**PRACTICAL NO.11**

**Aim:** Implementation of Distance Vector Algorithm (Bellmen Ford Algorithm) using C/C++.

**CODE:**

#include<stdio.h>

struct node

{unsigned dist[20];

   unsigned from[20];

}rt[10];

int main()

{int costmat[20][20];

   int nodes,i,j,k,count=0;

   printf("\nEnter the number of nodes : ");

   scanf("%d",&nodes);

   printf("\nEnter the cost matrix :\n");

   for(i=0;i<nodes ; i++)

   {

       for(j=0;j<nodes ; j++)

       {

           scanf("%d",&costmat[i][j]);

           costmat[i][i]=0;

           rt[i].dist[j]=costmat[i][j];

           rt[i].from[j]=j;

       }

   }

       do

       {

           count=0;

           for(i=0;i<nodes;i++)

           for(j=0;j<nodes;j++)

           for(k=0;k<nodes;k++)

               if(rt[i].dist[j]>costmat[i][k]+rt[k].dist[j])

               {

                   rt[i].dist[j]=rt[i].dist[k]+rt[k].dist[j];

                   rt[i].from[j]=k;

                   count++;

               }

       }while(count!=0);

       for(i=0;i<nodes;i++)

       {

           printf("\n\n For router %d\n",i+1);

           for(j=0;j<nodes;j++)

           {

               printf("\t\nnode %d via %d Distance %d ",j+1,rt[i].from[j]+1,rt[i].dist[j]);

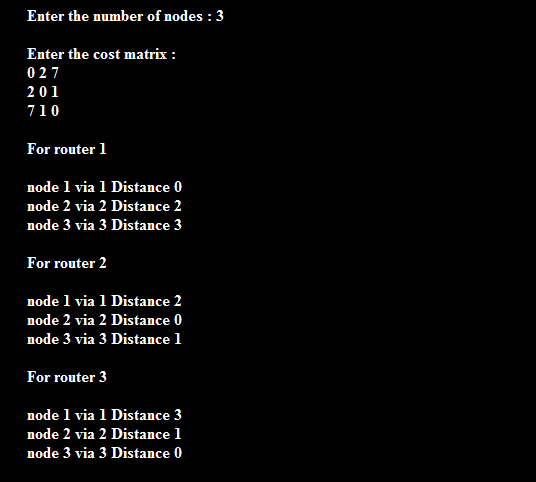
           }

       } printf("\n\n");

   getch();

}

**OUTPUT:**



**Conclusion:**  Hence we have studied and implemented Distance Vector Algorithm (Bellmen

Ford Algorithm) using C/C++.

**EXPERIMENT NO. 12**

**Aim:** To study CRC checksum and implement it on C/C++/JAVA.

**CODE:**

#include<stdio.h>

#include<string.h>

#define N strlen(g)

char t[28],cs[28],g[]="10001000000100001";

int a,e,c;

void xor(){

   for(c = 1;c < N; c++)

   cs[c] = (( cs[c] == g[c])?'0':'1');

}

void crc(){

   for(e=0;e<N;e++)

       cs[e]=t[e];

   do{

       if(cs[0]=='1')

           xor();

       for(c=0;c<N-1;c++)

           cs[c]=cs[c+1];

       cs[c]=t[e++];

   }while(e<=a+N-1);

}

void main()

{

   printf("\nEnter data : ");

   scanf("%s",t);

       printf("\nGeneratng polynomial : %s",g);

   a=strlen(t);

   for(e=a;e<a+N-1;e++)

       t[e]='0';

     printf("\nModified data is : %s",t);

     crc();

   printf("\nChecksum is : %s",cs);

   for(e=a;e<a+N-1;e++)

       t[e]=cs[e-a];

    printf("\nFinal codeword is : %s",t);

     printf("\nTest error detection 0(yes) 1(no)? : ");

   scanf("%d",&e);

   if(e==0)

   {

       do{

           printf("\nEnter the position where error is to be inserted : ");

           scanf("%d",&e);

       }while(e==0 || e>a+N-1);

       t[e-1]=(t[e-1]=='0')?'1':'0';

              printf("\nErroneous data : %s\n",t);

   }

   crc();

   for(e=0;(e<N-1) && (cs[e]!='1');e++);

       if(e<N-1)

           printf("\nError detected\n\n");

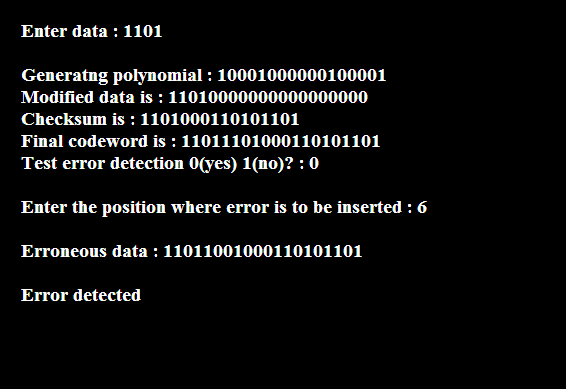
       else

           printf("\nNo error detected\n\n");

                getch();

}

**OUTPUT:**



**Conclusion:** Hence we have studied and implemented CRC checksum in C/C++.