Sliding window pgm

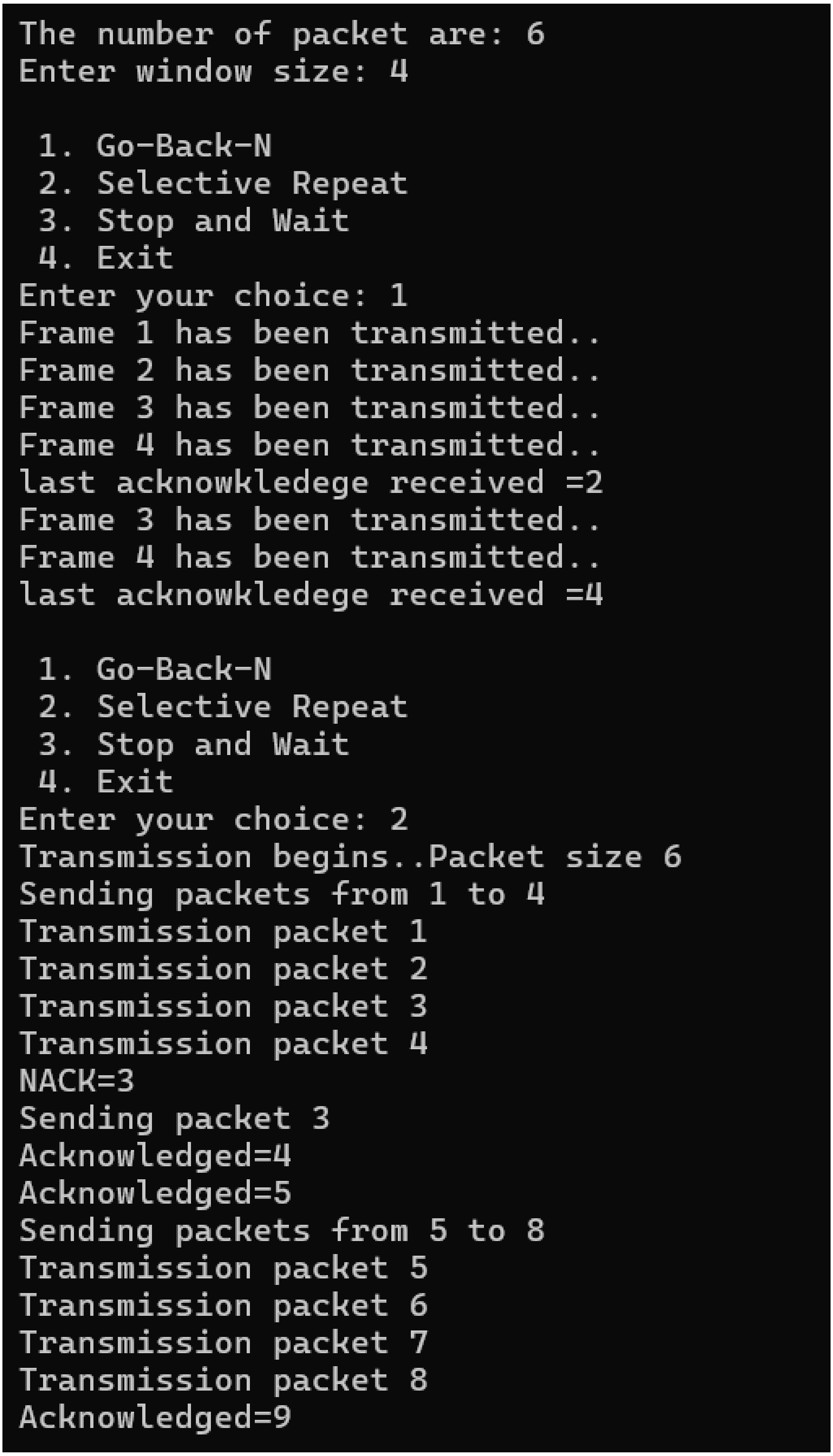
```
#include<stdio.h>
#include<stdlib.h>
int main(){
    int i,n,packet_index,ack,no_ack,choice,windowsize,sent;
    printf("The number of packet are: ");
    scanf("%d", &n);
    printf("Enter window size: ");
    scanf("%d", &windowsize);
    while(1){
        printf("\n 1. Go-Back-N\n 2. Selective Repeat\n 3. Stop and Wait\n 4. Exit\n");
        printf("Enter your choice: ");
        scanf("%d", &choice);
        switch(choice){
        case 1: //go back n
            sent = 0;
            while(1){
            for (i = 0;i<windowsize;i++){
                printf("Frame %d has been transmitted..\n",sent+1);
                sent++;
                if(sent==windowsize) break;
            }
            ack=rand()%windowsize+1;
            printf("last acknowkledege received =%d\n",ack);
            if(ack == windowsize) break;
            else sent = ack;
        }
        break;
        case 2: // selective repeat
        printf("Transmission begins..Packet size %d\n",n);
        i = 1;
        while(i<=n){
            printf("Sending packets from %d to %d\n",i,windowsize+i-1);
            for(packet_index=i;packet_index<windowsize+i;packet_index++)
                printf("Transmission packet %d\n",packet_index);
            no_ack = i + rand()%windowsize;
            if(no_ack==i){
                printf("Acknowledged=%d\n",windowsize+i);
                i=i+windowsize;
                continue;
            }
            printf("NACK=%d\n",no_ack);
            printf("Sending packet %d\n",no_ack);
            printf("Acknowledged=%d\n",no_ack+1);
            printf("Acknowledged=%d\n",windowsize+i);
            i=i+windowsize;
        }
        break;
        case 3: //stop and wait
            for (i=1;i<=n;i++){
                printf("The packet sent is: %d\n",i);
                if(rand()%2==1){
                    ack=rand()%2;
                    if(ack==1) printf("Acknowledgment number: %d\n",i+1);
                    else printf("No acknowledgment number:%d\n",i+1);
                }else{
                    printf("time out, resend\n");
                    i--;
                }
        }
            break;
        case 4:
            printf("Exited");exit(0);
        default:
            printf("Invalid input"); break;
        }
    }
}
```

Sliding window Algorithm

1. Read number or packets and window size
2. Read choice
3. If choice == 1 perform **go-back-N**
    a. The sender sends multiple frames, up to a certain window size, without waiting for individual acknowledgments.
