B M S. COLLEGE OF ENGINEERING

(Autonomous Institution)

RECORD OF PRACTICAL WORK

NAME : Ravi Sajjanar

SUBJECT : CN Lab

SEMESTER : Vth BRANCH : CSE

ROLL NO : USN : 1BM19CS127

Scanned by CamScanner

B M S. COLLEGE OF ENGINEERING

(Autonomous Institution)

LABORATORY CERTIFICATE

This is to Certify that Mr. / Ms. RAVI SAJJANAR

has Satisfactorily completed the course of experiments in Practical Computer Networks Lab Prescribed by the Visvesvaraya Technological University for 5th Semester B.E (CSE) Course in the Laboratory of the college in the year 2021 - 2022

Head of the Department

Staff incharge of the Batch

Date : .....

Marks	
Maximum	Obtained

Name of the Candidate : Ravi Sajjanar

Roll No : ..... USN : 1BM19CS127

Signature of the Candidate

Scanned by CamScanner

Particulars of the Experiments Performed

CONTENTS

Scanned by CamScanner

Particulars of the Experiments Performed

CONTENTS

Scanned by CamScanner

Date 20/12/21

Expt. No. 07 Page No. 16

Question: Write a program for error detecting code using CRC-CCITT (16 bits).

Aim: Error detection using CRC

Program:

```
def xor1(a,b):
    x = ""
    for i in range(1, len(a)):
        if a[i] == b[i]:
            x += "0"
        else:
            x += "1"
    return x

def modulo2(divident, divisor):
    divlen = len(divisor)
    temp = divident[0:divlen]
    while (divlen < len(divident)):
        if temp[0] == "1":
            temp = xor1(temp, divisor)
                + divident[divlen]
        else:
            temp = temp[1:divlen] +
                divident[divlen]
```

Teacher's Signature : _____

Scanned by CamScanner

Enter the key polynomial:

$x16 + x12 + x4 + 1$

[16, 12, 4, 0]

1000100000001000L

Enter the data polynomial:

$x15 + x12 + x11 + x8 + x7 + x4 + x3 + 1$

[15, 12, 11, 8, 7, 4, 3, 0]

1001100110011001

1000100000001000 ~ 1001100110011001

Remainder = 0000100100010010

Code = 1001100110011001 0000100100010010

Remainder we get when we do not have error = 0000 0000 0000 0000

Remainder we get when we have error = 0011 0011 0011 00000

```
        divlen += 1
        if temp[0] == "1":
            temp = xor(temp, divisor)
        if len(temp) < len(divisor):
            return "0" + temp
        return temp

def encode(data, key):
    append = data + "0" * (len(key))
    rem = modulo2(append, key)
    print("remainder =" + rem)
    code = data + rem
    print("code =" + code)

    rem = modulo2(code, key)
    print("Remainder we get when we do not
        have error =" + rem)
    code = code.replace("011", "101")
    rem = modulo2(code, key)
    print("Remainder we get when we have
            error =" + rem)

def polytobin(string):
    keys = []
    key = ""
    for i in string
        if i == '+':
            keys.append(int(key[1:]))
```

Teacher's Signature : _____

Scanned by CamScanner

```
            key = " "
            continue
        key += i
    if key != " ":
        keys.append(0)
    bina = ""
    j=0
    print(keys)
    for i in range(keys[0], -1, -1):
        if i == (keys[j]):
            bina += "1"
            j += 1
        else:
            bina += "0"
    print(bina)
    return bina

string = input("Enter the key polynomial: \n")
key = polytobin(string)
string = input("Enter the data polynomial:\n")
data = polytobin(string)
print(key, data)
encode(data, key)
```

Teacher's Signature : _____

Scanned by CamScanner

Date 27/12/21

Expt. No. 08 Page No. 19

Question: Write a program for distance vector algorithm to find suitable path for transmission.

Aim: Distance Vector algorithm to find path of shortest cost.

Program:

```
class Graph:
    def __init__(self, vertices):
        self.V = vertices
        self.graph = []

    def add_edge(self, s, d, w):
        self.graph.append([s, d, w])

    def print_solution(self, dist, src, next):
        print("Routing table for ", src)
        print("Dest \t Cost \t Next hop")
        for i in range(self.V):
            print("{0} \t {1} \t {2}".
            format(i, dist[i], next[i]))

    def bellman_ford(self, src):
        dist = [99] * self.V
        dist[src] = 0
        next_hop = {src : src}
```

Teacher's Signature : _____

Scanned by CamScanner

Output:

