WEEK - 1	Exp. No.: Date: -23

AIM: C program to convert Binary to Gray Code.

Program:

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

int main() {
    char bin[32], gray[32];
    printf("Enter a Binary code: ");
    int s = scanf("%s", bin);

    gray[0] = bin[0];
    for (int i = 1; bin[i]; i++)
        gray[i] = (bin[i] == bin[i - 1]) ? '0' : '1';

    gray[strlen(bin)] = '\0';

    printf("Gray code equivalent: %s\n", gray);
    return 0;
}
```

Output:

```
~/CN-Lab-217y1a05c0/Lab-1$ vi BintoGray.c
~/CN-Lab-217y1a05c0/Lab-1$ cc BintoGray.c
~/CN-Lab-217y1a05c0/Lab-1$ ./a.out
Enter a Binary code: 1011
Gray code equivalent: 1110
~/CN-Lab-217y1a05c0/Lab-1$
```

WEEK - 1	Exp. No.: Date: -23

AIM: Write a C program to convert Gray code to Binary.

Program:

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

void main() {
    char gray[32], bin[32];
    printf("Enter a Gray code: ");
    int sc = scanf("%s", gray);
    int len = strlen(gray),i;

    bin[0] = gray[0];
    for (i = 0; i < len - 1; i++) {
        bin[i + 1] = (gray[i + 1] == '0') ? bin[i] : (bin[i] == '0') ? '1' :
'0';
    }
    bin[i + 1] = '\0';

    printf("Binary equivalent: %s\n", bin);
}
```

Output:

```
~/CN-Lab-217y1a05c0/Lab-1$ cc GraytoBin.c
~/CN-Lab-217y1a05c0/Lab-1$ ./a.out
Enter a Gray code: 1110
Binary equivalent: 1011
~/CN-Lab-217y1a05c0/Lab-1$
```

WEEK - 2	Exp. No.: Date: -23

AIM: Write a C program for Error Detection and correction.

Program:

```
#include<stdio.h>
#include<stdlib.h>
int parity=0;
void createmsg(int **msg,int *n){
    free(*msg);
    printf("Enter size of msg block: ");
    int sc = scanf("%d",n);
    (*n)++;
    *msg = (int*)malloc(*n * sizeof(int));
}
void Generate(int *msg,int n){
    printf("Enter data bits: ");
    msg[0]=0;
    for (int i = 0; i < n; i++) {
        if (i & (i - 1)) {
            int sc = scanf("%d",&msg[i]);
            if (msg[i]==1) { parity ^=(i); }
        }
    }
    printf("Message   block: ");
    for (int i = 1; i < n; i++) {
        if (!(i & (i - 1))){
            msg[i]=parity & 1;
            parity >>= 1;
        }
        printf("%d ",msg[i]);
    }
}
void Correction(int *msg,int n){
    printf("Enter senders msg block: ");
    for (int i = 1; i < n; i++) {
        int sc = scanf("%d",&msg[i]);
        if (msg[i]==1) { parity ^=(i); }
    }
    if (parity==0) {
        printf("\nReceiver said \"No Unibit error :)\"");
    }else {
        printf("\nReceiver said \"Error at position: %d!\"",parity);
        msg[parity]^=1; /*corrects error 0-->1 : 1-->0 */
        printf("\nAfter Correction: ");
```

WEEK - 2	Exp. No.: Date: -23

```
        printf("\nAfter Correction: ");
        for (int i = 1; i < n; i++) {
            printf("%d ",msg[i]);
        }
    }
}
    /**
    * This program reads a non-negative integer to specify size of Message
block and data bits from user input, outputs a n-bit hamming code, then reads a
sender's message block from user input and checks for uni-bit errors.
     Example Input:
     input:(7,4)       1   0 1 0
     output:(7,4) 1 0 1 1 0 1 0
     input:(15,11)       1   0 1 0   0 1 0 1 0 0 1
     output:(15,11) 0 0 1 1 0 1 0 1 0 1 0 1 0 0 1
     Receiver said "No Unibit error :)"
    */
void main() {
    printf("1. Generation\n2. Error Detection & Correction\n3. Exit\n");
    int ch, n, *msg;
        createmsg(&msg,&n);
    while(1){
        parity=0;
        printf("\nEnter choice: ");
        int sc = scanf("%d",&ch);
        switch (ch)
        {
        case 1: /*Code Generation */
            Generate(msg,n);
            break;
        case 2: /* Error detection & Correction*/
            Correction(msg,n);
            break;
        case 3: exit(0);
        default:printf("invalid choice!! Try again.");
            break;
        }
        free(msg);
    }
}
```

