

CSE 1004

NETWORK AND COMMUNICATION



Assessment – 3

L23+L24 | PLBG17

WINTER SEMESTER 2020-21

by

SHARADINDU ADHIKARI

19BCE2105

PART 1**Go Back NRQ****Code:**

```
import random

n = int(input("Enter the total frames : "))
windSize = int(input("Enter the size of window : "));

recv=[]
for i in range(n):
    recv.append(0)

sent = 0

def lossless():
    global sent,windSize,n
    sent2 = sent+windSize
    end = n if n < sent2 else sent2
    for i in range(sent,end):
        recv[i]=1
    sent = end
    print("All frames Successfully received")

def frameLoss():
    global sent,windSize,n
    sent2 = sent +windSize
    end = n if n < sent2 else sent2
    framesLost = random.randint(sent,end)
    print("Frame number Lost "+str(framesLost))
    for i in range (sent,end):
        recv[i]=1;
    sent = end
    print("Frame "+str(framesLost)+" Not received")
    print("Resending All the frames")
    print("Discarding all frames except",str(framesLost))
    print("All frames Successfully received")

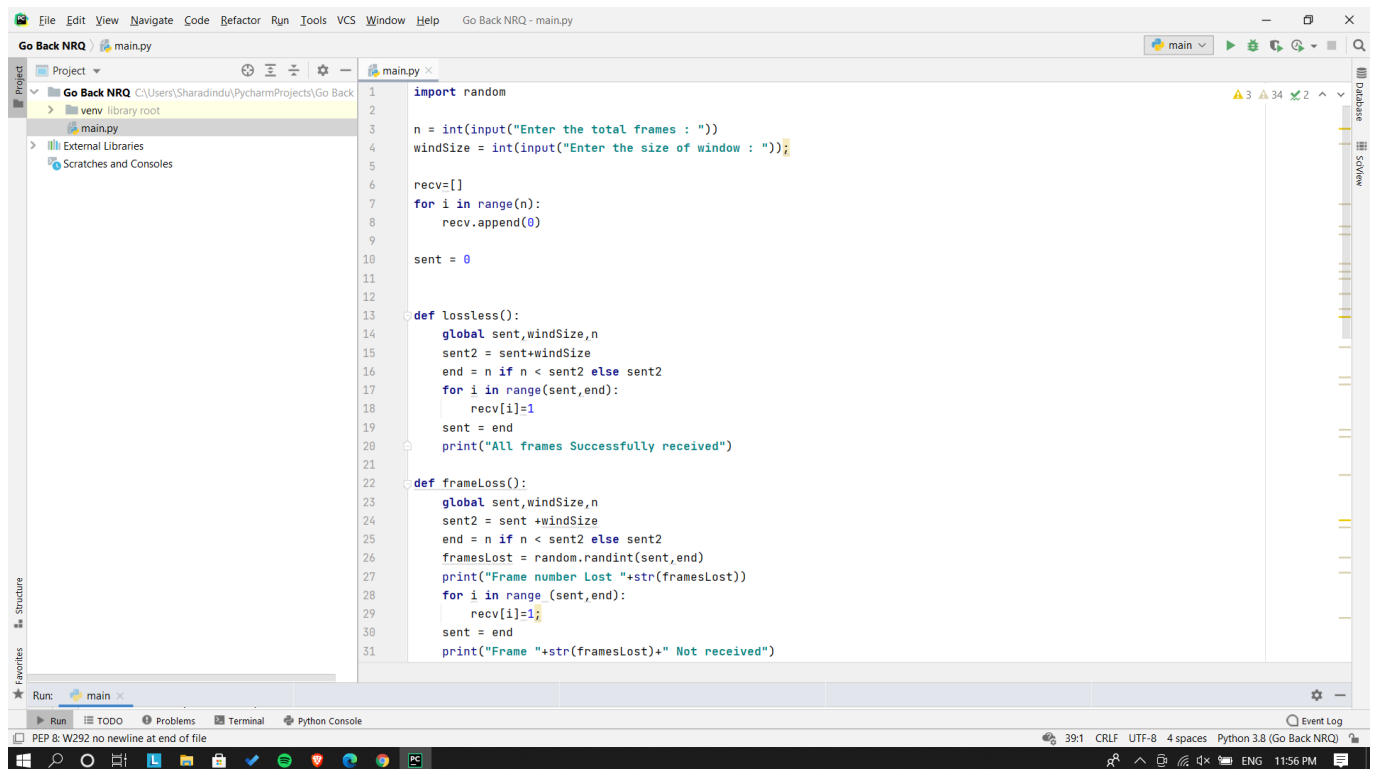
def ackLoss():
    global sent,windSize,n
    print("Sending Frames")
    print("All frames Successfully received")
    print("Didn't receive ACK resending Frames")
    sent2 = sent+windSize
    end = n if n < sent2 else sent2
    for i in range(sent,end):
        recv[i]=1
    sent = end
    print("On Receiver end:")
    print("Frames Already received, discarded the current ones")

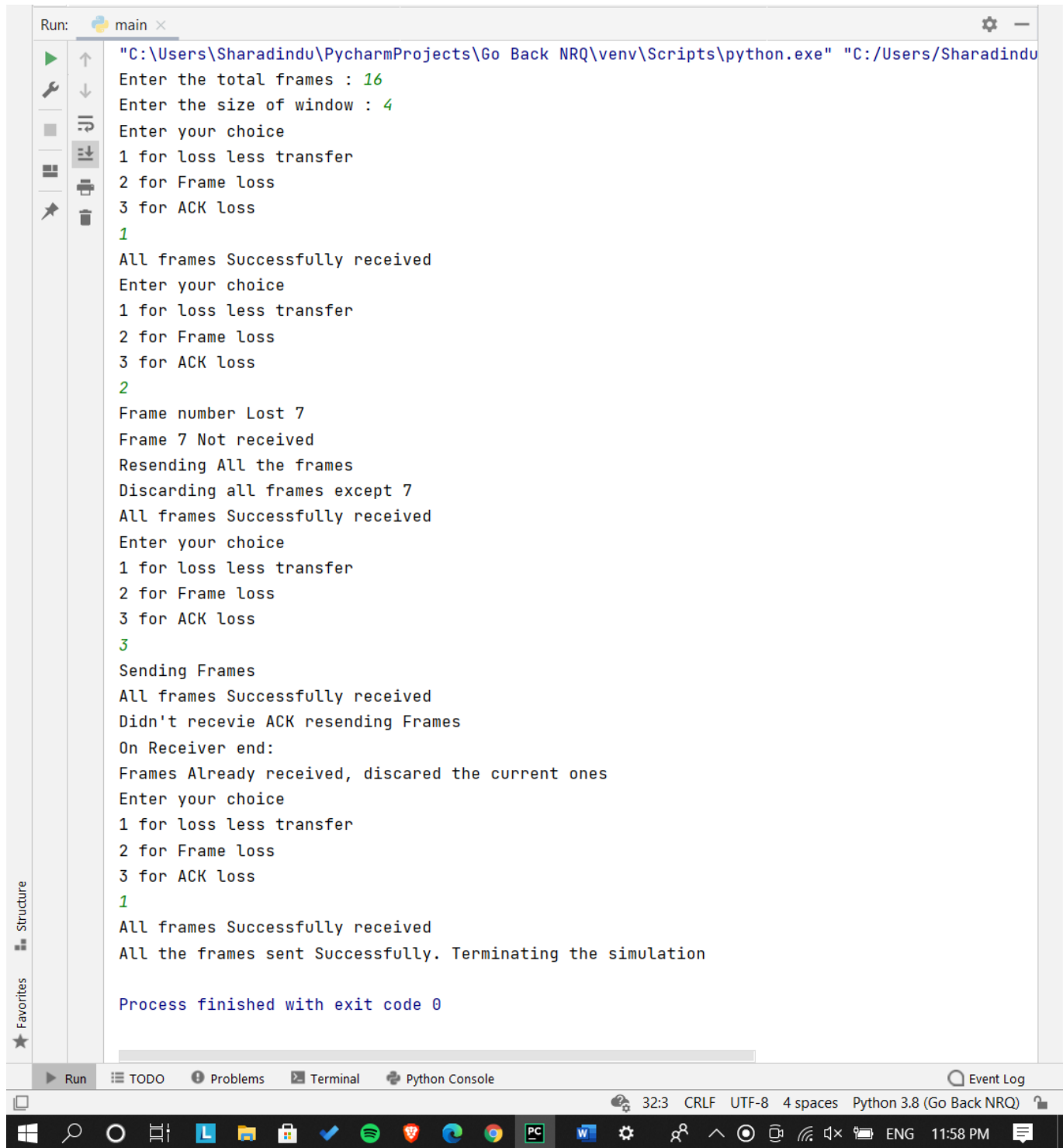
while(sent < n):
    choice = 0
    print("Enter your choice")
    print("1 for loss less transfer")
    print("2 for Frame loss")
    print("3 for ACK loss")
```

```
choice = int(input())
if(choice < 1 or choice > 3):
    print("Invalid choice, try again")
    continue
else:
    if(choice == 1):
        lossless()
    elif(choice == 2):
        frameLoss()
    else:
        ackLoss()

print("All the frames sent Successfully. Terminating the simulation")
```

