```
/***********************************************************\
 *
 * THUAT TOAN BELLMAN
 * Tim duong di nhat tu mot dinh den cac dinh con lai
 * (dung cho do thi co trong so am)
 *
 *
\***********************************************************/

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

#define MAX 100
#define VOCUC 32767 //MAXINT

struct graph
{
	int n;
	int a[MAX][MAX];
};

//cac bien cua thuat toan Bellman
int previous[MAX+1][MAX]; //luu lai duong di ngan nhat
int mincost[MAX+1][MAX];  //do dai duong di tu i den j
int step;

// hai bien phuc vu doc va xuat file
char* fname_in = "D:\\DT_MTTS.TXT";
char* fname_out = "D:\\DT_MTTS_KQ_BELLMAN.TXT";

void DocDt(char* fname_in, graph &g)
{
	FILE* f;
	f = fopen(fname_in, "rt");
	if (f==NULL)
	{
		printf("Khong doc duoc file!\n");
		exit(0);
	}

	fscanf(f,"%d", &g.n);
	for (int i=0; i<g.n; i++)
		for (int j=0; j<g.n; j++)
			fscanf(f,"%d", &g.a[i][j]);
	fclose(f);
}

//KHOI TAO BELLMAN
void Init(graph g, int x)
```

```
{
    step = 0;
    for (int i=0; i<g.n; i++)
    {
        mincost[step][i] = VOCUC;
        previous[step][i] = i;
    }
    mincost[step][x] = 0;
}

//THUAT TOAN BELLMAN
void BellmanAlg(graph g, int x)
{
    Init(g,x);
    for (step=1; step<=g.n; step++)
    {
        for (int t=0; t<g.n; t++)
        {
            mincost[step][t] = mincost[step-1][t];
            previous[step][t] = previous[step-1][t];
        }
        for (int i=0; i<g.n; i++)
        {
            //tim cac dinh j co duong noi tu j den i
            //va chi phi buoc step-1 cua j khac VOCUC
            for (int j=0; j<g.n; j++)
            {
                if (g.a[i][j] != 0 && mincost[step-1][i] != VOCUC)
                {
                    //cap nhat lai neu chi phi buoc step cua i la VOCUC
                    //hoc chi phi di qua j: mincost[step-1][j] + a[j][i]
toi uu hon
                    if (mincost[step][j] == VOCUC || mincost[step][j] >
mincost[step-1][i] + g.a[i][j])
                    {
                        mincost[step][j]    =    mincost[step-1][i]    +
g.a[i][j];
                        previous[step][j] = i;
                    }
                }
            }
        }

        //so sanh mincost[step] voi mincost[step-1] neu bang thi ket thuc
        int bSame = true;
        for (i=0; i<g.n; i++)
            if (mincost[step][i] != mincost[step-1][i])
            {
                bSame = false;
                break;
            }
        if (bSame)
            break;
    }
}

void XuatBellman(graph g, int x)
{
    FILE* f;
    f = fopen(fname_out, "wt");
    if (f == NULL)
    {
        printf("Khong ghi duoc file!\n");
        exit(0);
    }



    int z = 0;
    int k = x, s;
    int t[MAX];

    if (step == g.n+1)
        fprintf(f,"Do thi co chu trinh am \n");
    else
    {
        for (int i=0; i<g.n; i++)
        {
            if  (previous[step-1][i]  ==  i  ||  x==i  ||  previous[step-
1][i]==VOCUC)
                fprintf(f,"Tu dinh %d -> %d khong co duong di \n", x+1,
i+1);
            else
            {
                fprintf(f,"Tu  dinh  %d  ->  %d  co  do  dai  %d",  x+1,  i+1,
mincost[step-1][i]);
                s = i;
                t[0] = i;
                z = 1;
                do
                {
                    t[z++] = previous[step][s];
                    s = previous[step][s];
                } while (s != x);

                fprintf(f," duong di: ");
                for (int j=z-1; j>0; j--)
                    fprintf(f, "%d -> ", t[j]+1);
                fprintf(f,"%d", t[0]+1);
                fprintf(f,"\n");
            }
        }
```

```
     }

     fclose(f);
}

void main()
{
     graph g;
     DocDt(fname_in, g);

     BellmanAlg(g,2);
     XuatBellman(g,2);
}

6
0 1 2 0 0 0
-2 0 0 8 0 0
0 0 0 5 -1 4
0 0 0 0 0 0
0 0 0 0 0 2
0 0 0 1 0 0

//ma tran trong so
ket qua:
Tu dinh 3 -> 1 khong co duong di
Tu dinh 3 -> 2 khong co duong di
Tu dinh 3 -> 3 khong co duong di
Tu dinh 3 -> 4 co do dai 2 duong di: 3 -> 5 -> 6 -> 4
Tu dinh 3 -> 5 co do dai -1 duong di: 3 -> 5
Tu dinh 3 -> 6 co do dai 1 duong di: 3 -> 5 -> 6
```