```c
#include "const.h"
#include "graph.h"
#include "routage.h"

#include <stdlib.h> // rand ()

#include <stdio.h> // printf ()

#define inf 9999


routage* init(graphe* G, int taille) {
    int i, j;
    routage* R = malloc(sizeof(routage));


    for(i = 0; i < taille; ++i)
    {
        for(j = 0; j < taille; ++j)
        {
            R->poids[i][j] = G->list[i][j];

            if(R->poids[i][j] == -1)
            {
                R->poids[i][j] = inf;
                R->succ[i][j] = -1;
            }
            else
            {
                R->succ[i][j] = j;
            }
        }
    }

    for(i = 0; i < taille; ++i)
    {
        R->poids[i][i] = 0;
        R->succ[i][i] = i;
    }
    return R;
}

void Floyd_Warshall(routage* R, int taille) {
    int i, j, k;

    for(k = 0; k < taille; k++)
    {
        for(i = 0; i < taille; i++)
        {
            for(j = 0; j < taille; j++)
            {
                if(R->poids[i][k] != inf && R->poids[k][j] != inf
                    && (R->poids[i][j] > (R->poids[i][k] + R->poids[k][j])) )
                {
                    R->poids[i][j] = R->poids[i][k] + R->poids[k][j];
                    R->succ[i][j] = R->succ[i][k];
                }

            }
        }
    }
}

void afficher_chemin(routage* R, int deb, int fin) {
    int stock_deb = deb;
    int voisin[TAILLE_GRAPHE] = {-1};

    int i, suiv = R->succ[deb][fin];
    for(i = 0; suiv != fin && i < TAILLE_GRAPHE; i++)
    {
        deb = R->succ[deb][fin];
        voisin[i] = deb;
        suiv = R->succ[deb][fin];
    }

    if(suiv == fin)
    {
        voisin[i] = fin;

        printf("\nChemin de %d à %d :\n%d", stock_deb, fin, stock_deb);
        for(int j = 0 ; j < i+1; j++)
        {
            printf(" -> %d", voisin[j]);
        }
        printf("\n");
    }
    else
    {
        printf("error\n");
    }
}

void libere_routage(routage* R) { free(R); }

```