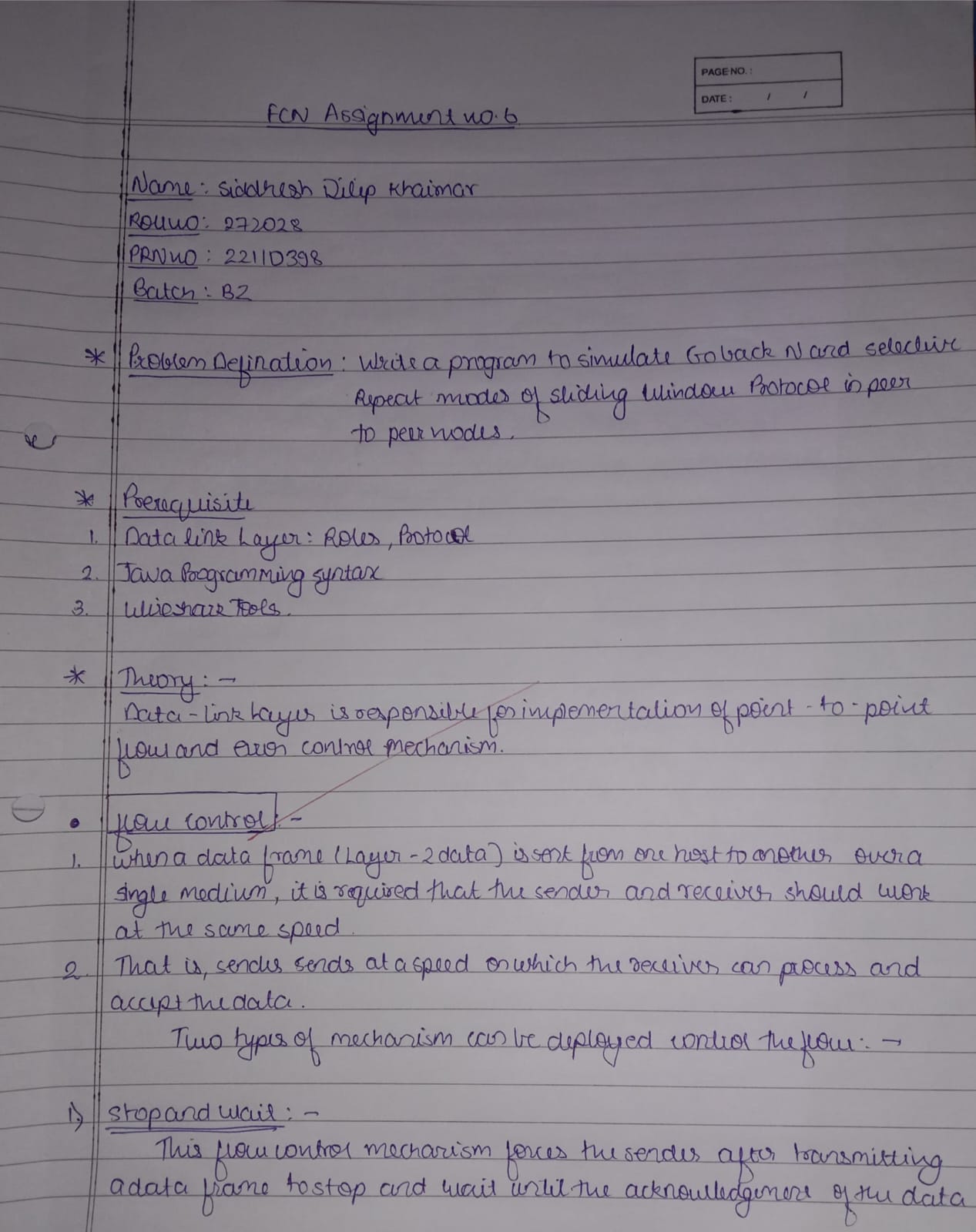
|  |  |  |  |
| --- | --- | --- | --- |
|  | Bansilal Ramnath Agarwal Charitable Trust's  Vishwakarma Institute of Information Technology  **Department of**  **Artificial Intelligence and Data Science** | | |
| Name: Siddhesh Dilip Khairnar | | | |
| Class: SY | Division: B | | Roll No: 272028 |
| Semester: IV | | Academic Year: 2022-2023 | |
| Subject Name & Code: Fundamentals of Computer Networks: ADUA22203 | | | |
| Title of Assignment: Write a program to simulate Go back N and Selective Repeat Modes of Sliding Window Protocol in Peer-to-Peer mode. | | | |