WEEK - 2	Exp. No.: Date: -23

Output:

```
~/CN-Lab-217y1a05c0/Lab-2$ vi Hamming.c
~/CN-Lab-217y1a05c0/Lab-2$ cc Hamming.c
~/CN-Lab-217y1a05c0/Lab-2$ ./a.out
1. Generation
2. Error Detection & Correction
3. Exit
Enter size of msg block: 7

Enter choice: 1
Enter data bits:     1   0 1 0
Message   block: 1 0 1 1 0 1 0
Enter choice: 2
Enter senders msg block: 1 0 1 0 0 1 0

Receiver said "Error at position: 4!"
After Correction: 1 0 1 1 0 1 0
Enter choice: 2
Enter senders msg block: 1 0 1 1 0 1 0

Receiver said "No Unibit error :)"
Enter choice: 3
~/CN-Lab-217y1a05c0/Lab-2$
```

WEEK - 3	Exp. No.: Date: -23

AIM: Write a C program to implement Bit Stuffing.

Program:

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

int main() {
    int *num,*cp,i,j,k,count,n;
    printf("Enter the number of bits: ");
    scanf("%d", &n);
    num = (int*)malloc(n * sizeof(int));
    cp = (int*)malloc(n * sizeof(int));
    printf("Enter sending bits: ");
    for(i = 0; i < n; ++i) {
        scanf("%d",&num[i]);
    }
    for ( i = 0, j = 0; i < n; i++, j++) {
        cp[j]=num[i];
        if(num[i]==1) {
            for(k = i + 1, count = 1; num[k] == 1 && k < n && count < 5;
k++,i++) {
                cp[++j] = num[k];
                if(++count == 5) {
                    cp[++j] = 0;
                }
            }
        }
    }
    printf("After Bit Stuffing: ");
    for(i=0; i<j; i++) {printf("%d ",cp[i]);}
    printf("\n");
    free(num);
    return 0;
}
```

WEEK - 3	Exp. No.: Date: -23

Output:

```
~/CN-Lab-217y1a05c0/Lab-3$ vi Bitstuffing.c
~/CN-Lab-217y1a05c0/Lab-3$ cc Bitstuffing.c -w
~/CN-Lab-217y1a05c0/Lab-3$ ./a.out
Enter the number of bits: 16
Enter sending bits: 1 1 0 1 0 1 1 1 1 1 0 1 0 1 1 1
After Bit Stuffing: 1 1 0 1 0 1 1 1 1 1 0 0 1 0 1 1 1
~/CN-Lab-217y1a05c0/Lab-3$
```

WEEK - 4	Exp. No.: Date: -23

AIM: Write a C program to implement FCFS.

Program:

```c
#include <stdio.h>
#include <string.h>
char data[20][20];
int n;
void main()
{
    int i, ch, j;
    char tmp[20][20];
    printf("Enter the number of frames:");
    scanf("%d", &n);
    for (i = 0; i <= n; i++) {
        if (i != 0) {
            printf("Frame%d: ", i);
            scanf("%s", data[i]);
        }
    }
    for (i = 0; i <= n; i++) {
        tmp[i][0] = 49 + strlen(data[i]);
        tmp[i][1] = '\0';
        strcat(tmp[i], data[i]);
    }
    printf("\n\t\t AT THE SENDER:\n");
    printf("Data as frames:\n");
    for (i = 1; i <= n; i++) {
        printf("frame%d: ", i);
        puts(tmp[i]);
    }
    printf("Data transmitted: ");
    for (i = 1; i <= n; i++)
        printf("%s", tmp[i]);
    printf("\n\t\t AT THE RECEIVER\n");
    printf("Data received after removing count char: ");
    for (i = 1; i <= n; i++) {
        ch = (int)(tmp[i][0] - 49);
        for (j = 1; j <= ch; j++)
            data[i][j - 1] = tmp[i][j];
        data[i][j - 1] = '\0';
    }
```

WEEK - 4	Exp. No.: Date: -23

```
    for (i = 1; i <= n; i++)
        printf("%s", data[i]);
    printf("\nData as frames:\n");
    for (i = 1; i <= n; i++)
    {
        printf("Frame%d: ", i);
        puts(data[i]);
    }
}
```

Output:

