

Name: Ekata Rajesh Chaudhari

Class: TE **Section:** A **Roll No:** 35

Experiment: 01

Title: Implementation of Bit Stuffing & De-stuffing in Framing.

```
// Program

#include<stdio.h>
#include<stdlib.h>
#include<string.h> // Include string.h for strlen function
#define MAXSIZE 100
int main() {
char *p, *q;
char in[MAXSIZE];
char stuff[MAXSIZE];
char destuff[MAXSIZE];
int count = 0;
printf("Enter the input character string (0's & 1's only) = ");
scanf("%s", in); // Input string from user
p = in;
q = stuff;

    // Bit Stuffing
while (*p != '\0') {
*q = *p; // Copy the current bit to output
q++;
if (*p == '1')
{
count++; // Count consecutive '1's
if (count == 5)
{
*q = '0'; // Stuff '0' after 5 consecutive '1's
q++;
count = 0; // Reset count
}
} else {
count = 0; // Reset count for '0's
}
p++;
}

*q = '\0'; // Add null terminator to end of stuffed string
printf("\nThe stuffed character string is = %s\n",
stuff);
```

```

    p = stuff;
q = destuff;

    // Bit Destuffing
while (*p != '\0') {
    *q = *p; // Copy the current bit to output
q++;
    if (*p == '1') {
        count++; // Count consecutive '1's
        if (count == 5) {
            p++; // Skip the stuffed '0'
            count = 0; // Reset count
        }
    }
    else {
        count = 0; // Reset count for '0's
    }
    p++;
}
*q = '\0'; // Add null terminator to end of destuffed string
printf("The destuffed character string is = %s\n", destuff);

    return 0;
}

/*
Output

Enter the input character string (0's & 1's only) = 00111111110000
The stuffed character string is = 001111101110000
The destuffed character string is = 00111111110000

*/

```

Name: Ekata Rajesh Chaudhari

Class: TE Section: A Roll No: 35

Experiment: 02

Title: Implementation of RLE data compression algorithm.

```
// Program
#include <stdio.h>
#include <string.h>
int main() {
    int i, j, cnt, l, count[50] = {0};
    char str[50];
    printf("Enter the string: ");
    scanf("%s", str);
    printf("Original String is: %s", str);
    printf("Encoded String is: ");
    l = strlen(str);
    for (i = 0; i < l; i += 1)
    {
        j = 0;
        count[i] = 1;
        do{
            j++;
            if (str[i + j] == str[i])
                count[i]++;
        }while (str[i + j] == str[i]);

        if (count[i] == 1)
            printf("%c", str[i++]);
        else{
            printf("%d%c", count[i], str[i]);
            i += count[i]; // i=i+count[i];
        }
    }
    return (0);
}
```

```
/*
Output
Enter the string: AAABBC
Original String is: AAABBC
Encoded String is: 3A2BC
*/
```

Name: Ekata Rajesh Chaudhari

Class: TE Section: A Roll No: 35

Experiment: 03

Title: Implementation of XOR Symmetric Cryptographic Algorithm

```
// Program import
java.io.*; import
java.lang.String; public
class Xor4
{ public static void main(String
args[])
{
String Msg = "CN-PRACTICAL";
char Key='$'; String CTxt="";
String DTxt=""; int xor,xor1;
char temp,temp1; for(int
i=0;i<Msg.length();i++ )
{ xor=Msg.charAt(i) ^
Key; temp =(char)xor ;
CTxt=CTxt + temp;
}
System.out.println("Encryption and Decryption using XOR Cryptogrphahy");
System.out.println("Original Message is:" +Msg);
System.out.println("Encrypted Message is:" +CTxt);
for(int i=0;i<CTxt.length();i++)
{ xor1=CTxt.charAt(i)^
Key; temp1 =(char) xor1;
DTxt=DTxt+ temp1;
}
System.out.println("Decrypted Message is:"+DTxt);
}
}

/*
Output
Encryption and Decryption using XOR Cryptogrphahy
Original Message is:CN-PRACTICAL
Encrypted Message is:gj tvegpmgeh
Decrypted Message is:CN-PRACTICAL
*/
```

Name: Ekata Rajesh Chaudhari

Class: TE Section: A Roll No: 35

Experiment: 04

Title: Implementation of RSA Asymmetric Cryptographic Algorithm

```
// Program
#include <stdio.h>
#include <math.h>
int main() {
int p, q, n, phi, d, e, i, j, CT, PT;
printf("\nEnter First Primeno");
scanf("%d", &p);
printf("\nEnter Second Primeno");
scanf("%d", &q);
n = p * q;
phi = (p - 1) * (q - 1);
printf("\nchose e such that it is relatively prime to\t %d", phi);
scanf("%d", &e);
for (d = 1; d < phi; d++)
{ if (((d * e) % phi) == 1)
break;
}
printf("\n Enter the plain text char");
scanf("%d", &PT);
CT = 1;
for (i = 0; i < e; i++)
CT = CT * PT % n;
printf("\n After Encryption %d", CT);
PT = 1;
for (i = 0; i < d; i++)
PT = PT * CT % n;
printf("\nAfter Decreption %d\n", PT);
return 0;
}
/*
Output
Enter First Primeno 1111
Enter Second Primeno 77
chose e such that it is relatively prime to 60
5959
Enter the plain text char 55
After Encryption 31
After Decreption 5
*/
```