    b. Receiver receives and checks frames:
        i. The receiver receives frames within the window.
        ii. If a frame is error-free:
            1. Receiver sends ACK for that frame and accepts it.
        iii. If a frame has errors:
            1. Receiver discards the frame without ACK.
    c. After sending the frames within the window, the sender waits for acknowledgments (ACKs) for the oldest unacknowledged frame
    d. If no ACK received within a specified time (timeout) for the oldest unacknowledged frame:
        i. Sender assumes frames after the oldest unacknowledged frame are lost or damaged
        ii. Return to step 1 (retransmit frames starting from the oldest unacknowledged frame).
    e. If the receiver receives a duplicate frame:
        i. Receiver ignores the duplicate frame.
        ii. Resend ACK for the last correctly received frame.
    f. If the receiver detects an error in a frame within the window:
        i. Receiver does not send ACK for that frame.
        ii. Receiver sends a negative acknowledgment (NAK) or ignores the frame, prompting the sender to retransmit frames from the oldest unacknowledged frame
    g. Repeat steps 1-6 until:
        i. All frames are successfully acknowledged.
        ii. Maximum number of retransmission attempts is reached.

```
4. Else if choice == 2 perform selective repeat
    a. The sender sends multiple frames individually, up to a certain window size.
    b. The receiver receives frames within the window:
        i. If a frame is error-free:
            1. Receiver sends ACK for that specific frame and accepts it.
        ii. 2.3. If a frame has errors:
            1. Receiver discards the frame without ACK.
        iii. Receiver buffers out-of-order frames within the window.
    c. After sending the frames within the window, the sender waits for
       acknowledgments (ACKs) for each individual frame.
    d. If no ACK received within a specified time (timeout) for a specific frame:
        i. Sender assumes that specific frame is lost or damaged.
        ii. Sender retransmits only that specific frame.
    e. If the receiver receives a duplicate frame:
        i. Receiver ignores the duplicate frame.
        ii. Resend ACK for that specific frame.
    f. If the receiver detects an error in a frame within the window:
        i. Receiver does not send ACK for that specific frame.
        ii. Receiver sends a negative acknowledgement (NAK) or ignores the
            frame, prompting the sender to retransmit only that specific frame.
    g. Repeat steps 1-6 until:
        i. All frames within the window are successfully acknowledged.
        ii. Maximum number of retransmission attempts is reached.
5. Else if choice == 3
    a. The sender sends a frame to the receiver.
    b. If the frame is error-free:
        i. Receiver sends ACK.
    c. If the frame has errors:
        i. Receiver discards the frame.
    d. Sender waits for ACK:
        i. After sending the frame, the sender waits for an acknowledgment
           (ACK) from the receiver.
    e. If no ACK received within a specified time (timeout):
        i. Sender assumes frame lost or damaged.
        ii. Return to step 1 (retransmit the frame).
    f. If the receiver detects an error in the frame:
        i. Receiver does not send ACK.
        ii. Receiver requests retransmission of the frame.
    g. Repeat steps 1-5 until:
        i. Sender receives ACK for the frame.
