Algorithms for rdt1.0 rdt2.1 and rdt2.2

Algorithm for Sender (rdt1.0)

```
1.  Create a datagram socket.
2.  Get a message from keyboard.
3.  Create a packet
4.  Send the packet to the receiver.
5.  Go to the Step 2.
```

Algorithm for Receiver(rdt1.0)

```
1.  Create a datagram socket.
2.  Bind the socket to the well-known address (IP, Port).
3.  Wait for Sender to send a packet.
4.  After getting the packet convert and extract the message.
5.  Display the message.
6.  Go to the Step 3.
```

Algorithm for Sender (rdt2.1)

```
1.  Create a datagram socket.
2.  Get a message from keyboard.
3.  Create a packet (0, data, checksum)
4.  Send the packet to the receiver.
5.  Wait for ACK or NAK for 0
6.  Received Packet Check
    If(notcorrupt(rcvpkt) && isACK(rcvpkt) )
            Go to the Step 7

   If(corrupt(rcvpkt) || isNAK(rcvpkt) )

              Go to the Step 3

7.   Get another message from keyboard.
8.  Create a packet (1, data2, checksum)
9.  Wait for ACK or NAK for 1
10. Received Packet Check
    If (notcorrupt(rcvpkt) && isACK(rcvpkt))

            Go to Step 2

    If(corrupt(rcvpkt) || isNAK(rcvpkt))

           Go to the Step 8
```

Algorithm for Receiver(rdt2.1)

```
1.  Create a datagram socket.
2.  Bind the socket to the well-known address (IP, Port).
3.  Wait for Sender to send a packet.
4.  Received packet extract sequence number, msg, checksum.
    If (notCurrupted(rcvpkt) && hasSequenceNo(rcvpkt,0))
            Print the msg
            Create a packet (ACK, checksum)
            Send to the Sender

    Else If (Currupt(rcvpkt))
```

```
            Create a packet (NAK, checksum)

            Send to the Sender

            Go to Step 3

    Else

            //Change in Sequence no

            Create a Packet (ACK, checksum)

            Send to the Sender

            Go to Step 3

5.  Wait for Sender to send a Packet
6.  Received packet extract sequence number, msg, checksum.
    If (notCurrupted(rcvpkt) && hasSequenceNo(rcvpkt,1))
            Print the msg
            Create a packet (ACK, checksum)
            Send back to the Sender
            Go to Step 3

    Else If (Currupt(rcvpkt))

            Create a packet (NAK, checksum)

            Send to the Sender

            Go to Step 5

    Else

            //Change in Sequence no

            Create a Packet (ACK, Checksum)

            Send to the Sender

            Go to Step 5
```

Algorithm for Sender (rdt2.2)

```
1.  Create a datagram socket.
2.  Get a message from keyboard.
3.  Create a packet (0, msg, checksum)
4.  Send the packet to the receiver.
5.  Wait for ACK
6.  Received Packet Check
    If(notcorrupt(rcvpkt) && isACK(rcvpkt,0))
            Go to the Step 7

    If(corrupt(rcvpkt) || isACK(rcvpkt,1))

            //Retransmission

              Go to the Step 3

7.   Get another message from keyboard
8.  Create a packet (1, msg2, checksum)
9.  Wait for ACK
10. Received Packet Check
    If (notcorrupt(rcvpkt) && isACK(rcvpkt,0))

            Go to the Step 2
```

```
If(corrupt(rcvpkt) || isACK(rcvpkt,1))
        //Retransmission
        Go to the Step 8
```

Algorithm for Receiver(rdt2.2)

```
1. Create a datagram socket.
2. Bind the socket to the well-known address (IP, Port).
3. Wait for Sender to send a packet.
4. Received packet extract sequence number, msg, checksum.
   If (notCurrupted(rcvpkt) && hasSequenceNo(rcvpkt,0))
           Print the msg
           Create a packet (ACK, 0, checksum)
           Send back to the Sender
   Else If (Currupt(rcvpkt))
           Create a packet (ACK, 1, checksum)
           Send to the Sender
           Go to Step 3
   Else
           //Change in Sequence Number
           Create a Packet (ACK, 0, checksum)
           Send to the Sender
           Go to Step 3
5. Wait for Sender to send a Packet
6. Received packet extract sequence number, msg, checksum.
   If (notCurrupted(rcvpkt) && hasSequenceNo(rcvpkt,1))
           Print the msg
           Create a packet (ACK, 1, checksum)
           Send back to the Sender
           Go to Step 3
   Else If (Currupt(rcvpkt))
           Create a packet (ACK, 0, checksum)
           Send to the Sender
           Go to Step 5
   Else
           //Change in Sequence Number
           Create a Packet (ACK, 1, Checksum)
           Send to the Sender
           Go to Step 5
```