HƯỚNG DẪN CODE NHẬP XUẤT MA TRẬN KỀ TỪ FILE

Chú ý: Trong hướng dẫn này, chỗ nào có cụm từ "bạn viết code" hay đại loại thế. Thì bạn phải viết code chỗ đó hén.

Đầu tiên: tạo một cấu trúc lưu trữ đồ thị như sau:

```
#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;
```

Giải thích: struct dùng để khai một cấu trúc DOTHI (đồ thị) gồm có số đỉnh của đồ thị là n, và ma trận kề là mảng 2 chiều a. Cách sử dụng cấu trúc struct này hoàn toàn tương tự cách sử dụng đối với kiểu dữ liệu thông thường cơ bản như int a, char c, …..

Ví dụ: bạn muốn tạo một biến đồ thị thì bạn làm như sau:

```
DOTHI g;
```

Còn bạn muốn gán giá trị số đỉnh của đồ thị vào g (chẳng hạn đồ thị có 8 đỉnh) thì bạn làm như sau:

```
g.n = 8; // giống như bạn sử dụng một biến int, char, float thông thường nhưng chỉ khác chỗ là có thêm ở đằng trước g.
```

Điều này có nghĩa là nếu bạn muốn truy cập vào thành phần nào của đồ thị thì bạn dùng <tên_biến_thuộc_kiểu_đồ_thị> và dấu chấm.

Ví dụ: `g.n` //truy cập vào giá trị n của biến g

g.a[i][j] //truy cập vào giá trị hàng i cột j của mảng a của biến g.

Còn nếu bạn muốn sử dụng các giá trị n (số đỉnh của đồ thị), a[i][j] (giá trị ma trận kề) thì bạn chỉ việc dùng <tên_biến_thuộc_kiểu_đồ_thị> và dấu chấm như trên

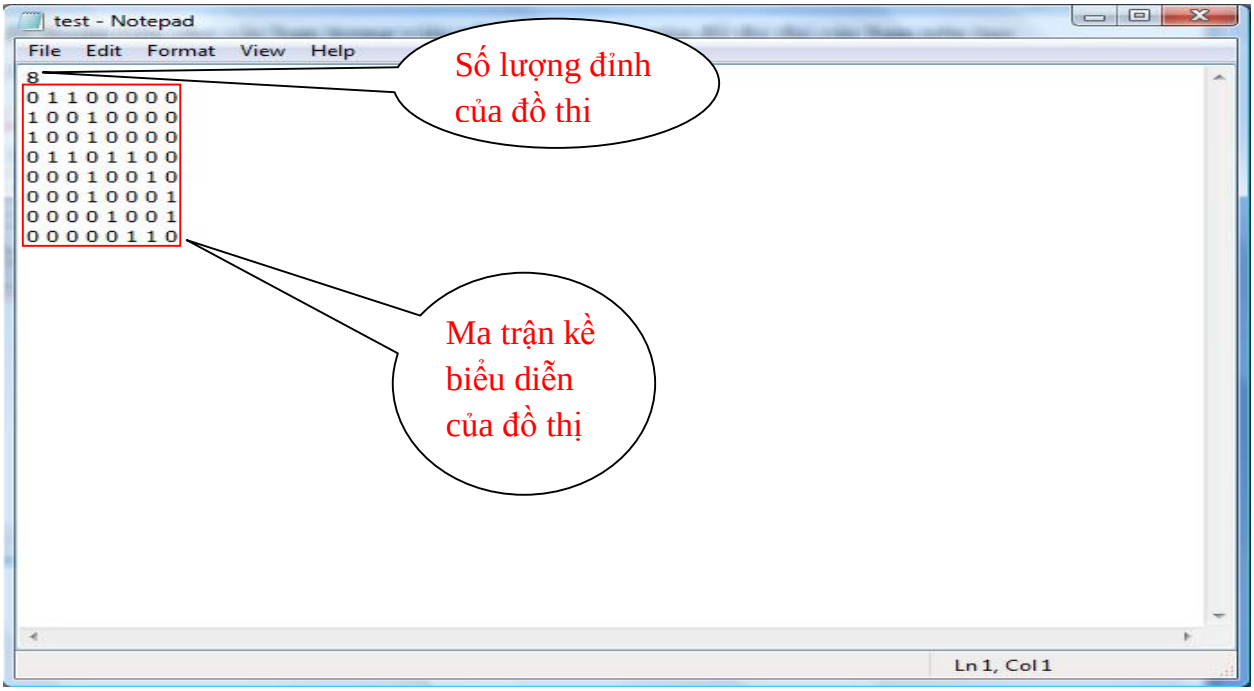
Ví dụ: lấy số đỉnh của đồ thị g gán vào biến int h và giá trị đầu tiên (chỉ số i = 0, j = 0) trong ma trận kề của đồ thị g vào biến int k thì làm như sau:

```
h = g.n;
k = g.a[0][0];
```

Câu hỏi đặt ra ở đây là: Sau khi bạn đã có cấu trúc DOTHI để lưu thông của 1 đồ thị. Thì vậy bạn lấy thông tin từ đâu để gán vào biến n (số đỉnh), và biến a (ma trận kề) của đồ thị g??? Bạn có thể nhập từ console nhưng sẽ lâu và tốn thời gian chẳng hạn như bạn nhập một đồ thị có 10 đỉnh thì ma trận kề của nó là 10*10 = 100. Bạn có đủ kiên nhẫn để nhập 1 + 100 = 101 số không??? Câu trả lời tất nhiên là **KO nhé**.

