Aim: Write a program to implement flow Control at data link layer using SLIDING WINDOW PROTOCOL. Simulate the flow of frames from one node to another.

Create a Sender program with following features:

1] Input Window size from the user
2] Input a text message from the user
3] Consider 1 character per frame
4] Create a frame with following fields
5] Send the frames
6] Wait for the acknowledgement from the receiver.
7] Reader a file called Receiver Buffer
8] check Ack field for the Acknowledgement number.
9] If acknowledgment number is expected, Send new Set of frames accordingly, else if NACK is received, resent the frame accordingly.

Create a receiver file with following features

1] Reader a file called Sender-buffer
2] check the Frame no.
3] If the fame no- are as expected. write the appropriate Ack no. in the Receiver-Buffer file Else write NACK no. in the Receiver-Buffer file.

Sender.py:

```
import time
import os
def Sender (window_size, message) :
        Sender_buffer = "Sender_Buffer.txt"
        receiver_buffer = "Receiver_buffer.txt"
        frame_no = 0
        frames = [(i, message[i]) for i in range (len(message))]

    while frame_no < len(frames):
        for i in range (window_size):
            if frame_no + i < len (frames):
            print(f"Sending frame: {frames[frame_no+i]}")
            with open (Sender_buffer, 'a') as f:
                f.write (f"{frames[frame_no + i][0]}
                                {frames[frame_no+i][1]}\n")
            time.sleep(1)

        while True:
            if os.path.exists (receiver_buffer):
                with open (receiver_buffer, 'r') as f:
                    ack_no = int (f.read().strip())
                os.remove (receiver_buffer)
                break

        if ack_no >= frame_no:
            print (f"Ack received for frame: {ack_no}")
            frame_no = ack_no + 1
        else:
            print (f"NACK received for frame: {frame_no}")
if __name__ == "__main__":
    window_size = int(input ("Enter window size: "))
    message = input (" Enter message: ")
    Sender (window_size, message)
```

>>>

Receiver.py:

```
import time
import os
def receiver ():
  Sender_buffer = "Sender_buffer.txt"
  receiver_buffer = "receiver_buffer.txt"
  expected_frame_no = 0

  while True:
     if os.path.exists (Sender_buffer):
        with open (Sender_buffer, 'r') as f:
           lines = f.readlines ()
        os.remove (Sender_buffer)

        for line in lines:
           frame = line.strip().split ()
           frame_no = int (frame [0])
           data = frame [1]

           if frame_no == expected_frame_no:
              print (f"received frame: {frame_no}, data: {data
              with open (receiver_buffer, 'w') as f:
                 f.write (str (frame_no))
              expected_frame_no += 1

           else:
              print (f"unexpected frame: {frame_no},
                          expected: {expected_frame_no}")
              with open (receiver_buffer, 'w') as f:
                 f.write (str (expected_frame_no - 1))

if __name__ == "__main__":
   receiver ()
```

>>>

Output:

```
Enter the message to send: cat
Enter the window size: 3

Sender: Sending frames from position 1 to 3

Frame 1: 'c' sent
Frame 2: 'a' sent
Frame 3: 't' sent

Receiver: Acknowledgement received from frame 1
Sender: Sending frames from position 2 to 3

Frame 2: 'a' sent
Frame 3: 't' sent

Receiver: Acknowledgement received from frame 2
Sender: Sending frames from position 3 to 3

Frame 3: 't' sent
Receiver: Acknowledgement received for frame 3

All frames sent successfully
```

Result:

Thus the sliding window protocol is executed and verified.