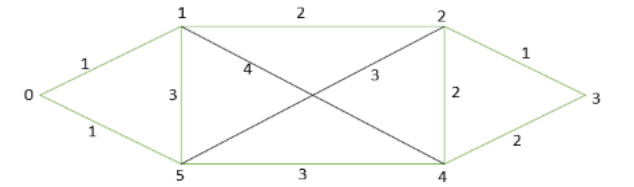
Name : Bijay Regmi
Roll Number : 210913032

DESTINATION SEQUENCED DISTANCE-VECTOR ROUTING PROTOCOL (DSDV)



Given Network

Source Code

```
#include <stdio.h>
#include <stdlib.h>
#define INFINITY 999
#define MAX 10
int SeqNo =0;
int RT[MAX][4];

void FindPath(int G[MAX][MAX], int n, int start, int seqNo)
{
   int i,j,count,min,nextNode;
   int C[MAX][MAX],dist[MAX],path[MAX],v[MAX];


   for(i=0;i<n;i++)
      for(j=0;j<n;j++)
         if(G[i][j]==0)
            C[i][j] = INFINITY;
         else
            C[i][j] = G[i][j];

   for(i=0;i<n;i++)
   {
      dist[i]=C[start][i];
      path[i]=start;
      v[i]=0;
      for(j=0;j<4;j++){
         RT[i][j]=0;
```