Screenshot:



OUTPUT:

```
Run: main x
"C:\Users\Sharadindu\PycharmProjects\Go Back NRQ\venv\Scripts\python.exe" "C:/Users/Sharadindu
Enter the total frames : 16
Enter the size of window : 4
Enter your choice
1 for loss less transfer
2 for Frame loss
3 for ACK loss
1
All frames Successfully received
Enter your choice
1 for loss less transfer
2 for Frame loss
3 for ACK loss
2
Frame number Lost 7
Frame 7 Not received
Resending All the frames
Discarding all frames except 7
All frames Successfully received
Enter your choice
1 for loss less transfer
2 for Frame loss
3 for ACK loss
3
Sending Frames
All frames Successfully received
Didn't receive ACK resending Frames
On Receiver end:
Frames Already received, discarded the current ones
Enter your choice
1 for loss less transfer
2 for Frame loss
3 for ACK loss
1
All frames Successfully received
All the frames sent Successfully. Terminating the simulation

Process finished with exit code 0
```

PART 2**Selective Repeat****Code:**

```
import random
import time

print("Attack on Titan. by Sharad, 19BCE2105")

n = int(input("Enter the total frames : "))
windSize = int(input("Enter the size of window : "));

recv=[]
for i in range(n):
    recv.append(0)

sent = 0

def lossless():
    global sent,windSize,n
    sent2 = sent+windSize
    end = n if n < sent2 else sent2
    for i in range(sent,end):
        recv[i]=1
    sent = end
    print("C : All frames Successfully received")

def frameLoss():
    global sent,windSize,n
    sent2 = sent +windSize
    end = n if n < sent2 else sent2
    framesLost = random.randint(sent,end)
    print("Frame number Lost "+str(framesLost))
    for i in range (sent,end):
        recv[i]=1;
    sent = end
    print("C : Frame "+str(framesLost)+" Not received")
    print("S : Resending "+str(framesLost)+" the frames")
    print("C : Discarding all frames except",str(framesLost))
    print("C : All frames Successfully received")

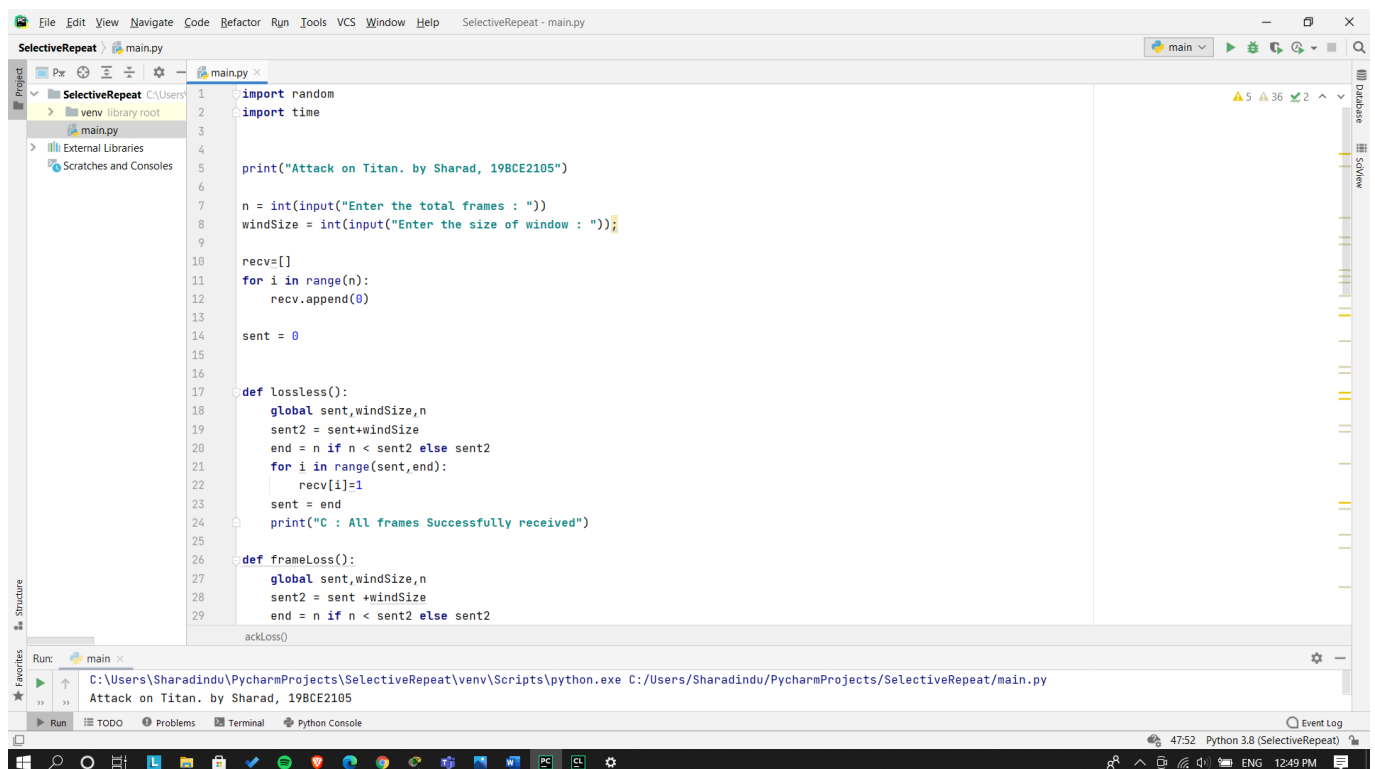
def ackLoss():
    global sent,windSize,n
    print("S : Sending Frames")
    print("C : All frames Successfully received")
    time.sleep(5);
    print("S : Timeout");
    print("S: Didn't receive ACK resending Frames")
    sent2 = sent+windSize
    end = n if n < sent2 else sent2
    for i in range(sent,end):
        recv[i]=1
    sent = end