```
~/CN-Lab-217y1a05c0/Lab-4$ vi Charcount.c
~/CN-Lab-217y1a05c0/Lab-4$ cc Charcount.c -w
~/CN-Lab-217y1a05c0/Lab-4$ ./a.out
Enter the number of frames:3
Frame1: Srinivas
Frame2: Rao
Frame3: Tammireddy

         AT THE SENDER:
Data as frames:
frame1: 9Srinivas
frame2: 4Rao
frame3: ;Tammireddy
Data transmitted: 9Srinivas4Rao;Tammireddy
         AT THE RECEIVER
Data received after removing count char: SrinivasRaoTammireddy
Data as frames:
Frame1: Srinivas
Frame2: Rao
Frame3: Tammireddy
~/CN-Lab-217y1a05c0/Lab-4$
```

WEEK - 4	Exp. No.: Date: -23

AIM: Write a C program to implement Character Stuffing.

Program:

```
#include <stdio.h>
#include <string.h>
#include <stdlib.h>
void charc();
void main() {
    int choice;
    while (1)
    {
        printf("\n 1.Character stuffing");
        printf("\n 2.Exit");
        printf("\n Enter choice: ");
        scanf("%d", &choice);
        if (choice > 2)
            printf("\n invalid option...please re-enter");
        switch (choice)
        {
        case 1:
            charc();
            break;
        case 2: exit(0);
        }
    }
}
void charc(void)
{
    char c[50], d[50], t[50];
    int i, m, j;
    printf("Enter the number of characters:\n");
    scanf("%d", &m);
    printf("Enter the characters:\n");
    for (i = 0; i < m + 1; i++) {
        scanf("%c", &c[i]);
    }
    printf("\nOriginal data: ");
    for (i = 0; i < m + 1; i++)
        printf("%c", c[i]);
    strcpy(d, "dlexstx");
    for (i = 0, j = 6; i < m + 1; i++, j++) {
        if ((c[i] == 'd' && c[i + 1] == 'l' && c[i + 2] == 'e'))
```

WEEK - 4	Exp. No.: Date: -23

```
        {
            d[j++] = 'd';
            d[j++] = 'l';
            d[j++] = 'e';
            m += 3;
        }
        d[j] = c[i];
    }
    m = m + 7;
    d[m++] = 'd';
    d[m++] = 'l';
    d[m++] = 'e';
    d[m++] = 's';
    d[m++] = 't';
    d[m++] = 'x';
    printf("\n\nTransmitted data:\n");
    for (i = 0; i < m; i++) {
        printf("%c", d[i]);
    }
    for (i = 6, j = 0; i < m - 6; i++, j++) {
        if (d[i] == 'd' && d[i + 1] == 'l' && d[i + 2] == 'e' && d[i + 3] ==
'd' && d[i + 4] == 'l' && d[i + 5] == 'e')
            i = i + 3;
        t[j] = d[i];
    }
    printf("\n\nReceived data: ");
    for (i = 0; i < j; i++)
    {
        printf("%c", t[i]);
    }
}
```

WEEK - 4	Exp. No.: Date: -23

Output:

```
~/CN-Lab-217y1a05c0/Lab-4$ vi Charstuffing.c
~/CN-Lab-217y1a05c0/Lab-4$ cc Charstuffing.c -w
~/CN-Lab-217y1a05c0/Lab-4$ ./a.out

 1.Character stuffing
 2.Exit
 Enter choice: 1
Enter the number of characters:
8
Enter the characters:
Srinivas

Original data:
Srinivas

Transmitted data:
dlestx
Srinivasdlestx

Received data:
Srinivas
 1.Character stuffing
 2.Exit
 Enter choice: 2
~/CN-Lab-217y1a05c0/Lab-4$
```

WEEK - 5	Exp. No.: Date: -23

AIM: Write a C program to implement C.R.C.

Program:

```
#include<stdio.h>
#include<stdlib.h>
#include<string.h>
void get_crc(char *msg, char *div,int *m,int *n) {
    *m = strlen(msg);
    *n = strlen(div);
    printf("Message is: %s",msg);
    for (int i = 0; i < *n - 1; i++) { msg[*m + i] = '0';}
    for (int i = 0; i < *m; i++) {
        if (msg[i] == '1') {
            for (int j = 0; j < *n; j++) {
            msg[i + j] = (msg[i + j] == div[j]) ? '0' : '1';
            }
        }
    }
    return ;
}
void main(){
    char msg[100],div[10];
    int n=0,m=0,ch,i,j;
    printf("1.Generate\n2.Detect\n3.Exit");
    while(1){
        printf("\nEnter choice:");
        scanf("%d",&ch);
        switch(ch){
            case 1:/*Generation*/;
            printf("Enter CRC code: ");
            scanf("%s",msg);
            printf("Enter equation Coefficient:");
            scanf("%s",div);
            get_crc(msg,div,&m,&n);
            printf(" ");
            for (i = m; i < m + n - 1; i++) {
                printf("%c",msg[i]);
            }break;
            case 2:
            printf("Enter Sender's CRC code: ");
            scanf("%s",msg);
            get_crc(msg,div,&m,&n);
```

WEEK - 5	Exp. No.: Date: -23

```
            for (i = m; i < m + n - 1; i++) {
                if (msg[i] == '1') {
                    printf("\nError detected!");
                    break;
                }
            }
            printf("\nNo error detected.");
            break;
            case 3:exit(0);
            default: printf("Invalid Choice! Try again.");
        }
    }
}
```

Output:

```
~/CN-Lab-217y1a05c0/Lab-5$ cc CRC.c -w
~/CN-Lab-217y1a05c0/Lab-5$ ./a.out
1.Generate
2.Detect
3.Exit
Enter choice:1
Enter CRC code: 101110101
Enter equation Coefficient:101
Message is: 101110101 11
Enter choice:2
Enter Sender's CRC code: 10111010111
Message is: 10111010111
No error detected.
Enter choice:3
~/CN-Lab-217y1a05c0/Lab-5$
```

WEEK - 6	Exp. No.: Date: -23

AIM: Write a C program to implement Dijkstra's Algorithm.

Program:

```
#include <stdio.h>
#define INFINITY 9999
#define MAX 10
void dijkstra(int G[MAX][MAX], int n, int startnode);
int main()
{
    int G[MAX][MAX], i, j, n, u;
    printf("Enter no. of vertices:");
    scanf("%d", &n);
    printf("\nEnter the adjacency matrix:\n");
    for (i = 0; i < n; i++)
    {
        for (j = 0; j < n; j++)
        {
            scanf("%d", &G[i][j]);
        }
    }
    printf("\nEnter the starting node:");
    scanf("%d", &u);
    dijkstra(G, n, u);
    return 0;
}
void dijkstra(int G[MAX][MAX], int n, int startnode)
{
    int cost[MAX][MAX], distance[MAX], pred[MAX];
    int visited[MAX], count, mindistance, nextnode, i, j;
    for (i = 0; i < n; i++) {
        for (j = 0; j < n; j++) {
            if (G[i][j] == 0)
            {
                cost[i][j] = INFINITY;
            }
            else
            {
                cost[i][j] = G[i][j];
            }
        }
     }
```

WEEK - 6	Exp. No.: Date: -23

```
for (i = 0; i < n; i++)
{
    distance[i] = cost[startnode][i];
    pred[i] = startnode;
    visited[i] = 0;
}
distance[startnode] = 0;
visited[startnode] = 1;
count = 1;
while (count < n - 1)
{
    mindistance = INFINITY;
    for (i = 0; i < n; i++)
    {
        if (distance[i] < mindistance && !visited[i])
        {
            mindistance = distance[i];
            nextnode = i;
        }
        for (i = 0; i < n; i++)
        {
            if (!visited[i])
            {
                if (mindistance + cost[nextnode][i] < distance[i])
                {
                    distance[i] = mindistance + cost[nextnode][i];
                    pred[i] = nextnode;
                }
            }
        }
    }
    count++;
}
```

WEEK - 6	Exp. No.: Date: -23

```
    for (i = 0; i < n; i++)
    {

        if (i != startnode)
        {
            printf("\nDistance of node %d = %d", i, distance[i]);
            printf("\nPath = %d", i);
            j = i;
            do
            {
                j = pred[j];
                printf(" <- %d", j);
            } while (j != startnode);
        }
    }
}
```

Output:

```
~/CN-Lab-217y1a05c0/Lab-6$ cc DSP.c -w
~/CN-Lab-217y1a05c0/Lab-6$ ./a.out
Enter no. of vertices:3

