

CSE62 COMPUTER NETWORKS LAB MANUAL

CONTENTS

Program #	Program statement
1	Write a program for error detecting code-using CRC-CCITT (16-bits).
2	Write a program to implement Go-Back N and Selective repeat sliding window protocol.
3	Write a program for implementation of stop and wait.
4	Using TCP/IP Sockets, write a client-server program to make client sending the file name and the server to send back the contents of the requested file if present.
5	Design, develop, and execute a program in C under UNIX / LINUX environment to implement a simple echo server and demonstrate its working. Both the server and client are to be connectionless and use UDP. The system works as follows: Client reads a line from the standard input and writes the line to the server; the server reads a line from its network input and echoes the line back to the client; the client reads the echoed line and prints it on its standard output.
6	Write a program for Congestion control using the leaky bucket algorithm
7	Write a program for Distance Vector Algorithm to find suitable path for transmission.
8	Write a program for Link State Algorithm to find suitable path for transmission.
9	Write a program for encryption and decryption using RSA algorithm.
10	Write a program to implement Diffie Hellman Key exchange.
11	a)Simulate Capturing and analysing Ethernet frames. b)Simulate capturing a bulk TCP transfer from your computer to a remote server.
12	Simulate i) Analysis of ICMP and PING messages ii) Analysis of ICMP and Traceroute
	Sample VIVA Questions

7. Write a program for Distance Vector Algorithm to find suitable path for transmission.

```
#include<stdio.h>
#include<string.h>
```

```

int main()
{
    int copy[20][20], dist[20][20], via[20][20];
    int n=0, i=0, j=0, k=0, count=0;

    system("clear");

    printf("Enter The Number Of Nodes\n");
    scanf("%d",&n);

    printf("Enter The Cost Matrix\n");

    for(i=1;i<=n;i++)

    for(j=1;j<=n;j++)
    {
        scanf("%d",&dist[i][j]);

        dist[i][i]=0;
        copy[i][j]=dist[i][j];
        via[i][j]=j;
    }

    do
    {
        count=0;

        for(i=1;i<=n;i++)

        for(j=1;j<=n;j++)

        for(k=1;k<=n;k++)

        if(copy[i][j]>(dist[i][k]+copy[k][j]))
        {
            copy[i][j]=dist[i][k]+copy[k][j];
            via[i][j]=k;
            count++;
        }
    } while (count!=0);

    for(i=1;i<=n;i++)
    {
        printf("Shortest path from Router %d \n",i);

        for(j=1;j<=n;j++)
            printf("\t\t To %d Via %d Is %d\n", j, via[i][j], copy[i][j]);
    }
}

```

```
}  
}
```

Output:

```
Enter The Number Of Nodes  
3  
Enter The Cost Matrix  
0 1 9  
2 0 8  
5 0 2  
Shortest path from Router 1  
    To 1 Via 1 Is 0  
    To 2 Via 2 Is 1  
    To 3 Via 3 Is 9  
Shortest path from Router 2  
    To 1 Via 1 Is 2  
    To 2 Via 2 Is 0  
    To 3 Via 3 Is 8  
Shortest path from Router 3  
    To 1 Via 2 Is 2  
    To 2 Via 2 Is 0  
    To 3 Via 3 Is 0
```

8. Write a program for Link State Algorithm to find suitable path for transmission.

```
#include <stdio.h>  
#define infinity 999  
  
void dij(int n,int v,int cost[10][10],int dist[])  
{  
    int i,u,count,w,flag[10],min;  
    for(i=1;i<=n;i++)  
        flag[i]=0,dist[i]=cost[v][i];  
  
    count=2;  
  
    while(count<=n)  
    {  
        min=99;  
  
        for(w=1;w<=n;w++)  
            if(dist[w]<min && !flag[w])  
                min=dist[w],u=w;  
  
        flag[u]=1;  
        count++;  
  
        for(w=1;w<=n;w++)
```

```

        if((dist[u]+cost[u][w]<dist[w]) && !flag[w])
dist[w]=dist[u]+cost[u][w];
    }
}

void main()
{
int n,v,i,j,cost[10][10],dist[10];

printf("\n Enter the number of nodes:");
scanf("%d",&n);

printf("\n Enter the cost matrix:\n");
for(i=1;i<=n;i++)
    for(j=1;j<=n;j++)
    {
scanf("%d",&cost[i][j]);
        if(cost[i][j]==0)
            cost[i][j]=infinity;
    }

printf("\n Enter the source matrix:");
scanf("%d",&v);
dij(n,v,cost,dist);
printf("\n Shortest path:\n");
for(i=1;i<=n;i++)
    if(i!=v)
printf("%d->%d,cost=%d \n",v,i,dist[i]);
}

```

Output:

```

Enter the number of nodes: 3
3

Enter the cost matrix:
0 1 7
3 0 5
6 2 0
Enter the source matrix: 7
7

Shortest path:
7->1,cost=0
7->2,cost=0
7->3,cost=0

```

9. Write a program for encryption and decryption using RSA algorithm.

```
#include<stdio.h>
#include<math.h>

int gcd(int a, int h)
{
    int temp;
    while(1)
    {
        temp = a%h;
        if(temp==0)
            return h;
        a = h;
        h = temp;
    }
}

int main()
{
    double p = 3;
    double q = 7;
    double n=p*q;
    double count;
    double totient = (p-1)*(q-1);

    // (public key) e stands for encrypt
    double e=2;

    while(e<totient){
        count = gcd(e,totient);
        if(count==1)
            break;
        else
            e++;
    }

    //(private key) d stands for decrypt
    double d;
    double k = 2;

    d = (1 + (k*totient))/e;
    double msg = 12;
    double c = pow(msg,e);
    double m = pow(c,d);
    c=fmod(c,n);
    m=fmod(m,n);
}
```

```

printf("Message data = %lf",msg);
printf("\np = %lf",p);
printf("\nq = %lf",q);
printf("\nn = pq = %lf",n);
printf("\ntotient = %lf",totient);
printf("\ne = %lf",e);
printf("\nd = %lf",d);
printf("\nEncrypted data = %lf",c);
printf("\nOriginal Message Sent = %lf",m);

return 0;
}

```

Output:

```

/tmp/FoYtBgRvVA.o
Message data = 12.000000
p = 3.000000
q = 7.000000
n = pq = 21.000000
totient = 12.000000
e = 5.000000
d = 5.000000
Encrypted data = 3.000000
Original Message Sent = 12.000000

```

10. Write a program to implement Diffie Hellman Key exchange.

```

#include <stdio.h>
#include <math.h>

```

```

long int power(long int a, long int b, long int P)
{
    if(b==1)
        return a;
    else
    {
        double d = (((long int) pow(a, b)) % P);
        return d;
    }
}

```

```

    }
}

int main()
{
    long int P, G, x, a, y, b, ka, kb;

    P = 31;
    printf("The value of P : %ld \n", P);
    G = 3;
    printf("The value of G : %ld \n\n", G);

    a = 11;
    printf("The private key a for Alice : %ld \n", a);
    x = power(G, a, P);

    b = 7;
    printf("The private key b for Bob : %ld \n\n", b);
    y = power(G, b, P);

    ka = power(y, a, P);
    kb = power(x, b, P);

    printf("Secret key for the Alice is : %ld \n", ka);
    printf("Secret Key for the Bob is : %ld \n", kb);

    return 0;
}

```

Output:

```
The value of P : 31
The value of G : 3

The private key a for Alice : 11
The private key b for Bob : 7

Secret key for the Alice is : 22
Secret Key for the Bob is : 22
```

Include Programs – 11 and 12 later