    print("C : Frames Already received, discarded the current ones")
    print("S : ACK Received")

while(sent <n):
    choice = 0
```

```
print("Enter your choice")
print("1 for loss less transfer")
print("2 for Frame loss")
print("3 for ACK loss")
choice = int(input())
if(choice < 1 or choice > 3):
    print("Invalid choice, try again")
    continue
else:
    if(choice == 1):
        lossless()
    elif(choice ==2):
        frameLoss()
    else:
        ackLoss()

print("All the frames sent Successfully. Terminating the simulation")
```

Screenshot:



OUTPUT:

The screenshot displays the PyCharm IDE with a project named 'SelectiveRepeat'. The main.py file contains the following Python code:

```
1 import random
2 import time
3
4
5 print("Attack on Titan. by Sharad, 19BCE2105")
6
7 n = int(input("Enter the total frames : "))
8 windSize = int(input("Enter the size of window : "));
9
10 recv=[]
11 for i in range(n):
12     recv.append(0)
13
14 sent = 0
15
16
17 def lossless():
18     global sent,windSize,n
19     sent2 = sent+windSize
20     end = n if n < sent2 else sent2
21     for i in range(sent,end):
22         recv[i]=1
23
24 while (sent < n) > else > if (choice == 1)
```

The Run console shows the execution of the program:

```
Run: main
C:\Users\Sharadindu\PycharmProjects\SelectiveRepeat\venv\Scripts\python.exe C:/Users/Sharadindu/PycharmProjects/SelectiveRepeat/main.py
Attack on Titan. by Sharad, 19BCE2105
Enter the total frames : 16
Enter the size of window : 4
Enter your choice
1 for loss less transfer
2 for Frame loss
3 for ACK loss
1
C : All frames Successfully received
Enter your choice
1 for loss less transfer
2 for Frame loss
3 for ACK loss
2
Frame number Lost 4
C : Frame 4 Not received
S : Resending 4 the frames
C : Discarding all frames except 4
C : All frames Successfully received
Enter your choice
1 for loss less transfer
2 for Frame loss
3 for ACK loss
3
S : Sending Frames
C : All frames Successfully received
1
S : Timeout
S: Didn't receive ACK resending Frames
C : Frames Already received, discarded the current ones
S : ACK Received
Enter your choice
1 for loss less transfer
2 for Frame loss
3 for ACK loss
C : All frames Successfully received
All the frames sent Successfully. Terminating the simulation

Process finished with exit code 0
```