Enter the adjacency matrix:
Enter 99 for infinity

0	1	5	99	99
1	0	3	99	9
5	3	0	4	99
99	99	4	0	2
99	9	99	2	0

Routing table for 0

Dest	Cost	NextHop
0	0	0
1	1	1
2	4	1
3	8	1
4	10	1

Routing table for 1

Dest	Cost	NextHop
0	1	0
1	0	1
2	3	2
3	7	2
4	9	4

Scanned by CamScanner

```
for _ in range (self.V-1):
    for s,d,w in self.graph:
        if dist[s] != 99 and dist[s]+
                                w < dist[d]:
            dist[d] = dist[s] + w
            if s == src:
                next_hop[d] = d
            elif s in next_hop:
                next_hop[d] = next_hop[s]

for s,d,w in self.graph:
    if dist[s] != 99 and dist[s]+
                                w < dist[d]:
        print("Graph Contains negative
            weight cycle")
        return
self.print_solution(dist, src, next_hop)

def main():
    matrix = []
    print("Enter the no. of routers:")
    n = int(input())
    print("Enter the adjacency matrix:
            Enter 99 for infinity")
    for i in range(0,n):
        a = list(map(int, input().split(" ")))
        matrix.append(a)
```

Teacher's Signature : _____

Scanned by CamScanner

Routing table for 2

Dest	Cost	NextHop
0	4	1
1	3	1
2	0	2
3	4	3
4	6	3

Routing table for 3

Dest	Cost	NextHop
0	8	2
1	7	2
2	4	2
3	0	3
4	2	4

Routing table for 4

Dest	Cost	NextHop
0	10	1
1	9	1
2	6	3
3	2	3
4	0	4

Scanned by CamScanner

```
g= Graph(n)
for i in range(0,n):
    for j in range(0,n):
        g.add_egde(i,j,matrix[i][j])

for k in range(0,n):
    g.bellman_ford(k)

main()
```

Teacher's Signature : _____

Scanned by CamScanner

Date 27/12/21

Expt. No. 9 Page No. 22

Question: Implement Dijkstra's to compute the shortest path for a given topology

Aim: Dijkstras Algorithm:

Program:

```
#include <bits/stdc++.h>
using namespace std;

#define V 5

int minDistance (int dist[], bool sptSet[]) {
    int min = 9999, min_index;

    for (int v = 0; v < V; v++)
        if (sptSet(v) == false && dist[v]<=min)
            min = dist[v], min_index = v;

    return min_index;
}

void printPath (int parent[], int j) {
    if (parent[j] == -1)
        return;
    printPath (parent, parent[j]);
    cout << j << " ";
}
```

Teacher's Signature : _____

Scanned by CamScanner

Date _____

Expt. No. _____ Page No. 23

```
void printSolution(int dist[], int n, int parent[]){
    int src=0;
    Cout<<"Vertex\t Distance\t Path "<<endl;
    for (int i=1; i<V; i++) {
        Cout<<"\n"<<src<<" -> "<<i<<
        "\t "<<dist[i]<<"\t\t"<<src<<" ";
        printPath(parent, i);
    }
}

void dijkstra(int graph[V][V], int src) {
    int dist[V];
    bool sptset[V];
    int parent[V];

    for (int i=0; i<V; i++) {
        parent[0] = -1;
        dist[i] = 9999;
        sptSet[i] = false;
    }

    dist[src]=0;

    for (int Count=0; Count<V-1; Count++) {
        int u = minDistance(dist, sptSet);
        sptSet[u] = true;
```

Teacher's Signature : _____

Scanned by CamScanner

Output:

Enter the graph (Enter 99 for infinity)

0	1	5	99	99
1	0	3	99	99
5	3	0	4	99
99	99	4	0	2
99	99	99	2	0

Enter the source

0

Vertex	Distance	Path
0→1	1	0 1
0→2	4	0 1 2
0→3	8	0 1 2 3
0→4	10	0 1 4

Scanned by CamScanner

```
        for (int v=0 ; v < V ; v++)
            if (!sptSet[v] && graph[u][v] &&
                dist[u] + graph[u][v] < dist[v]){

                parent[v] = u;
                dist[v] = dist[u] + graph[u][v];
            }
        }
    printSolution(dist, V, parent);
}

int main() {
    int graph[V][V];
    Cout << "Enter the graph (Enter 99 for
            infinity): " <<endl;
    for (int i=0 ; i<V ; i++) {
        for (int j=0 ; j<V ; j++)
            Cin >> graph[i][j];
    }
    Cout << "Enter the source :" <<endl;
    int src;
    Cin >> src;

    dijkstra(graph, src);
    Cout << endl;
    return 0;
}
```

Teacher's Signature : _____

Scanned by CamScanner

Date 03/1/22

Expt. No. 10 Page No. 25

Question: Write a program for congestion control using Leaky bucket algorithm.

Aim: Implement Leaky bucket Algorithm..