Name: Ekata Rajesh Chaudhari

Class: TE **Section:** A **Roll No:** 35

Experiment: 05

Title: Implementation of TCP Socket (TCP Server and TCP Client)

```
// TCP SERVER
import java.io.*;
import java.net.*;
class TCPServer
{
    public static final int PORT = 1222;
    public static void main(String a[]) throws
    Exception
    {
        ServerSocket s = new ServerSocket(PORT);
        System.out.println("Server Started");
        Socket c = s.accept();
        System.out.println("Connection accepted:" + c);
        BufferedReader in = new BufferedReader(new
InputStreamReader(c.getInputStream()));
        PrintWriter out = new PrintWriter(new BufferedWriter(new
OutputStreamWriter(c.getOutputStream())), true);
        String send;        while (true)
        {
            String str = in.readLine();
            send = ("Echo:" + str);
            out.println(send);
            System.out.println("Echoing: " + str);
            if (str.equals("END"))                break;
        }
        System.out.println("Closing server");
        c.close();
        s.close();
    }
}
/*
Output
Server Started
Connection accepted:Socket[addr=/127.0.0.1,port=64594,localport=1222]

Echoing: Welcome
Echoing: Client
*/
```

```

// TCP CLIENT
import java.io.*;
import java.net.*;
class TCPClient
{ public
    static void main(String a[]) throws Exception
    {
        Socket cp = new Socket("localhost", 1222);
        System.out.println("Client connected" + cp);
        String s1;
        BufferedReader abc = new BufferedReader(new
InputStreamReader(cp.getInputStream()));
        BufferedReader in = new BufferedReader(new InputStreamReader(System.in));
        PrintWriter out = new PrintWriter(new BufferedWriter(new
OutputStreamWriter(cp.getOutputStream()), true));
        while (true)
        {
            System.out.println("Enter the string to sent to server");
s1 = in.readLine();          out.println(s1);
            String str = abc.readLine();
            System.out.println("From server:" + str);
            if (str.equals("END"))          break;
        }
        System.out.println("Closing client");
        cp.close();
    }
}

/*
Output
Client connectedSocket[addr=localhost/127.0.0.1,port=1222,localport=64594]
Enter the string to sent to server

Welcome
From server:Echo:Welcome
Enter the string to sent to server
Client
From server:Echo:Client
Enter the string to sent to server
*/

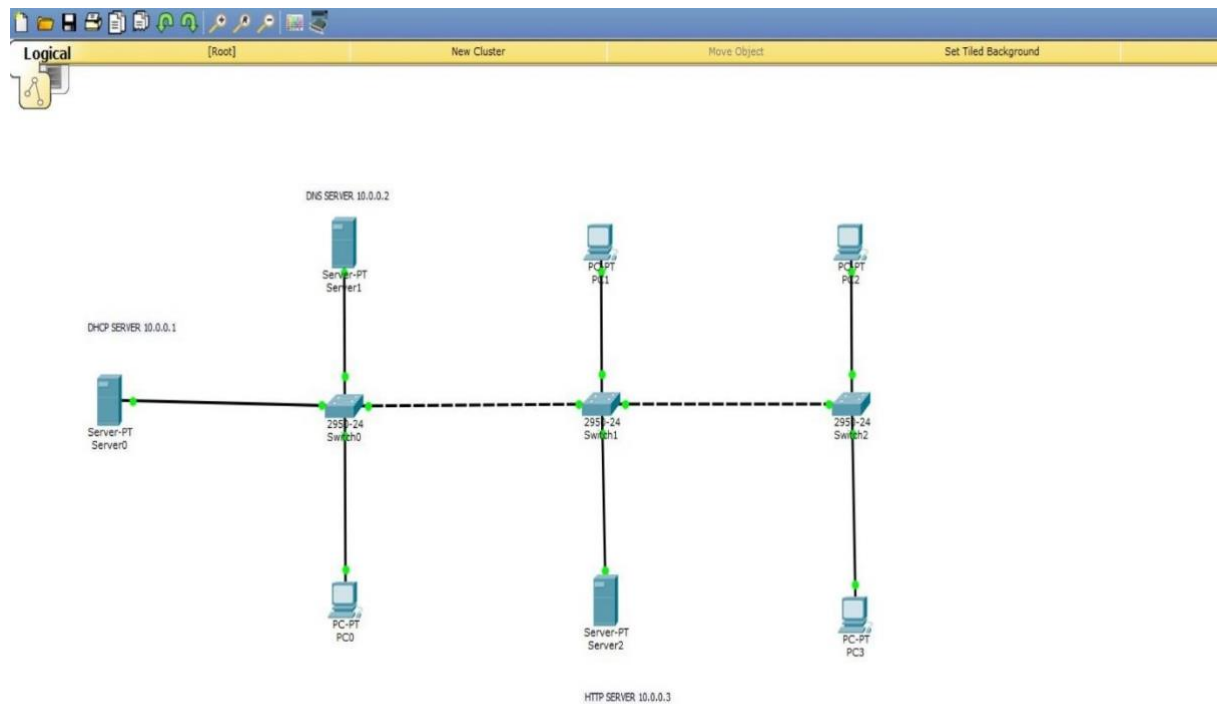
```

Name: Ekata Rajesh Chaudhari

Class: TE Section: A Roll No: 35

Experiment: 07

Title: Configuring DHCP, DNS and HTTP Server in Cisco Packet Tracer



Name: Ekata Rajesh Chaudhari

Class: TE Section: A Roll No: 35

Experiment: 08

Title: Simulation of TCP in Cisco Packet Tracer.