PART 3**Logical Addressing****Instructions**

1. write a program to check whether the given address is IPV4 or IPV6
2. write a program to determine the class, network id, host id, default mask.

Code:**Problem 1 sol**

```
class Solution(object):
    def validIPAddress(self, IP):
        def isIPv4(s):
            try: return str(int(s)) == s and 0 <= int(s) <= 255
            except: return False
        def isIPv6(s):
            if len(s) > 4:
                return False
            try : return int(s, 16) >= 0 and s[0] != '-'
            except:
                return False
        if IP.count(".") == 3 and all(isIPv4(i) for i in IP.split(".")):
            return "IPv4"
        if IP.count(":") == 7 and all(isIPv6(i) for i in IP.split(":")):
            return "IPv6"
        return "Invalid"

ob = Solution()
print("Enter the IP address")
pl=input()
print(ob.validIPAddress(pl))
```



The screenshot shows a code editor window with a toolbar at the top containing icons for Run, Debug, Stop, Share, Save, Beautify, and a download icon. The language is set to Python 3. The code in the editor matches the provided code block. Below the editor is a terminal window. The terminal shows the prompt 'Enter the IP address', the user input '75.123.253.255', and the program output 'IPv4'. At the bottom, it says '...Program finished with exit code 0' and 'Press ENTER to exit console.'

```
main.py
1- class Solution(object):
2-     def validIPAddress(self, IP):
3-         def isIPv4(s):
4-             try: return str(int(s)) == s and 0 <= int(s) <= 255
5-             except: return False
6-         def isIPv6(s):
7-             if len(s) > 4:
8-                 return False
9-             try : return int(s, 16) >= 0 and s[0] != '-'
10-            except:
11-                return False
12-        if IP.count(".") == 3 and all(isIPv4(i) for i in IP.split(".")):
13-            return "IPv4"
14-        if IP.count(":") == 7 and all(isIPv6(i) for i in IP.split(":")):
15-            return "IPv6"
16-        return "Invalid"
17- ob = Solution()
18- print("Enter the IP address")
19- pl=input()
20- print(ob.validIPAddress(pl))

input
Enter the IP address
75.123.253.255
IPv4

...Program finished with exit code 0
Press ENTER to exit console.
```



```
input
Enter the IP address
2401:4900:28c8:d220:640f:7134:e3e:f3a9
IPv6

...Program finished with exit code 0
Press ENTER to exit console.
```

