Ex. NO: 7
Date: 11/9/24

Practial - 7
Sliding window

Aim :
write a program to implement flow control to data link layer using sliding window protocol. stimulate the flow of frames from one node to another.

Code :

Sender:

```
 import time
 def create_frames(window_size, message):
        frames = []
     for i in range (len(message)):
          frames. append((i %. window_.size, message[i])
   return frames
def  send_ frames (frames, start, window_size):
   print ("sending frames:")
   with open ("sender_Buffer.txt", "w") as f:
   for i in range (start, start+ window_size):
        if i <len (frames):
   print ~~(~~f" Frame No: {frames[i][0]}, Data:{
                                  frames[i][1]}")
   f.write (f"{frames[i][0]} {frames[i][1]}\n")
```

```
def check acknowledgem          size)
    with open (Receiver-buffer.txt, "r") as f:
        acks = f.readlines()
    acks = [int (x.strip()) for x in acks]
    if acks [0] == 1:                    # received
        print ("NACK received, resending frames.")
        return start
    else:
        print ("ACK received frames :", acks
        return start + len (acks)

def main ():
    window size = int (input ("Enter the wind
                                          size
    message = input ("Enter the message:")
    frames = create_ frames (window size, message
    current frame = 0
    while current frame < len (frames):
        send_ frames (frames, current frames, window
                                          size)
        print ("waiting for acknowledgement ....")
        time.sleep(2)                         # includes
                                                delay
        current frame = check acknowledgements (current-fra
                                          window size)
```

```
if _name_ =='_main_':
    main()
```

Receiver :

```
def read_sender_buffer():
with open ("sender_buffer.txt", "r") as f:
    frames = f.readlines()
frames = [frame.strip().split() for frame in
                                      frames]
return [(int (frame[0]), frame[1]) for frame in
                                      frames]
def send_acknowledgments (frames):
    expected_frame_no = 0
        acks = []
    for frame_no, data in frames:
        if frame no == expected frame_no:
    print (f" Received expected frame : {frame_no},
                                   Data: {data}")
          acks. append (frame no)
          expected_frame_no += 1
else:
    print(f" Error in frame : {frame_no}, Expected:
         {expected frame_no}. sending NACK.")
        acks = ...
        break
    with open ("Receiver_ Buffer.txt", "w") as f :
        for acks in acks:
            f.write (f "{ack} \n")

def main ():
    print ("Reading frames from sender_ buffer...")
    frames = Read_ sender_ buffer ( )
send _acknowledgments (frames)

if _name_ == "_main_" :
    main ()
```

output:-

```
Enter window size: 3
Enter text message : Hello
Sending Frames :
Frame NO : 0    Data : H
Frame No : 1    Data : e
Frame No : 2    Data : l
waiting for acknowledgment.....
```

~~Result:~~ NACK received , resending frames sender buffer.txt.

```
0 H
1 e
2 l
```

Receiver:

```
~~Reconding~~ Reading frames from sender_buffer...
Received expected frame 0, Data: H
Received expected frame 1, Data: e
Received expected frame 2, Data: l
```

Error:-

Senderbuffer.txt.

```
~~OH~~ 0 l
       2 0
```

```
Reading frames from sender_buffer....
Received expected frame 0, Data l
Error in frame 2, Expected: 1 sending NACK.
```

receiver_buffer.txt:

```
-1
```

Result:-

Thus the sliding window with error detection & correction is studied.

[illegible]