**ASSIGNMENT NO. 6**

A piece of paper with writing

Description automatically generated with medium confidenceText, letter

Description automatically generated

**Program and Output:**

1. **Go-Back-N ARQ:**

#Go-Back-N ARQ

import random

tf = int(input("Enter the Total number of frames: "))

N = int(input("Enter the Window Size: "))

def transmission(i, N, tf):

    tt = 0

    while i <= tf:

        z = 0

        for k in range(i, min(i + N, tf + 1)):

            print("Sending Frame", k, "...")

            tt += 1

        for k in range(i, min(i + N, tf + 1)):

            f = random.randint(0, 1)

            if not f:

                print("Acknowledgment for Frame", k, "...")

                z += 1

            else:

                print("Timeout!! Frame Number:", k, "Not Received")

                print("Retransmitting window...")

                break

        print()

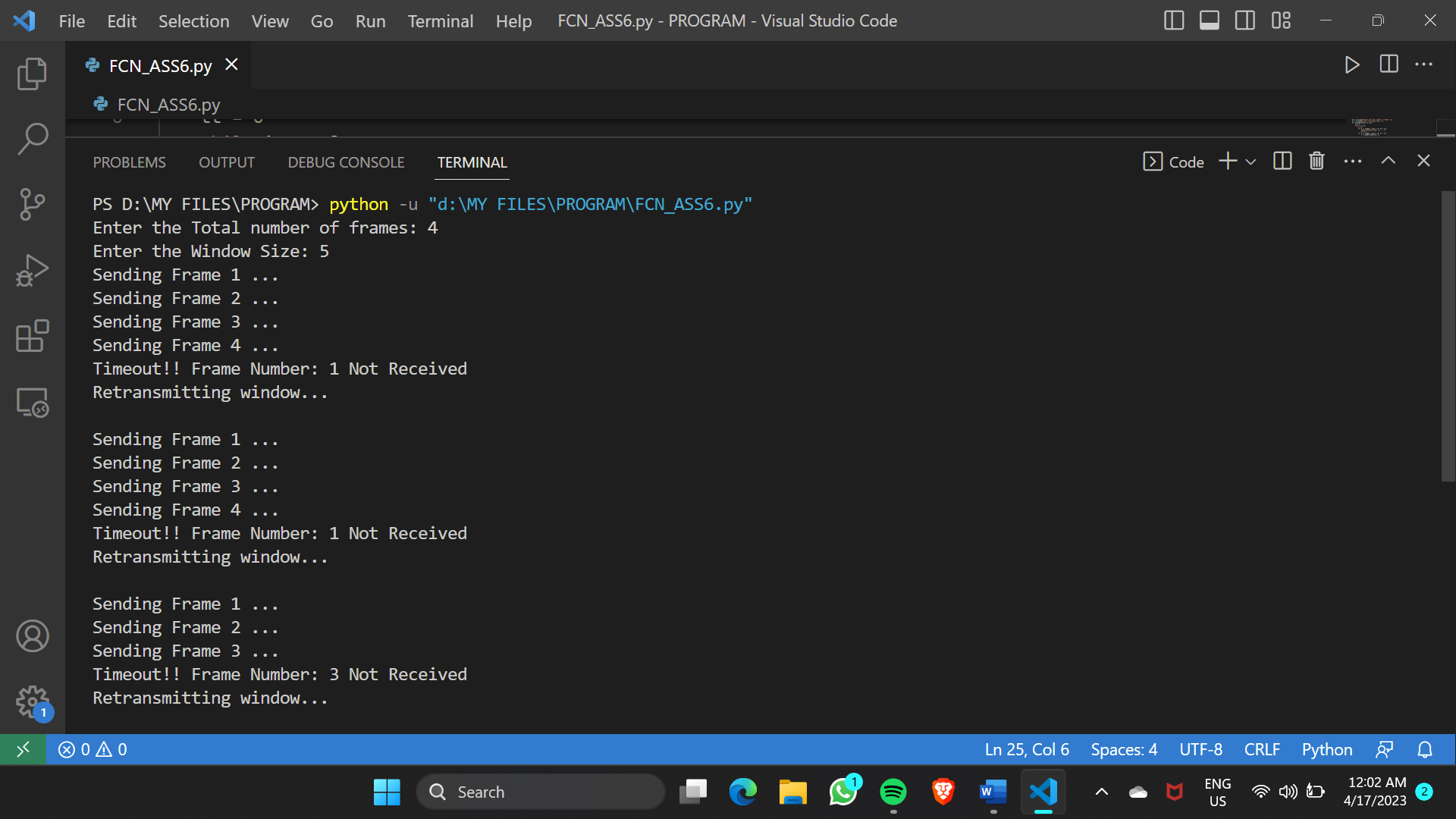
        i += z

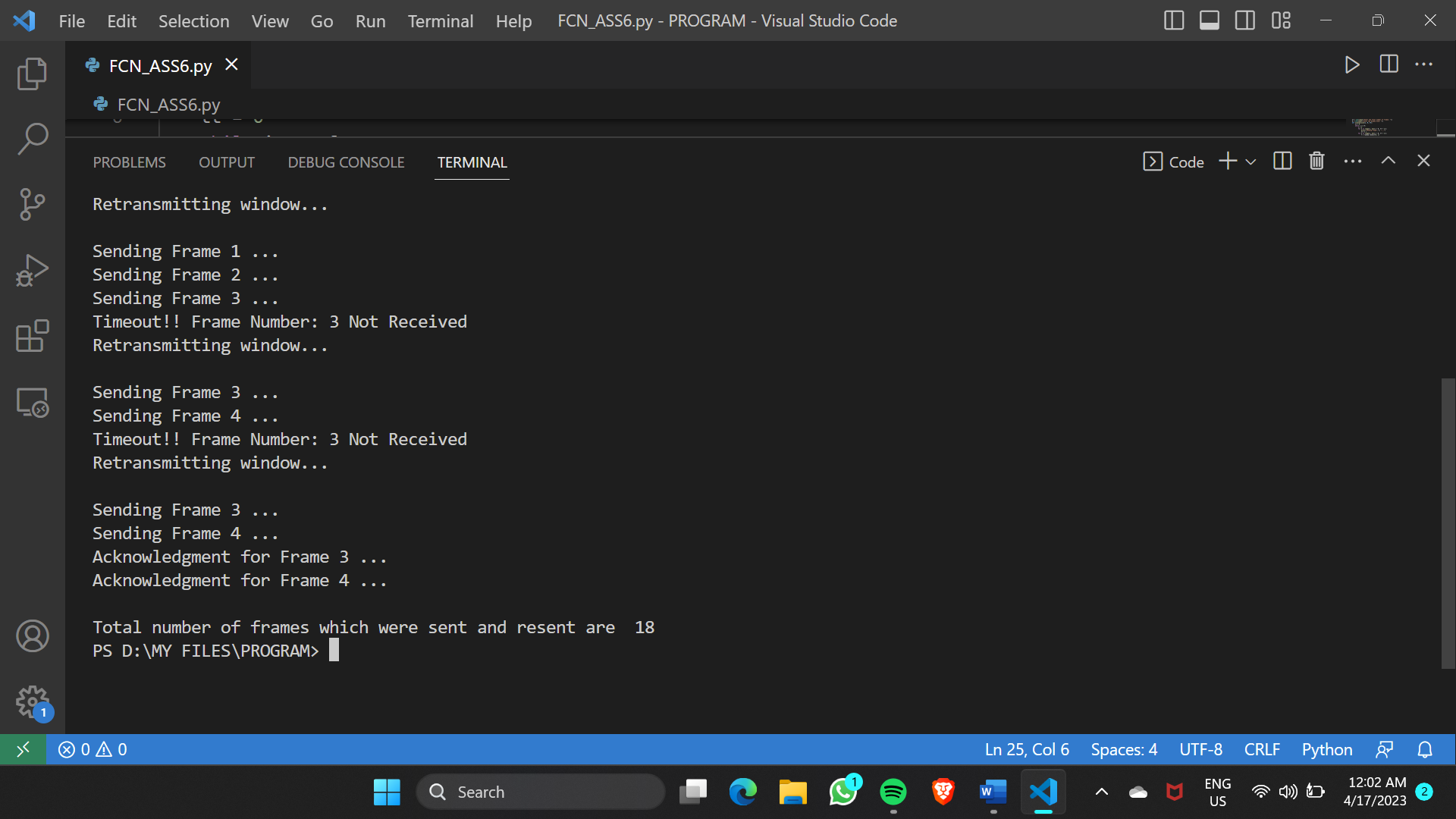
    return tt

i = 1

tt = transmission(i, N, tf)

print("Total number of frames which were sent and resent are ", tt)

****

****

1. **Selective Repeat ARQ:**

// Selective Repeat ARQ

#include <iostream>

using namespace std;

#include <conio.h>

#include <stdlib.h>

#include <time.h>

#include <math.h>

#define TOT\_FRAMES 500

#define FRAMES\_SEND 10

class sel\_repeat

{

private:

    int fr\_send\_at\_instance;

    int arr[TOT\_FRAMES];

    int send[FRAMES\_SEND];

    int rcvd[FRAMES\_SEND];

    char rcvd\_ack[FRAMES\_SEND];

