**Bài 1: (Mảng)**

* Viết chương trình thực hiện sắp xếp Topo

Để tổ chức sinh nhật bạn Tom có n công việc chuẩn bị. Mỗi công việc muốn thực hiện phải thực hiện một số công việc tiên quyết.

1. Tổ chức dữ liệu để lưu trữ các công việc của bạn Tom.
2. Chỉ ra một thứ tự công việc bạn Tom phải thực hiện để có buổi sinh nhật. Trong trường hợp không có thứ tự hãy thông báo cho bạn Tom.

Thuật toán sắp xếp Topo:

+ Sắp xếp thứ tự tuyến tính các đối tượng mà quan hệ tiên quyết

+ Phát hiện một đồ thị có chu trình hay ko?

Vd: Tom tổ chức sinh nhật

|  |  |
| --- | --- |
| 1. Đưa thiệp | 2 |
| 1. Viết Thiệp |  |
| 1. Đặt bánh | 6,7,8 |
| 1. Phở |  |
| 1. Cari | 9,10 |
| 1. Đi chợ |  |
| 1. Mua gà | 6 |
| 1. Gia vị | 6 |
| 1. Thịt bò | 6 |
| 1. Bánh mì | 6 |

Tổ chức lưu trữ dữ liệu cung từ i-> j nếu I công việc tiên quyết

1🡨------------------- 2

9,8,7🡨---------------------6

3🡨---------------------6,7,8

5🡨---------------------10,9

+ Lưu trữ đồ thị

Ma trận kề

|  |  |
| --- | --- |
|  | 1 2 3 4 5…..10 |
| 1 | 0 0 |
| 2 | 1 |
| … | aij |
| 10 |  |
|  |  |

Chọn 1 đỉnh **j** không có cung đếm (cột j mang giá trị 0)

Xóa tất cả các cung xuất phát từ j

Nếu hết j có 1 thứ tự Topo(đồ thị không chu trình). Ngược lại không sắp xếp được tuyến tính(đồ thị có chu trình)

1. Tạo sẵn trong chương trình
2. Nhập từ bàn phím
3. Đọc từ file

INP.dat

10->n là số đỉnh ma trận

0 0

1 0

0…….

**Bài 2: (Danh sách liên kết)**

1. Trình bày tổ chức dữ liệu để quản lý văn bản bằng danh sách đa liên kết với yêu cầu sau:

Mỗi nút của danh sách liên kết chứa một dòng, mỗi nút sẽ có thành phần liên kết để trỏ đến dòng đến trước, thành phần trỏ đến dòng đứng sau.

....................................................................

1. Viết thủ tục in ra các dòng văn bản từ dòng thứ i đến dòng thứ j.
2. Viết thủ tục chèn 1 dòng vào cuối văn bản, với nội dung dong mới nhập từ bàn phím.
3. Viết thủ tục xóa từ dòng thứ i đến dòng thứ j.

Tổ chức thực hiện

Struct nut{

Char nd[80];

Nut \*sau,\*truoc;};

}

Nut \*dau;

Void KhoiTao(nut \*&dau)

{ dau=NULL;

}

* Khai báo dữ liệu

Char<tên biến[<kth>];

Char nd[80];

* Nhập dữ liệu:

Fflush(stdin);// Xóa bộ đệm

Gets( tên biến);

* Xuất dữ liệu:

Printf(“%s”,<tên biến>);

* Gán dữ liệu:

Strcpy(<b1>,<b2>);// b2 gán cho biến b1

* So sánh chuỗi:

Strcmp(<b1>,<b2>);//b1==b2 ->0

Cấu trúc chuỗi ký tự:

Chuỗi kí tự == mảng 1 chiều, kết thúc “\0”;

Nếu nhập chuỗi gõ Enter, biến chuỗi có giá trị là “\0”

\*\*\*

1. Nhập văn bản:

+ Lần lượt nhập các chuỗi ký tự lưu thành dslk kết thúc khi gõ Enter

1. Xoá dòng i

+ tìm dòng i: p

* T,s trỏ đến dòng trước sau
* T-> sau=s;
* S-> truoc=t;
* Xóa p: delete p

+ p== dau

{dau= dau-> sau;

Dau->truoc = NULL;

Delete p;}

+ if( p-> sau =NULL)

{

T=p-> truoc;

T-> sau = NULL;

Delete p;}