Enter the adjacency matrix:
2 0 4
5 1 9
1 2 3

Enter the starting node:2

Distance of node 0 = 1
Path = 0 <- 2
Distance of node 1 = 2
~/CN-Lab-217y1a05c0/Lab-6$
```

WEEK - 6	Exp. No.: Date: -23

AIM: Write a C program to implement Go-Back-N protocol.

Program:

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

int main() {
    int frame_size;
    int N  = 10;       /* Window size */
    int Rn = 0;        /* Request number `Rn = Ack + 1`                */
    int Sn = 0;        /* `[Sb:Sm)` Sequence number  [0 1 2 3 4 5] 6  */
    int Sb = 0;        /* `Sb = 0`Sequence base      (0)1 2 3 4 5     */
    int Sm = 11;       /* `Sm = N + 1` Sequence max   0 1 2 3 4 5 (6) */
    int count = 0,n;
    printf("Enter size of frame: ");
    scanf("%d",&frame_size);
    printf("Enter sliding window size: ");
    scanf("%d",&N);
    Sm = N;
    printf("Enter which nth frame is courrupted: ");
    scanf("%d",&n);

    if(N <= 0 || frame_size < N || n < 1) {
            printf("Invalied inputs.");
            return 1;
    } printf("\nGoBack-%d ARQ:",N);

    for (int ack = -1; ack < frame_size - 1; printf(" //timeout"))
    {
        for ( Sn = Sb; Sn < Sm ; Sn++) {
            count++; /*frame sent*/
            ack = (count%n == 0)? ack : (Sn == Sb)? Sn : ack;

            printf("\n%2d. Frame %2d is send. Ack:%2d",count,Sn+1,ack+1);

            Sm = (ack + (Sm - Sb + 1) >= frame_size)? frame_size : ack + (Sm -
Sb + 1);
            Sb = ack + 1;
            if (count%n == 0)
            { printf(" (courrupted!)"); }
        }
    }printf("\n");
    return 0;
}
```

WEEK - 6	Exp. No.: Date: -23

Output:

```
~/CN-Lab-217y1a05c0/Lab-6$ vi GoBack-N.c
~/CN-Lab-217y1a05c0/Lab-6$ cc GoBack-N.c -w
~/CN-Lab-217y1a05c0/Lab-6$ ./a.out
Enter size of frame: 6
Enter sliding window size: 5
Enter which nth frame is courrupted: 4

GoBack-5 ARQ:
 1. Frame  1 is send. Ack: 1
 2. Frame  2 is send. Ack: 2
 3. Frame  3 is send. Ack: 3
 4. Frame  4 is send. Ack: 3 (courrupted!)
 5. Frame  5 is send. Ack: 3
 6. Frame  6 is send. Ack: 3 //timeout
 7. Frame  4 is send. Ack: 4
 8. Frame  5 is send. Ack: 4 (courrupted!)
 9. Frame  6 is send. Ack: 4 //timeout
10. Frame  5 is send. Ack: 5
11. Frame  6 is send. Ack: 6 //timeout
~/CN-Lab-217y1a05c0/Lab-6$
```

WEEK - 6	Exp. No.: Date: -23

AIM: Write a C program to implement Sliding Window Protocol.

Program:

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

int main(){
        int w,f,*frames;
        printf("Enter size of sliding window:");
        scanf("%d",&w);
        printf("Enter no.of frames:");
        scanf("%d",&f);
        if(w <= 0 || f <= 0){
                printf("Invalied inputs.");
                return 1;
        }
        frames = (int*)malloc(f * sizeof(int));
        printf("Enter %d frames:",f);
        for(int i = 0;i < f;i++){
                scanf("%d",&frames[i]);
        }
        printf("----\nThefollowing is by using Sliding windoew
protocol\n----\nSender: ");
        for(int i = 0;i < f;i++){
                if((i+1)%w==0){
                        printf("%d ",frames[i]);
                        printf("\nAcknowledgement recieved");
                        printf("\nSender: ");
                }else{
                        printf("%d ",frames[i]);
                }
        }
        if(f%w != 0){
                printf("\nAcknowledgement recieved\n");
        }free(frames);
        return 0;
}
```

WEEK - 6	Exp. No.: Date: -23

Output:

```
~/CN-Lab-217y1a05c0/Lab-6$ vi  Slidingwindow.c
~/CN-Lab-217y1a05c0/Lab-6$ cc  Slidingwindow.c -w
~/CN-Lab-217y1a05c0/Lab-6$ ./a.out
Enter size of sliding window:4
Enter no.of frames:6
Enter 6 frames:1 2 3 4 5 6
----
Thefollowing is by using Sliding windoew protocol
----
Sender: 1 2 3 4
Acknowledgement recieved
Sender: 5 6
Acknowledgement recieved
~/CN-Lab-217y1a05c0/Lab-6$
```

WEEK - 7	Exp. No.: Date: -23

AIM: Write a C program to find a Broadcast Tree.

Program:

```
#include<stdio.h>
struct edge{
    int s,d,t; /*s:source, d:destination, t:traffic/travel-time*/
};
int main(){
    int n, graph[10][10], node[10] = {0,0,0,0,0,0,0,0,0,0};
    struct edge edgeq[100],temp; /*An Edges Queue 'edgeq' for storing
sorted-paths*/
    printf("Enter no.of nodes:");
    scanf("%d",&n);

    for (int i = 0; i<n;i++){
        for(int j = 0; j < n;j++){
            if (i<j) {
                printf("Enter traffic between (%c)--(%c): ",65+i,65+j);
                scanf("%d",&graph[i][j]);
            } else if (i==j) {
                graph[i][j] = -2;
            } else {
                graph[i][j] = graph[j][i];
            }
            edgeq[n*i+j].s = i;
            edgeq[n*i+j].d = j;
            edgeq[n*i+j].t = graph[i][j];
            /* Insertion Sort: for all the edges */
            for(int k = n*i+j; k > 0; k--){
                if(edgeq[k].t < edgeq[k-1].t){
                    temp       = edgeq[k];
                    edgeq[k]   = edgeq[k-1];
                    edgeq[k-1] = temp;
                }
            }
        }
    }
    int cost = 0;
    printf("\nBroadcast tree:\n");
    for(int i = n,count = 1; count < n , i < n*n ; i++) { /*First n edges are
'-2'*/
```

WEEK - 7	Exp. No.: Date: -23

```
        if (!(node[edgeq[i].d]) && edgeq[i].t > 0 ) {
            printf("(%c)-->(%c) = %2d\n",65 + edgeq[i].s,65 +
edgeq[i].d,edgeq[i].t);
            cost += edgeq[i].t;
            node[edgeq[i].s] = 1;
            node[edgeq[i].d] = 1;
            count++;
        }
    }
    printf("Total distance: %d",cost);
    return 0;
}
```

Output:

```
~/CN-Lab-217y1a05c0/Lab-7$ cc BroadcastTree.c -w
~/CN-Lab-217y1a05c0/Lab-7$ ./a.out
Enter no.of nodes:6
Enter traffic between (A)--(B): 1
Enter traffic between (A)--(C): -1
Enter traffic between (A)--(D): 4
Enter traffic between (A)--(E): 3
Enter traffic between (A)--(F): -1
Enter traffic between (B)--(C): -1
Enter traffic between (B)--(D): 4
Enter traffic between (B)--(E): 2
Enter traffic between (B)--(F): -1
Enter traffic between (C)--(D): -1
Enter traffic between (C)--(E): 4
Enter traffic between (C)--(F): 5
Enter traffic between (D)--(E): 4
Enter traffic between (D)--(F): -1
Enter traffic between (E)--(F): 7

Broadcast tree:
(A)-->(B) =  1
(B)-->(E) =  2
(A)-->(D) =  4
(E)-->(C) =  4
(C)-->(F) =  5
~/CN-Lab-217y1a05c0/Lab-7$
```

WEEK - 8	Exp. No.: Date: -23

AIM: Write a C program to implement Distance Vector Routing.

Program:

```
#include<stdio.h>
struct node {
        unsigned cost[20];
        unsigned from[20];
} rt[10];
int main() {
        int costMat[20][20];
        int n,i,j,k,count=0;
        printf("\nEnter the number of nodes: ");
        scanf("%d",&n);
        printf("Enter the cost matrix:\n");

        for(i=0;i<n;i++){
            for(j=0;j<n;j++) {
                scanf("%d",&costMat[i][j]);

                costMat[i][i] = 0;
                rt[i].cost[j] = costMat[i][j];
                rt[i].from[j] = j;
            }
        }
        do{
            count=0;
            for(i = 0; i < n; i++){
                for(j = 0; j < n; j++){
                    for(k = 0; k < n; k++){
                        if( rt[i].cost[j] > costMat[i][k] + rt[k].cost[j]) {
                            rt[i].cost[j] = rt[i].cost[k] + rt[k].cost[j];
                            rt[i].from[j] = k;
                            count++;
                        }
                    }
                }
            }
        } while(count!=0);
```

WEEK - 8	Exp. No.: Date: -23