    int sw;

    int rw; // tells expected frame

public:

    void input();

    void sender(int);

    void receiver(int);

};

void sel\_repeat::input()

{

    int n; // no. of bits for the frame

    int m; // no. of frames from n bits int i;

    int i;

    cout << "Enter the no. of bits for the sequence no.: ";

    cin >> n;

    m = pow(2, n);

    int t = 0;

    fr\_send\_at\_instance = (m / 2);

    for (i = 0; i < TOT\_FRAMES; i++)

    {

        arr[i] = t;

        t = (t + 1) % m;

    }

    for (i = 0; i < fr\_send\_at\_instance; i++)

    {

        send[i] = arr[i];

        rcvd[i] = arr[i];

        rcvd\_ack[i] = 'n';

    }

    rw = sw = fr\_send\_at\_instance;

    sender(m);

}

void sel\_repeat::sender(int m)

{

    for (int i = 0; i < fr\_send\_at\_instance; i++)

    {

        if (rcvd\_ack[i] == 'n')

            cout << "SENDER: Frame " << send[i] << " is sent\n";

    }

    receiver(m);

}

void sel\_repeat::receiver(int m)

{

    time\_t t;

    int f;

    int j;

    int f1;

    int a1;

    char ch;

    srand((unsigned)time(&t));

    for (int i = 0; i < fr\_send\_at\_instance; i++)

    {

        if (rcvd\_ack[i] == 'n')

        {

            f = rand() % 10;

            // if f-5 frame is discarded for some reason

            // else frame is correctly recieved

            if (f != 5)

            {

                for (int j = 0; j < fr\_send\_at\_instance; j++)

                    if (rcvd[j] == send[1])

                    {