Problem 2 sol

```
#include<stdio.h>

#include<string.h>

// Function to find out the Class
char findClass(char str[]) {
    // storing first octet in arr[] variable
    char arr[4];
    int i = 0;
    while (str[i] != '.') {
        arr[i] = str[i];
        i++;
    }
    i--;

    int ip = 0, j = 1;
    while (i >= 0) {
        ip = ip + (str[i] - '0') * j;
        j = j * 10;
        i--;
    }

    if (ip >= 1 && ip <= 126)
        return 'A';

    else if (ip >= 128 && ip <= 191)
        return 'B';

    else if (ip >= 192 && ip <= 223)
        return 'C';

    else if (ip >= 224 && ip <= 239)
        return 'D';

    else
        return 'E';
}

// Function to separate Network ID as well as
// Host ID and print them
void separate(char str[], char ipClass) {

    char network[12], host[12];
    for (int k = 0; k < 12; k++)
        network[k] = host[k] = '\\0';

    if (ipClass == 'A') {
        int i = 0, j = 0;
        while (str[j] != '.')
            network[i++] = str[j++];
        i = 0;
    }
}
```

```
j++;
while (str[j] != '\0')
    host[i++] = str[j++];
printf("Network ID is %s\n", network);
printf("Host ID is %s\n", host);
printf("Default Mask is : 255.0.0.0\n");
} else if (ipClass == 'B') {
    int i = 0, j = 0, dotCount = 0;

    while (dotCount < 2) {
        network[i++] = str[j++];
        if (str[j] == '.')
            dotCount++;
    }
    i = 0;
    j++;

    while (str[j] != '\0')
        host[i++] = str[j++];

    printf("Network ID is %s\n", network);
    printf("Host ID is %s\n", host);
    printf("Default Mask is : 255.255.0.0\n");
} else if (ipClass == 'C') {
    int i = 0, j = 0, dotCount = 0;

    while (dotCount < 3) {
        network[i++] = str[j++];
        if (str[j] == '.')
            dotCount++;
    }
    i = 0;
    j++;

    while (str[j] != '\0')
        host[i++] = str[j++];

    printf("Network ID is %s\n", network);
    printf("Host ID is %s\n", host);
    printf("Default Mask is : 255.255.255.0\n");
} else
    printf("In this Class, IP address is not"
           " divided into Network and Host ID\n");
}

int main()
{
    char str[100];
    printf("Enter the IP Address\n");
    scanf("%s", &str);
    char ipClass = findClass(str);
    printf("Given IP address belongs to Class %c\n", ipClass);
    separate(str, ipClass);
    return 0;
}
```

```
1 #include<stdio.h>
2
3 #include<string.h>
4
5 // Function to find out the Class
6 char findClass(char str[]) {
7     // storing first octet in arr[] variable
8     char arr[4];
9     int i = 0;
10    while (str[i] != '.') {
11        arr[i] = str[i];
12        i++;
13    }
14    i--;
15
16    int ip = 0, j = 1;
17    while (i >= 0) {
18        ip = ip + (str[i] - '0') * j;
19        j = j * 10;
20        i--;
21    }
22
23    if (ip >= 1 && ip <= 126)
24        return 'A';
25
26    else if (ip >= 128 && ip <= 191)
27        return 'B';
28
29    // if (ip >= 192 && ip <= 223)
30    //     return 'C';
31
32    // if (ip >= 224 && ip <= 255)
33    //     return 'D';
34
35    return '\0';
36 }
```

```
Enter the IP Address
75.123.253.255
Given IP address belongs to Class A
Network ID is 75
Host ID is 123.253.255
Default Mask is : 255.0.0.0
```

```
...Program finished with exit code 0
Press ENTER to exit console.
```

PART 4**Routing Algorithm****Instructions**

Write a program for Dijkstra shortest path algorithm(Link State)

