SOURCE CODE ĐẦY ĐỦ THAM KHẢO VỀ ĐỒ THỊ LIÊN THÔNG

Đây là source code, các bạn có thể tham khảo nhé. Nếu trong quá trình tham khảo mà bạn không hiểu rõ chỗ nào thì chạy qua mở các file Hướng dẫn Code xem chú thích nhé.

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

#define MAX 10 // định nghĩa giá trị MAX
#define  inputfile  “C:/test.txt” // định nghĩa đường dẫn tuyệt đối đến file chứa thông tin của đồ thị
typedef struct GRAPH {
        int n; // số đỉnh của đồ thị
        int a[MAX][MAX]; // ma trận kề của đồ thị
}DOTHI;
// doc ma tran ke
int DocMaTranKe(char sTenFile[100], DOTHI &g)
{
        FILE* f; // một biến FILE
        f = fopen(sTenFile, "rt");// mở một file có đường dẫn là TenFile
        if (f == NULL) // nếu file mở được thì biến f != NULL và file không mở được thì biến f = NULL
        {
                printf("Khong mo duoc file\n");
                return 0; // không đọc được file trả về kết quả 0
        }
```

```
	fscanf(f, "%d", &g.n); // Đọc giá trị đỉnh của đồ thị vào biến n của cấu trúc DOTHI g
	// Đọc giá trị của ma trận a từ file vào (dùng 2 vòng for, dòng trước, cột sau để đọc từng phần tử của ma trận)
	// với a[i][j] là giá trị ma trận tại dòng I, cột j
	int i, j;
	for (i=0; i<g.n; i++)
	{
		for (j=0; j<g.n; j++)
		{
			fscanf(f, "%d", &g.a[i][j]); // đọc từng giá trị và gán vào ma trận kề a
		}
	}
	// đóng file đã mở ở trên.
	fclose(f);
	return 1; // trả về kết quả 1 cho biết đã đọc file và sử lý nhập thông tin đồ thị xong
}

//xuat thong tin cua do thi
void XuatMaTranKe (DOTHI g)
{
	printf("So dinh cua do thi la %d\n", g.n);
	printf("Ma tran ke cua do thi la\n");
	for (int i = 0; i < g.n; i++)
	{
		printf ("\t");
		for (int j = 0; j < g.n; j++)
		{
			printf("%d ",g.a[i][j]);
```

```
                }
                printf("\n");
        }
}
// kiem tra ma tran ke hop le
int KiemTraMaTranKeHopLe(DOTHI g)
{
        int i;
        for (i=0; i<g.n; i++)
        {
                if (g.a[i][i] != 0) /* kiểm tra nếu tồn tại một giá trị trên đường chéo khác 0. Thì trả về giá trị 0 (ma trận kề không hợp lệ) */
                        return 0;
        }
        return 1; // trả về 1 nếu tất cả các giá trị trên đường chéo là 0
}
// kiem tra do thi vo huong
int KiemTraDoThiVoHuong(DOTHI g)
{
        int i, j;
        for (i=0; i<g.n; i++)
        {
                for (j=0; j<g.n; j++)
                {
                        if (g.a[i][j] != g.a[j][i]) /* kiểm tra nếu tồn tại một giá trị a[i][j] != a[j][i] thì tức ma trận kề không đối xứng, lúc đó đồ thị không phải là vô hướng. trả về kết quả 0 (đồ thị không phải là vô hướng) */
                                return 0;
                }
```

```
        }
        return 1;
}
// đi tìm các đỉnh liên thông với đỉnh I, tức có cạnh nối trực tiếp tới đỉnh i
void DiTimCacDinhLienThong (DOTHI g, int nhan[MAX], int i)
{
        for (int j = 0; j < g.n; j++)
        {
                if (g.a[i][j] != 0 && nhan[j] != nhan[i]) // nếu tồn tại cạnh giữa đỉnh I và đỉnh j, đồng thời nhãn của đỉnh j khác với nhãn của đỉnh i (nhãn thành phần liên thông) thì thực hiện gán nhãn của đỉnh j = nhãn của đỉnh i và DiTimCacDinhLienThong với đỉnh j
                {
                        nhan[j] = nhan[i]; // gán nhãn cho đỉnh j
                        DiTimCacDinhLienThong (g,nhan,j); // tiếp tục DiTimCacDinhLienThong  với đỉnh j và gán nhãn tương ứng
                }
        }
}

//xet tinh lien thong cua do thi
void XetLienThong(DOTHI g)
{
        int Nhan[MAX]; // tạo một mảng Nhãn để lưu lại nhãn của  các đỉnh trong đồ thị g
        int i;
        for (i=0;i<g.n;i++)// gán nhãn ban đầu cho tất cả các đỉnh của đồ thị g là 0
                Nhan[i] =0;
```

```
int SoThanhPhanLT = 0; // lưu lại số thành phần liên thông trong đồ thị g, ban đầu là 0. Tức chưa có thành phần nào.

// duyệt lần lượt tất cả các đỉnh và chọn đỉnh có nhãn là 0. Ta bắt đầu xét
for (i=0; i<g.n; i++)
{
    if (Nhan[i] == 0) // có một đỉnh trong đồ thị có nhãn là 0
    {
        SoThanhPhanLT ++; // tăng số thành phần liên thông lên
        Nhan[i] = SoThanhPhanLT; // gán nhãn cho đỉnh đó bởi
    }
    if (Nhan[i] != 0) // nếu đỉnh i đã gán nhãn rồi thì từ đó đi tìm những đỉnh khác thuộc về thành phần liên thông của nó
        DiTimCacDinhLienThong (g, Nhan, i); // gọi hàm đi tìm các đỉnh liên thông với đỉnh i và gán nhãn cho nó. Hàm này được khai báo ở sau

}

printf ("So thanh phan lien thong la %d\n",SoThanhPhanLT);
for (i = 1; i <= SoThanhPhanLT; i++)
{
    printf("Thanh phan lien thong thu %d gom cac dinh ", i);
    for (int j = 0; j < g.n; j++)
    {
        if (Nhan[j] == i) /* kiểm tra trong mảng nhãn đó thì những đỉnh j nào được gán nhãn là I tức đỉnh j thuộc về thành phần liên thông thứ I thì xuất ra*/
            printf (" %d ",j);
    }
    printf("\n");
```

```
	}
}
void main()
{
	DOTHI g;
	clrscr();
	if (DocMaTranKe(inputfile, g) == 1)
	{
		printf("Da lay thong tin do thi tu file thanh cong.\n\n");
		XuatMaTranKe(g);
		printf("Bam 1 phim bat ki de tien hanh kiem tra do thi ...\n\n");
		getch();
		if (KiemTraMaTranKeHopLe(g) == 1)
			printf ("Do thi hop le.\n");
		else
			printf ("Do thi khong hop le.\n");
		if (KiemTraDoThiVoHuong(g) == 1)
			printf ("Do thi vo huong.\n");
		else
			printf ("Do thi co huong.\n");
		printf("Bam 1 phim bat ki de bat dau xet tinh lien thong cua do thi ...\n\n");
		getch();
		XetLienThong(g);
	}
	getch();
}
```

Chúc các bạn may mắn và học tốt môn này

GOOD LUCK TO U