                        cout << "Reciever:Frame " << rcvd[j] << " recieved correctly\n";

                        rcvd[j] = arr[rw];

                        rw = (rw + 1) % m;

                        break;

                    }

                int j;

                if (j == fr\_send\_at\_instance)

                    cout << "Reciever:Duplicate frame " << send[i] << " discarded\n";

                a1 = rand() % 5;

                // if al--3 then ack is lost

                // else recieved

                if (a1 == 3)

                {

                    cout << "(Acknowledgement " << send[i] << " lost)\n";

                    cout << "(Sender timeouts-->Resend the frame)\n";

                    rcvd\_ack[i] = 'n';

                }

                else

                {

                    cout << "(Acknowledgement " << send[i] << " recieved) \n";

                    rcvd\_ack[1] = 'p';

                }

            }

            else

            {

                int ld = rand() % 2;

                // if 0 then frame damaged

                // else frame lost

                if (ld == 0)

                {

                    cout << "RECEIVER : Frame " << send[i] << " is damaged\n";

                    cout << "RECEIVER : Negative Acknowledgement " << send[i] << " sent \n";

                }

                else

                {

                    cout << "RECEIVER : Frame " << send[i] << " is lost\n";

                    cout << "(SENDER TIMEOUTS-->RESEND THE FRAME)\n";