Program:

```
#include <bits/stdc++.h>
# include< unistd.h>
using namespace std;
# define bucketSize 500

void bucketInput (int a, int b) {
    if ( a> bucketSize)
        Cout << "\n\t\t Bucket Overflow";
    else {
        sleep(5);
        while (a>b) {
            Cout << "\n\t\t " << b<<"
            bytes outputted. ";
            a - = b;
            sleep (5);
        }
    if (a>0)
        cout <<" \n\t \t Last " <<a <<
    " bytes set \t";
        Cout <<" \n\t \t Bucket output
            success full "; }
}
```

Teacher's Signature : _____

Scanned by CamScanner

Output :

Enter Output rate : 100

packet no 1 : Packet size = 267
100 bytes outputted
100 bytes outputted
Last 67 bytes sent.
Bucket Output Successful

Packet no 2 : Packet size = 600
Bucket Overflow

Packet no 3 : Packet Size = 324
100 bytes outputted
100 bytes outputted
100 bytes outputted
24 bytes sent
Bucket Output Successful.

Packet no 4 : Packet size = 658
Bucket Overflow

Packet no 5 : Packet size = 664
Bucket Size Overflow.

Scanned by CamScanner

Date _____

Expt. No. _____ Page No. 26

```
int main() {
    int op, pktSize;
    cout << "Enter output rate: ";
    cin >> op;
    for (int i=1; i<=5; i++) {
        sleep(rand() % 10);
        pktSize = rand() % 700;
        cout << "\n Packet no " << i <<
        "\t Packet size = " << pktSize;
        bucketInput(pktSize, op);
    }
    cout << endl;
    return 0;
}
```

Teacher's Signature : _____

Scanned by CamScanner

Output:

(a) Server Window:

```
$ python3 server.py
The server is ready to receive
```

(b) Client Window:

```
$ python3 client.py

Enter file name:
demo.txt

From Server:
hello this is demo file
```

Scanned by CamScanner

Date 03/1/22

Expt. No. 11 Page No. 27

Question: Using TCP/IP sockets, write a client-server program to make client sending the filename and the server to send back the contents of the requested file if present.

Aim: Execute Client Server Program.

Program:

Client.py

```
from socket import *
serverName = "127.0.0.1"
serverPort = 12000
clientSocket = socket(AF_INET, SOCK_STREAM)
clientSocket.connect(serverName, serverPort))
sentence = input("Enter file name")
clientSocket.send(sentence.encode())
fileContents = clientSocket.recv(1024).decode()
print('From Server :', fileContents)
clientSocket.close()
```

Teacher's Signature :

Scanned by CamScanner

Expt. No. _____ Date _____

Page No. 28

```
# Server.py

from socket import *
serverName = "127.0.0.1"
serverPort = 12000
serverSocket = socket(AF_INET, SOCK_STREAM)
serverSocket.bind((serverName, serverPort))
serverSocket.listen(1)
print("The server is ready to receive")
while 1:
    connectionSocket, addr = serverSocket.accept()
    sentence = connectionSocket.recv(1024).
                                        decode()
    file = open(sentence, "r")
    l = file.read(1024)
    connectionSocket.send(l.encode())
    file.close()
    connectionSocket.close()
```

Teacher's Signature : _____

Scanned by CamScanner

Date 03/1/22

Expt. No. 12 Page No. 29

Program: Using UDP sockets, write a client-server program to make client sending the filename and the server to send back the contents of the requested file if present.

Aim: Execute client, server program using UDP sockets.

Program Code:

```
#Client.py

from socket import *
serverName = "127.0.0.1"
serverPort = 12000
clientSocket = socket(AF_INET, SOCK_DGRAM)
sentence = input ("Enter file name")
clientSocket.sendto(bytes(sentence,"utf-8"),
    (serverName, serverPort))
fileContents, serverAddress = clientSocket.recvfrom
                              (2048)
print ('From Server:', fileContents)
clientSocket.close()
```

Teacher's Signature : _____

Scanned by CamScanner

Output:

(a) Sever Window:

```
$ python3 sever.py
The Sever is ready to receive..
```

(b) Client window:

```
$ python3 client.py
Enter filename
demo.txt
From Sever:
hello this is demo file.
```

Scanned by CamScanner

Expt. No. _____ Date _____

Page No. 30

```
# Server.py

from socket import *
serverPort = 12000
serverSocket = socket(AF_INET, SOCK_DGRAM))
serverSocket.bind (("127.0.0.1", serverPort))
print ("The Server is ready to receive")
while 1 :
    sentence, clientAddress =
            server_Socket.recvfrom (2048)
    file = open (sentence, "r")
    l = file.read (2048)
    serverSocket.sendto( bytes(l, "utf-8"),
                clientAddress)
    print ("sent back to client", l)
    file.close()
```

Teacher's Signature : _____

Scanned by CamScanner