```
        for(i=0;i<n;i++) {
            printf("\nState value for router %d is:\n",i+1);
            for(j=0;j<n;j++) {
                printf("Node %d via %d Distance:
%d\n",j+1,rt[i].from[j]+1,rt[i].cost[j]);
            }
        }
        printf("\n");
}
```

Output:

```
~/CN-Lab-217y1a05c0/Lab-8$ vi DVR.c
~/CN-Lab-217y1a05c0/Lab-8$ cc DVR.c -w
~/CN-Lab-217y1a05c0/Lab-8$ ./a.out

Enter the number of nodes: 3
Enter the cost matrix:
0 2 8
6 0 5
9 5 0

State value for router 1 is:
Node 1 via 1 Distance: 0
Node 2 via 2 Distance: 2
Node 3 via 2 Distance: 7

State value for router 2 is:
Node 1 via 1 Distance: 6
Node 2 via 2 Distance: 0
Node 3 via 3 Distance: 5

State value for router 3 is:
Node 1 via 1 Distance: 9
Node 2 via 2 Distance: 5
Node 3 via 3 Distance: 0

~/CN-Lab-217y1a05c0/Lab-8$
```

WEEK - 9	Exp. No.: Date: -23

AIM: Write a C program to implement DES Encryption and Decryption.

Program:

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

void DES_Shift(char* data, int mode) {
    int len = strlen(data), shift;
    printf("Enter the key: ");
    scanf("%d", &shift);
    shift  = (mode == 1) ? shift : 26 - shift;
    for (int i = 0; i < len; i++) {
        if (data[i] >= 'a' && data[i] <= 'z') {
            data[i] = 'a' + (data[i] - 'a' + shift) % 26;
        } else if (data[i] >= 'A' && data[i] <= 'Z') {
            data[i] = 'A' + (data[i] - 'A' + shift) % 26;
        }
    }
}
int main() {
    printf("1. Encrypt\n2. Decrypt\n3. Exit\n");
    char* data = NULL;
    int ch,size,shift;
    size_t buffer = 0;
    while (1) {
        printf("Enter your choice: ");
        scanf("%d", &ch);
        getchar();
        switch (ch) {
        case 1:
            printf("Enter the message to encrypt: ");
            getline(&data, &buffer, stdin);
            DES_Shift(data, 1);
            printf("Encrypted message: %s\n", data);
            break;
        case 2:
            printf("Enter the message to decrypt: ");
            getline(&data, &buffer, stdin);
            DES_Shift(data, 0);
            printf("Decrypted message: %s\n", data);
            break;
```

WEEK - 9	Exp. No.: Date: -23

```
        case 3:
            exit(0);
        default:
            printf("Invalid choice\n");
            break;
        } free(data);
        data = NULL;
    }
    return 0;
}
```

Output:

```
~/CN-Lab-217y1a05c0/Lab-9$ vi DESEncryption.c
~/CN-Lab-217y1a05c0/Lab-9$ cc DESEncryption.c -w
~/CN-Lab-217y1a05c0/Lab-9$ ./a.out
1. Encrypt
2. Decrypt
3. Exit
Enter your choice: 1
Enter the message to encrypt: Srinivas!
Enter the key: 3
Encrypted message: Vulqlydv!

Enter your choice: 2
Enter the message to decrypt: Vulqlydv!
Enter the key: 3
Decrypted message: Srinivas!

Enter your choice: 3
~/CN-Lab-217y1a05c0/Lab-9$
```

WEEK - 9	Exp. No.: Date: -23

AIM: Write a C program to implement Frame sorting technique using Buffer.

