
Frame 4 sent
Enter the received frame(Enter -1 if the frame is lost or not received) : 4
Frame 4 received
```

4.Piggybacking in Go-Back-N ARQ:

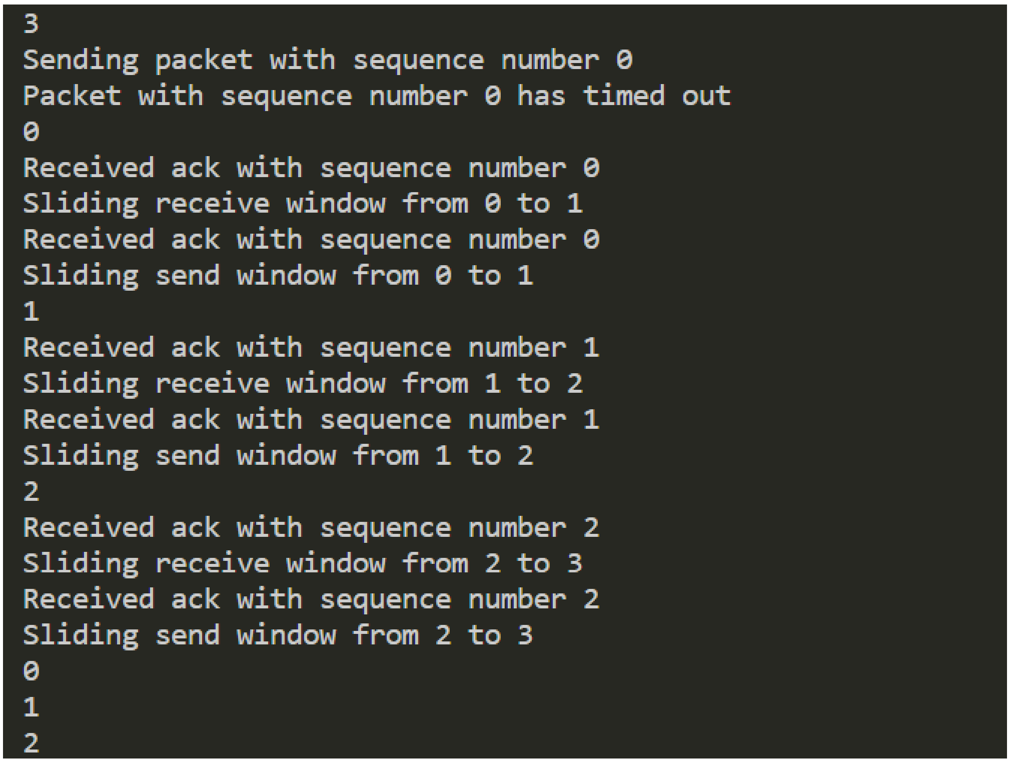
Code:

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

#define BUF_SIZE 1024
#define WINDOW_SIZE 4

typedef struct
{
    int seq_num;
    bool ack_received;
    char payload[BUF_SIZE];
} Packet;

typedef struct
{
    Packet packets[WINDOW_SIZE];
    int front;
    int rear;
    int count;
} Window;

int main()
{
    // Define variables
    int next_seq_num = 0;
    int send_base = 0;
    int recv_base = 0;
    char buffer[BUF_SIZE];
    bool done_sending = false;
    Window send_window = {0};
    Window recv_window = {0};

    // Send loop
    while (!done_sending)
    {
        // Check if we can send more packets
        if (next_seq_num < send_base + WINDOW_SIZE && fgets(buffer, BUF_SIZE,
stdin) != NULL)
        {
            // Create packet
            Packet packet;
            packet.seq_num = next_seq_num;
            packet.ack_received = false;
            strcpy(packet.payload, buffer);
            send_window.packets[next_seq_num % WINDOW_SIZE] = packet;

            // Send packet
            printf("Sending packet with sequence number %d\n",
packet.seq_num);
            next_seq_num++;
        }

        // Check if any packets have timed out
        for (int i = send_base; i < next_seq_num; i++)
        {
            Packet *packet = &send_window.packets[i % WINDOW_SIZE];
            if (!packet->ack_received)
            {
                printf("Packet with sequence number %d has timed out\n",
packet->seq_num);
                next_seq_num = i;
                break;
            }
        }

        // Receive loop
        char ack[BUF_SIZE];
        while (fgets(ack, BUF_SIZE, stdin) != NULL)
        {
            // Parse ack
            int ack_num = atoi(ack);

            // Update receive window
            if (ack_num >= recv_base && ack_num < recv_base + WINDOW_SIZE)
            {
                printf("Received ack with sequence number %d\n", ack_num);
                Packet *packet = &recv_window.packets[ack_num % WINDOW_SIZE];
                packet->ack_received = true;
                while (recv_window.packets[recv_base %
WINDOW_SIZE].ack_received)
                {
                    printf("Sliding receive window from %d to %d\n",
recv_base, recv_base + 1);
                    recv_base++;
                }
            }

            // Update send window
            if (ack_num >= send_base && ack_num < send_base + WINDOW_SIZE)
            {
                printf("Received ack with sequence number %d\n", ack_num);
```



```
                Packet *packet = &send_window.packets[ack_num % WINDOW_SIZE];
                packet->ack_received = true;
                while (send_window.packets[send_base %
WINDOW_SIZE].ack_received)
                {
                    printf("Sliding send window from %d to %d\n", send_base,
send_base + 1);
                    send_base++;
                }
            }

            // Check if we're done sending
            if (send_base == next_seq_num)
            {
                done_sending = true;
                break;
            }
        }
    }

    return 0;
}
```

Input – Output:



```
3
Sending packet with sequence number 0
Packet with sequence number 0 has timed out
0
Received ack with sequence number 0
Sliding receive window from 0 to 1
Received ack with sequence number 0
Sliding send window from 0 to 1
1
Received ack with sequence number 1
Sliding receive window from 1 to 2
Received ack with sequence number 1
Sliding send window from 1 to 2
2
Received ack with sequence number 2
Sliding receive window from 2 to 3
Received ack with sequence number 2
Sliding send window from 2 to 3
0
1
2
```