Code:

```
n = int(input("Enter the no. of routers:"))

print("Enter the matrix: ")

m = []

for i in range(0, n):
    l = list(map(int, (input().rstrip().split(" "))))

    m.append(l)

def min_dist(l, q):
    min = 9999999
    n = -1
    for node, dist in l.items():
        if node not in q:
            continue
        if (dist < min):
            n = node
            min = dist

    return n

def dijsktra(node_number, matrix):
    q = set()
    dist = {}
    prev = {}

    n = len(matrix)

    for i in range(0, n):
        dist[i] = 9999999
        prev[i] = -1
        q.add(i)

    dist[node_number] = 0

    while (len(q) != 0):
        u = min_dist(dist, q)
        if u in q:
            q.remove(u)

        for v in q:
            if matrix[u][v] == -1:
                continue

            alt = dist[u] + matrix[u][v]

            if (alt < dist[v]):
                dist[v] = alt
                prev[v] = u
```

```

    return dist, prev

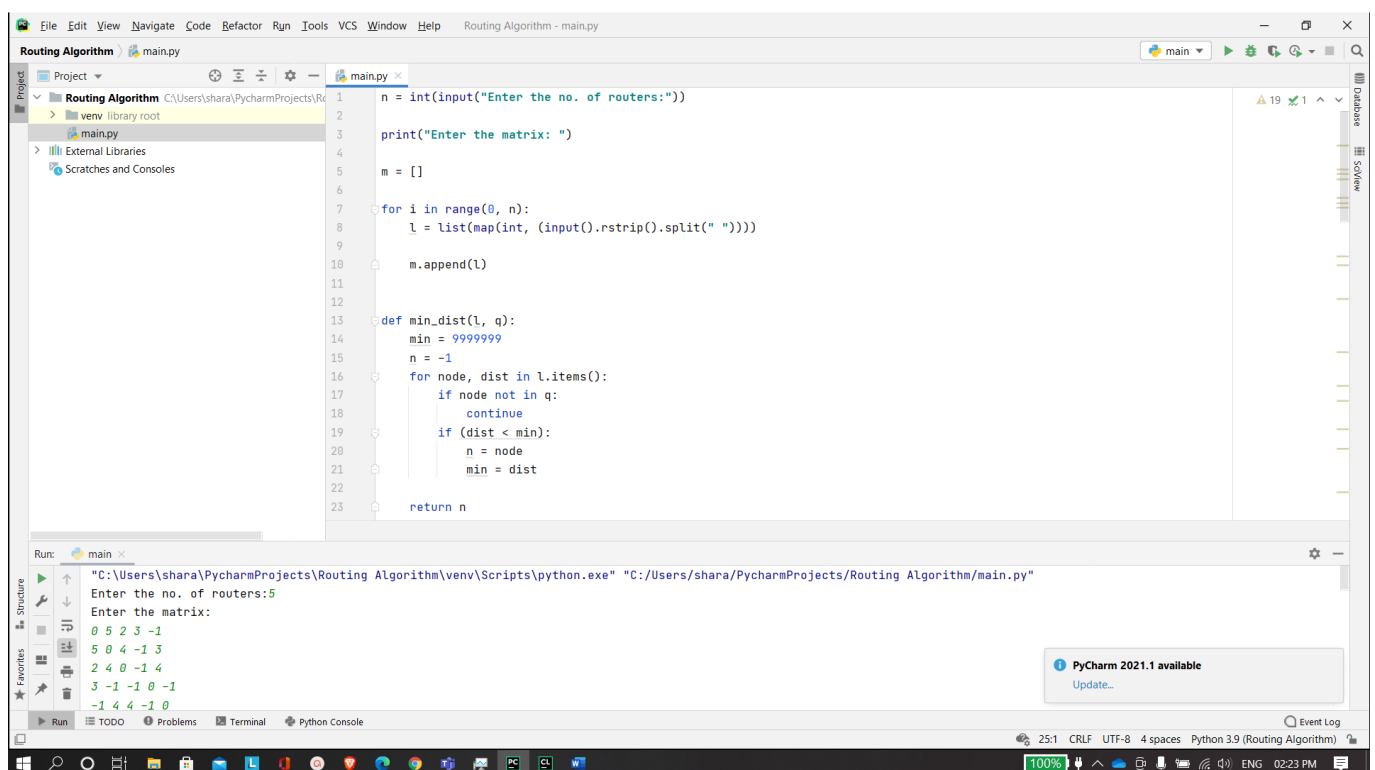
def find_next(p, c, t):
    s = []
    if p[t] != -1 or t == c:
        while (t != -1):
            s.append(t)
            t = p[t]
    return s[-2]

def print_table(d, p, n):
    print("Node|Distance|Next")
    for node, dist in d.items():
        print((chr(ord('A') + node)) + "    |" + str(dist), end="")
        if (p[node] == -1 or p[node] == n):
            print("    |")
        else:
            print("    |" + chr(ord('A') + find_next(p, n, node)))

for i in range(0, len(m)):
    d, p = dijkstra(i, m)
    print("Routing table for " + chr(ord('A') + i) + ": ")
    print_table(d, p, i)

```

Screenshot:



OUTPUT:

```

"C:\Users\shara\PycharmProjects\Routing Algorithm\venv\Scripts\python.exe" "C:/Users/shara/Py
Enter the no. of routers:5
Enter the matrix:
0 5 2 3 -1
5 0 4 -1 3
2 4 0 -1 4
3 -1 -1 0 -1
-1 4 4 -1 0
Routing table for A:
Node|Distance|Next
A |0 | -
B |5 | -
C |2 | -
D |3 | -
E |6 | C
Routing table for B:
Node|Distance|Next
A |5 | -
B |0 | -
C |4 | -
D |8 | A
E |3 | -
Routing table for C:
Node|Distance|Next
A |2 | -
B |4 | -
C |0 | -
D |5 | A
E |4 | -
Routing table for D:
Node|Distance|Next
A |3 | -
B |8 | A
C |5 | A
D |0 | -
E |9 | A
Routing table for E:
Node|Distance|Next
A |6 | C
B |4 | -
C |4 | -
D |9 | C
E |0 | -

Process finished with exit code 0

```