Program:

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

#define FRAM_TXT_SIZ 3
#define MAX_NOF_FRAM 127

char str[FRAM_TXT_SIZ * MAX_NOF_FRAM];

struct frame {
    char text[FRAM_TXT_SIZ];
    int seq_no;
} fr[MAX_NOF_FRAM], shuf_ary[MAX_NOF_FRAM];

int assign_seq_no() {
    int k = 0, i, j;
    for (i = 0; i < strlen(str); k++)
    {
        fr[k].seq_no = k;
        for (j = 0; j < FRAM_TXT_SIZ && str[i] != '\0'; j++)
            fr[k].text[j] = str[i++];
    }
    printf("\nAfter assigning sequence numbers:\n");
    for (i = 0; i < k; i++)
        printf("%d:%s ", i, fr[i].text);
    return k; // k gives no of frames
}
void generate(int *random_ary, const int limit) {
    int r, i = 0, j;
    while (i < limit)
    {
        r = rand() % limit;
        for (j = 0; j < i; j++)
            if (random_ary[j] == r)
                break;
        if (i == j)
            random_ary[i++] = r;
    }
}
```

WEEK - 9	Exp. No.: Date: -23

```
void shuffle(const int no_frames) {
    int i, k = 0;
    int* random_ary = (int*)malloc(no_frames * sizeof(int));
    generate(random_ary, no_frames);
    for (i = 0; i < no_frames; i++)
        shuf_ary[i] = fr[random_ary[i]];
    printf("\n\nAFTER SHUFFLING:\n");
    for (i = 0; i < no_frames; i++)
        printf("%d:%s ", shuf_ary[i].seq_no, shuf_ary[i].text);
    free(random_ary);
}
void sort(const int no_frames) {
    int i, j, flag = 1;
    struct frame hold;
    for (i = 0; i < no_frames - 1 && flag == 1; i++) {
        flag = 0;
        for (j = 0; j < no_frames - 1 - i; j++)
            if (shuf_ary[j].seq_no > shuf_ary[j + 1].seq_no) {
                hold = shuf_ary[j];
                shuf_ary[j] = shuf_ary[j + 1];
                shuf_ary[j + 1] = hold;
                flag = 1;
            }
    }
}
int main() {
    int no_frames, i;
    printf("Enter the message: ");
    scanf("%s", str);
    no_frames = assign_seq_no();
    shuffle(no_frames);
    sort(no_frames);
    printf("\n\nAFTER SORTING:\n");
    for (i = 0; i < no_frames; i++)
        printf("%s", shuf_ary[i].text);
    printf("\n\n");
}
```

WEEK - 9	Exp. No.: Date: -23

Output:

```
~/CN-Lab-217y1a05c0/Lab-9$ ./a.out
Enter the message: Srinivas

After assigning sequence numbers:
0:Sri 1:niv 2:as

AFTER SHUFFLING:
1:niv 0:Sri 2:as

AFTER SORTING:
Srinivas

~/CN-Lab-217y1a05c0/Lab-9$
```

WEEK - 10	Exp. No.: Date: -23

AIM: Write a C program to implement Leaky Bucket congestion control.

Program:

```
#include <stdio.h>
int main() {
    int in, out, buck_size, n, store = 0;
    printf("Enter bucket size, outgoing rate and no of inputs: ");
    scanf("%d %d %d", &buck_size, &out, &n);
    while (n != 0) {
        printf("Enter the in packet size : ");
        scanf("%d", &in);
        printf("in packet size: %d\n", in);
        if (in <= (buck_size - store)) {
            store += in;
            printf("Bucket buffer size %d out of %d\n", store, buck_size);
        } else {
            printf("Dropped %d no of packets \n", in - (buck_size - store));
            printf("Bucket buffer size %d out of: %d\n", buck_size, buck_size);
            store = buck_size;
        }
        store = store - out;
        printf("After outgoing %d packets left out of %d in buffer:\n", store,
buck_size);
        n--;
    }
}
```

Output:

```
~/CN-Lab-217y1a05c0/Lab-10$ vi Leakybucket.c
~/CN-Lab-217y1a05c0/Lab-10$ cc Leakybucket.c -w
~/CN-Lab-217y1a05c0/Lab-10$ ./a.out
Enter bucket size, outgoing rate and no of inputs: 20 5 3
Enter the in packet size : 10
in packet size: 10
Bucket buffer size 10 out of 20
After outgoing 5 packets left out of 20 in buffer:
Enter the in packet size : 7
in packet size: 7
Bucket buffer size 12 out of 20
After outgoing 7 packets left out of 20 in buffer:
Enter the in packet size : 14
in packet size: 14
Dropped 1 no of packets
Bucket buffer size 20 out of: 20
After outgoing 15 packets left out of 20 in buffer:
~/CN-Lab-217y1a05c0/Lab-10$
```