Do đó, để thuận tiện cho các bạn trong việc nhập thông tin của đồ thị thì các bạn nên tạo một cái file có cấu trúc như sau lưu trữ thông tin của đồ thị bạn cần test.

Bước 1, bạn mở NOTEPAD lên tạo một file có cấu trúc tổ chức như sau:



Sau đó bạn lưu lại với tên file bất kì gì đó chẳng hạn như bạn lưu với tên test.txt, ở ổ **C**.

Bước 2: Lấy/đọc những thông tin đồ thị từ file bạn đã lưu ở trên (ví dụ C:/test.txt) gán vào một biến DOTHI g. Việc đó được làm thông qua hàm DocMaTranKe như sau:

```
int DocMaTranKe(char TenFile[100], DOTHI &g)
{
    FILE* f;  // một biến FILE
    f = fopen(TenFile, "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
    }
    // Đọc giá trị đỉnh của đồ thị vào biến n của cấu trúc DOTHI g
    fscanf(f, "%d", &g.n);
```

```
	// 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
}
```

Cách thức gọi hàm DocMaTranKe này trong hàm main như sau:

```
void main()
{
	DOTHI g;
	DocMaTranKe("C:/test.txt",g);
}
```

Tuy nhiên, để xem kết quả đọc thông tin đồ thị từ file có đúng hay không? bạn viết hàm XuatMaTranKe như sau để xuất thông tin của đồ thị g hiện tại nhé:

```
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");
	}
}
```

Bước 3: Kiểm tra tính hợp lệ của ma trận kề nhập vào: nó có phải là biểu diễn của một trong 2 dạng đồ thị mà giáo viên hướng dẫn đã trình bày cho các bạn không? Thì các bạn viết hàm **KiemTraMaTranHopLe** như sau:

```
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 nghĩa là a[i][j] != 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
}
```

Bước 4: kiểm tra đồ thị bạn nhập vào có là **vô hướng hay là có hướng**, bạn dùng hàm sau KiemTraDoThiVoHuong sau.

```
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; // trả về kết quả 1 nếu đồ thị là vô hướng
}
```

Bước 5: Code trong hàm main để gọi hàm các hàm tương ứng và chạy. Có thể làm như sau:

```
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");
    }
    getch();
}
```

HƯỚNG DẪN CHI TIẾT TỚI CỠ NÀY RỒI MÀ BẠN VẪN KHÔNG LÀM ĐƯỢC NỮA THÌ BẠN CHUẨN BỊ TINH THẦN ĐI HÉN J

LƯU Ý

Các bạn nhớ làm bài tập nhé, các bạn làm thì mới có điểm, không làm thì không có điểm. Khi đó bạn biết số phận của mình rồi hén.

Đối với những bài tập mà tôi yêu cầu các bạn nộp bài, thì cách thức nộp bài như sau:

- Các bạn xóa hết các thư mục râu ria như debug,….
- Sau đó nén chỉ mỗi file source của bạn lại với tên như sau:
  [Tên_Lớp]_[MaSoSinhVien]_[HovaTen]_[TenBaiTap].zip hoặc
  [Tên_Lớp]_[MaSoSinhVien]_[HovaTen]_[TenBaiTap].rar
  Ví dụ tôi học lớp 08HC, mã số sinh viên 0412006, làm bài tập chu trình và đường đi Euler thì tôi nén lại và đặt tên như sau

08HC_0412006_NguyenKimHung_Euler.zip hoặc

08HC_0412006_NguyenKimHung_Euler.rar

- Về thời gian và hình thức nộp tôi sẽ thông báo sau.

Tuy nhiên, cũng sẽ có trường hợp, tôi sẽ cho các bạn làm bài tập tại chỗ thực hành rồi tôi chấm luôn, khi đó các bạn không cần phải nộp bài. Hoặc có thể tôi sẽ kiểm tra những bài thực hành tuần trước của bạn (phần nào là do tôi ngẫu hứng chọn, điều này sẽ là hên xui cho các bạn hén **J**), yêu cầu bạn giải thích những đoạn code của bạn: đoạn đó làm gì? Kết quả ra sao hén? Chạy chương trình,.....

Chúc các bạn may mắn và học tốt môn này

GOOD LUCK TO U

----------HẾT---------