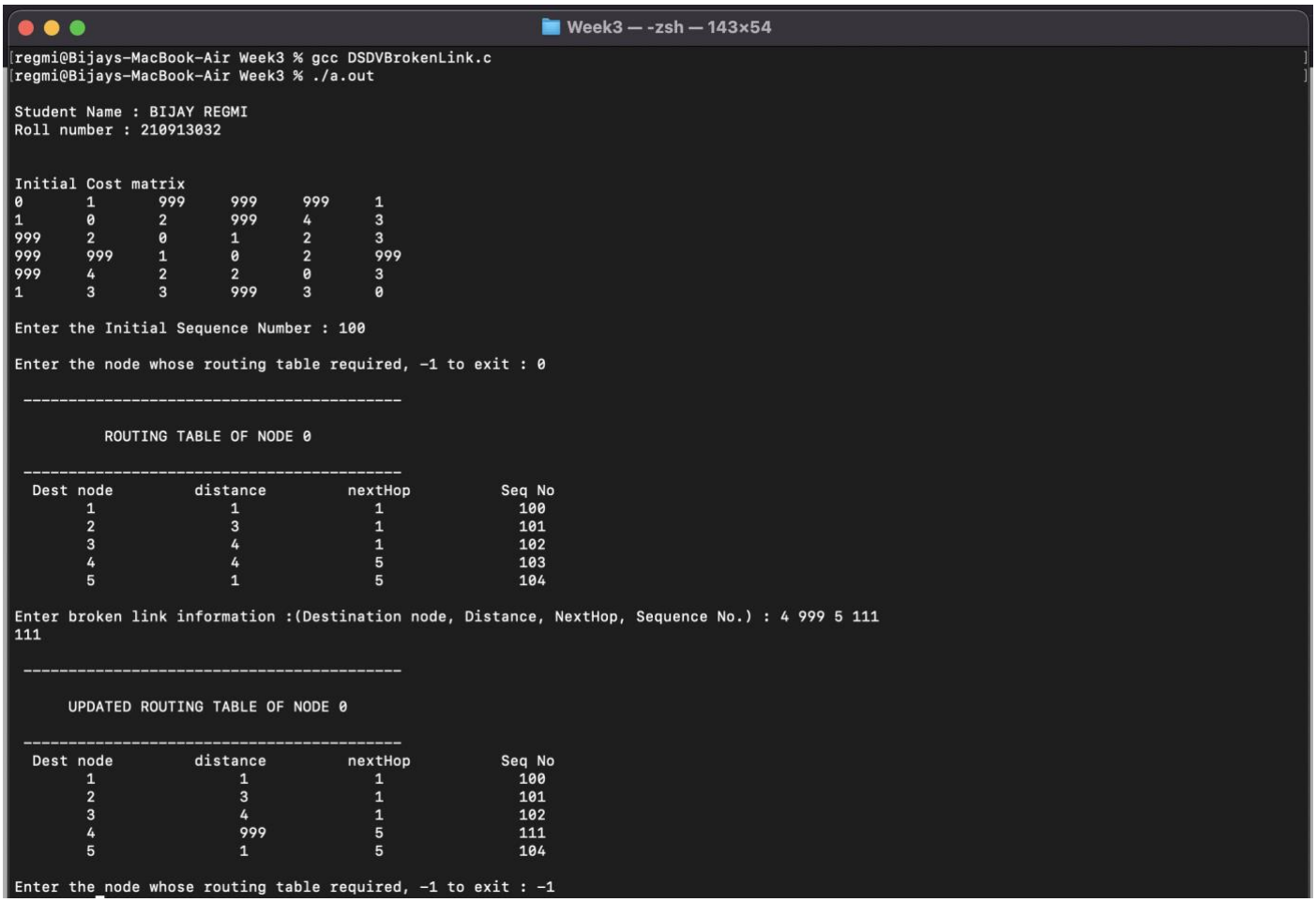
```
    }
    dist[start]=0;
    v[start]=1;
    count=1;
}

while(count < n-1)
{
    min = INFINITY;
    for(i=0;i<n;i++){
        if(dist[i] < min && !v[i]){
            min = dist[i];
            nextNode = i;
        }
    }
    v[nextNode] =1;
    for(i=0;i<n;i++){
        if(!v[i]){
            if(min+C[nextNode][i] < dist[i]){
                dist[i] = min+C[nextNode][i];
                path[i]= nextNode;
            }
        }
    }
    count++;
}

int k=0; int prev = start;
for(i=0;i<n;i++){
        if(i != start){
            if(min+C[nextNode][i] < dist[i]) {
                dist[i] = min+C[nextNode][i];
                path[i]= nextNode;
             }
        RT[k][0] = i;
        RT[k][1] = dist[i];
        j=i;
        while(j !=start){
            prev = j;
            j= path[j];
             }
        RT[k][2] = prev; RT[k][3] = seqNo; seqNo++;
        k++;
        }
```

```
        }
    printf("\n ----------------------------------------\n");
    printf("\n          ROUTING TABLE OF NODE %d \n",start);
    printf("\n ----------------------------------------\n");
    printf("  Dest node         distance         nextHop          Seq No\t\n");
    for(i=0;i<n-1;i++){
        printf("\t%d\t\t%d\t\t%d\t\t%d\t",RT[i][0],RT[i][1],RT[i][2],RT[i][3]);
        printf("\n");
    }
}

void UpdateTable(int dest, int dis , int next , int seq, int n) {
    int i;
    for(i =0 ; i < n-1 ; i++) {
        if(RT[i][0] == dest) {
            if(RT[i][3] < seq){
                RT[i][1] = dis;
                RT[i][2] = next;
                RT[i][3] = seq;
            }
            printf("%d\n",RT[i][3]); }
        }
 }

int main()
{
    printf("\nStudent Name : BIJAY REGMI\nRoll number : 210913032\n\n");
    int i,j,n,s;
    // int C[MAX][MAX];
    // printf("Enter no. of vertices: ");
    // scanf("%d",&n);

    // printf("Enter the cost matrix<999 for infinity>:\n");
    // for(i=0;i<n;i++)
    //     for(j=0;j<n;j++)
    //         scanf("%d",&C[i][j]);

    n=6;
    int C[MAX][MAX] = {
        {0,  1,   999,   999,   999,   1},
        {1,  0,   2,   999,   4,   3},
        {999,  2,   0,   1,   2,   3},
        {999,  999,   1,   0,   2,   999},
        {999,  4,   2,   2,   0,   3},
```

```
    {1,  3,   3,   999,   3,   0}
  };


  printf("\nInitial Cost matrix \n");
  for(i=0;i<n;i++)
    {
      for(j=0;j<n;j++){
        if(C[i][j] < 0) {
          printf(" \n\n ERROR..!! Negative Values are denied in the network\n");
          return 0;
        }
      printf("%d \t" , C[i][j]);
      }
    printf("\n");
    }

  printf("\nEnter the Initial Sequence Number : ");
  scanf("%d",&SeqNo);

  for(;;)
  {
    printf("\nEnter the node whose routing table required, -1 to exit : ");
    scanf("%d",&s);
    if((s == -1) || (s > n-1)) {
      printf("EXIT ..!!");
      break;
      }
    FindPath(C,n,s,SeqNo);
    int uDest,uDist,uNext,uSeq;
    printf("\nEnter broken link information :(Destination node, Distance, NextHop, Sequence No.) : "); scanf("%d %d %d %d",
&uDest, &uDist, &uNext, &uSeq);
    UpdateTable(uDest,uDist,uNext,uSeq,n);
    printf("\n ----------------------------------------\n");
    printf("\n      UPDATED ROUTING TABLE OF NODE %d \n",s);
    printf("\n ----------------------------------------\n");
    printf("  Dest node         distance         nextHop          Seq No\t\n");
    for(i=0;i<n-1;i++)
    {
      printf("\t%d\t\t %d\t\t%d\t\t%d\t",RT[i][0],RT[i][1],RT[i][2],RT[i][3]);
      printf("\n");
    }

  }
```

```
return 0;
}
```

INPUT/OUTPUT



```
Week3 — -zsh — 143×54
regmi@Bijays-MacBook-Air Week3 % gcc DSDVBrokenLink.c
regmi@Bijays-MacBook-Air Week3 % ./a.out

Student Name : BIJAY REGMI
Roll number : 210913032


Initial Cost matrix
0       1       999     999     999     1
1       0       2       999     4       3
999     2       0       1       2       3
999     999     1       0       2       999
999     4       2       2       0       3
1       3       3       999     3       0

Enter the Initial Sequence Number : 100

Enter the node whose routing table required, -1 to exit : 0

 ----------------------------------------

          ROUTING TABLE OF NODE 0

 ----------------------------------------
  Dest node         distance         nextHop          Seq No
       1               1               1               100
       2               3               1               101
       3               4               1               102
       4               4               5               103
       5               1               5               104

Enter broken link information :(Destination node, Distance, NextHop, Sequence No.) : 4 999 5 111
111

 ----------------------------------------

      UPDATED ROUTING TABLE OF NODE 0

 ----------------------------------------
  Dest node         distance         nextHop          Seq No
       1               1               1               100
       2               3               1               101
       3               4               1               102
       4               999             5               111
       5               1               5               104

Enter the node whose routing table required, -1 to exit : -1
```