

# BÀI TẬP THỰC HÀNH

## NGƯỜI ĐƯA THƯ – CHU TRÌNH HAMILTON

### 1 Quy định

- Thời gian làm bài: **1 tuần** (deadline xem trên moodle)
- Loại bài tập: cá nhân
- Cấu trúc bên trong thư mục MSSV bao gồm các thư mục
  - **Source**: thư mục chứa toàn bộ source code.
  - **Release**: thư mục chứa file thực thi (MSSV.exe).
- Nén toàn bộ thư mục thành file MSSV.zip hoặc .rar, nộp bài lên moodle.
- Tất cả các bài làm sai qui định sẽ được **điểm 0**.

### 2 Đề bài

Trong thành phố có  $N$  khu dân cư, đánh số từ 0 đến  $N-1$  ( $N < 20$ ). Giữa các khu dân cư có thể có hoặc không có đường đi trực tiếp, nhưng nếu có thì đó là đường 2 chiều. Một người đưa thư xuất phát từ khu dân cư thứ  $k$ , muốn đi qua **tất cả các khu dân cư đúng một lần** và cuối cùng **trở về điểm xuất phát**.

Cho bản đồ của thành phố và vị trí xuất phát của người đưa thư trong một tập tin đầu vào. Hãy viết chương trình tìm xem có tồn tại **chu trình** đó hay không, nếu có thì in chu trình và tổng chiều dài của chu trình đó trong tập tin đầu ra. Chương trình chạy tham số dòng lệnh có dạng sau:

*<Tên chương trình> <Tập tin đầu vào> <Tập tin đầu ra>*

#### 2.1 Cấu trúc file dữ liệu đầu vào

Bản đồ thành phố được biểu diễn bằng đồ thị  $N$  đỉnh:

- Dòng đầu tiên: khu dân cư xuất phát của người đưa thư ( $k$ )
- Dòng thứ 2: Số lượng khu dân cư trong thành phố, chính là số đỉnh của đồ thị ( $N$ )
- $N$  dòng tiếp theo: *ma trận kề* của đồ thị với quy ước:
  - $A[i][j] = 0$ : Không có đường đi trực tiếp giữa  $i$  và  $j$ .
  - $A[i][j] = w > 0$ : Đường đi giữa  $i$  và  $j$  có chiều dài  $w$  (km).

Do tất cả đường đi nếu có đều là 2 chiều nên Ma trận kề là ma trận đối xứng.

## 2.2 Cấu trúc file dữ liệu đầu ra

- Nếu không tìm được ghi là “NULL” (KHÔNG có dấu “”)
- Nếu tìm được chu trình thì:
  - Dòng đầu tiên: Chiều dài của chu trình.
  - Dòng thứ 2: lần lượt xuất ra các đỉnh trên chu trình đó (cách nhau bởi **khoảng trắng**).

## 2.3 Ví dụ:

Tập tin đầu vào	Bản đồ thành phố	Tập tin đầu ra
3 5 0 3 0 1 0 3 0 2 5 0 0 2 0 0 4 1 5 0 0 6 0 0 4 6 0	<pre> graph LR     0 --- 3  1     1 --- 2  2     0 --- 1  3     1 --- 5  3     2 --- 4  4     3 --- 6  4             </pre>	16 3 4 2 1 0 3