                }

                rcvd\_ack[i] = 'n';

            }

        }

    }

    for (int j = 0; j < fr\_send\_at\_instance; j++)

    {

        if (rcvd\_ack[j] == 'n')

            break;

    }

    int i = 0;

    for (int k = j; k < fr\_send\_at\_instance; k++)

    {

        send[i] = send[k];

        if (rcvd\_ack[k] == 'n')

            rcvd\_ack[i] = 'n';

        else

            rcvd\_ack[i] = 'p';

        i++;

    }

    if (i != fr\_send\_at\_instance)

    {

        for (int k = i; k < fr\_send\_at\_instance; k++)

        {

            send[k] = arr[sw];

            sw = (sw + 1) % m;

            rcvd\_ack[k] = 'n';

        }

    }

    cout << "want to continue (press y otherwise 1:)";

    cin >> ch;

    cout << "\n";

    if (ch == 'y')

        sender(m);

    else

        exit(0);

}

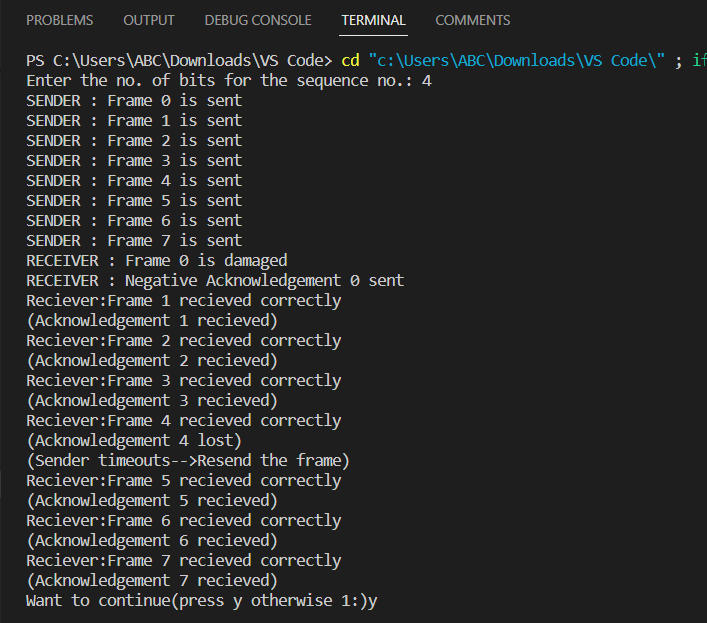
int main()

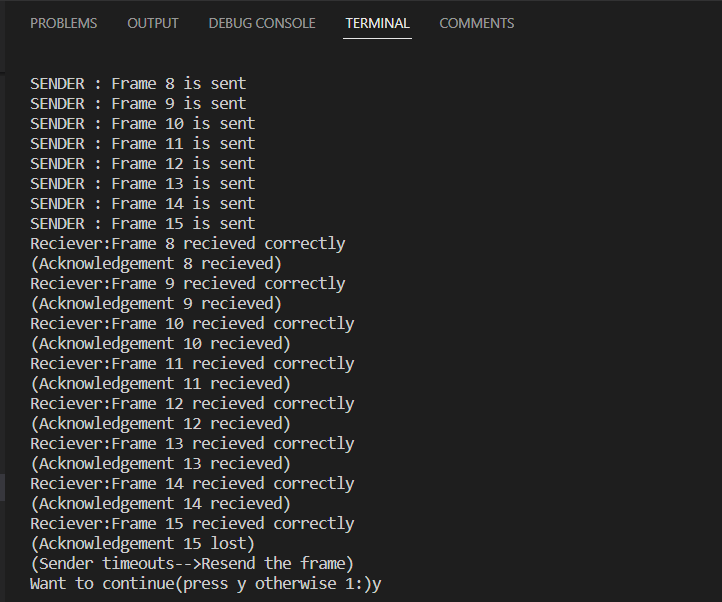
{

    sel\_repeat sr;

    sr.input();

}

****

****

**Conclusion:** Hence, we have studied and understood the concept of Go back N and Selective Repeat Modes of Sliding Window Protocol in Peer-to-Peer mode.