        ii. Maximum number of retransmission attempts is reached
6. Else if choice == 4
    a. Exit
7. Stop
```

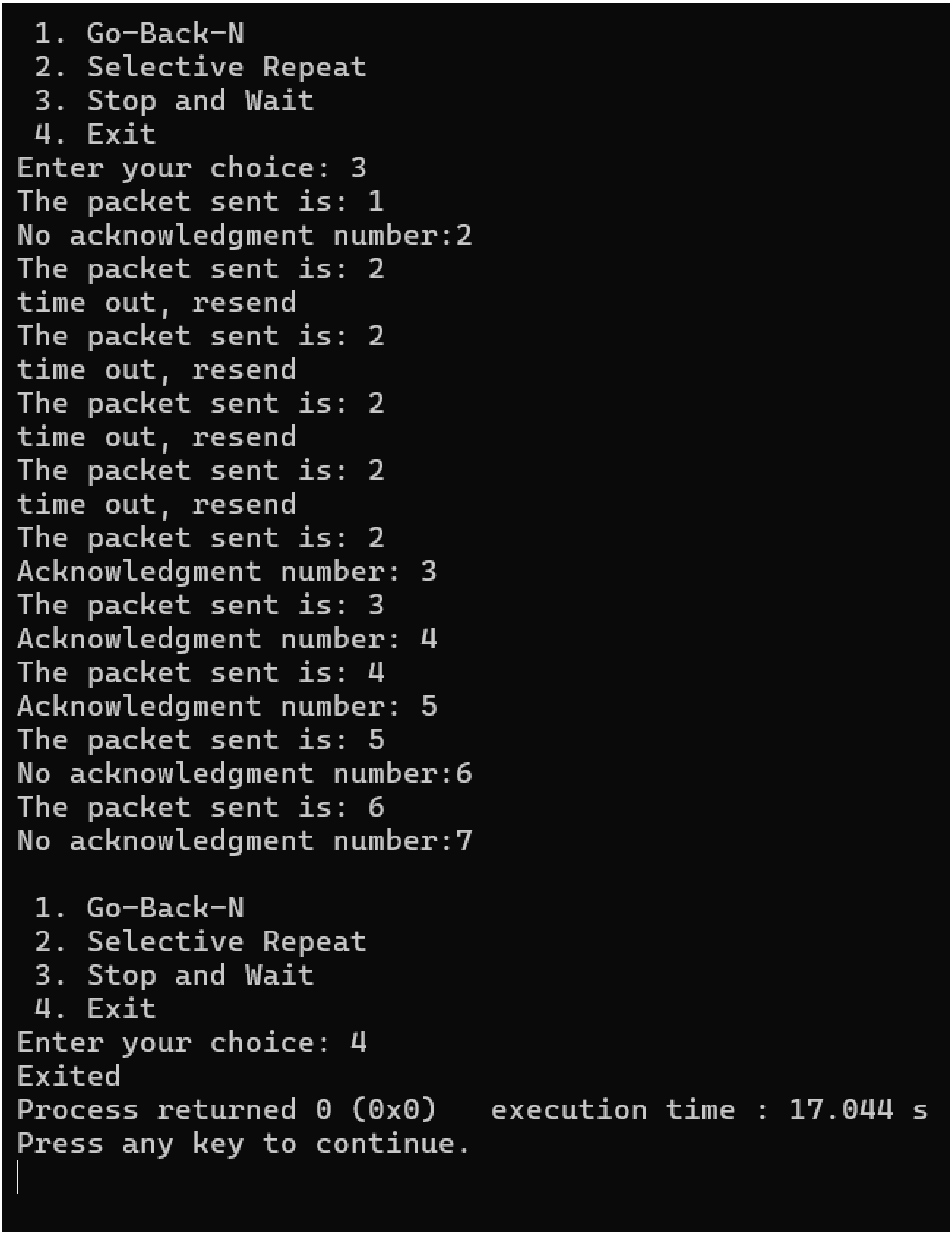
output:-



```
The number of packet are: 6
Enter window size: 4

 1. Go-Back-N
 2. Selective Repeat
 3. Stop and Wait
 4. Exit
Enter your choice: 1
Frame 1 has been transmitted..
Frame 2 has been transmitted..
Frame 3 has been transmitted..
Frame 4 has been transmitted..
last acknowkledege received =2
Frame 3 has been transmitted..
Frame 4 has been transmitted..
last acknowkledege received =4

 1. Go-Back-N
 2. Selective Repeat
 3. Stop and Wait
 4. Exit
Enter your choice: 2
Transmission begins..Packet size 6
Sending packets from 1 to 4
Transmission packet 1
Transmission packet 2
Transmission packet 3
Transmission packet 4
NACK=3
Sending packet 3
Acknowledged=4
Acknowledged=5
Sending packets from 5 to 8
Transmission packet 5
Transmission packet 6
Transmission packet 7
Transmission packet 8
Acknowledged=9
```



```
 1. Go-Back-N
 2. Selective Repeat
 3. Stop and Wait
 4. Exit
Enter your choice: 3
The packet sent is: 1
No acknowledgment number:2
The packet sent is: 2
time out, resend
The packet sent is: 2
time out, resend
The packet sent is: 2
time out, resend
The packet sent is: 2
time out, resend
The packet sent is: 2
Acknowledgment number: 3
The packet sent is: 3
Acknowledgment number: 4
The packet sent is: 4
Acknowledgment number: 5
The packet sent is: 5
No acknowledgment number:6
The packet sent is: 6
No acknowledgment number:7

 1. Go-Back-N
 2. Selective Repeat
 3. Stop and Wait
 4. Exit
Enter your choice: 4
Exited
Process returned 0 (0x0)   execution time : 17.044 s
Press any key to continue.
```