

## BÀI THỰC HÀNH SỐ 3 – Đường đi ngắn nhất

**Bài 1.** Cho đồ thị G (có hướng hoặc vô hướng), có số đỉnh là N ( $1 \leq N \leq 20$ ). Dùng thuật toán Dijkstra, dùng ngôn ngữ lập trình C (không chấp nhận C++ và ngôn ngữ khác) cài đặt chương trình tìm đường ngắn nhất từ một đỉnh bắt đầu u tới đỉnh cuối v cho trước.

Input: là ma trận kề. Ma trận được chứa trong file có tên “*dijkstra.inp*”, có định dạng sau:

- Dòng đầu tiên gồm ba số lần lượt là: số N – số đỉnh của đồ thị, đỉnh bắt đầu u, đỉnh cuối v. Mỗi số cách nhau ít nhất một khoảng trắng.
- N dòng tiếp theo là ma trận kề.

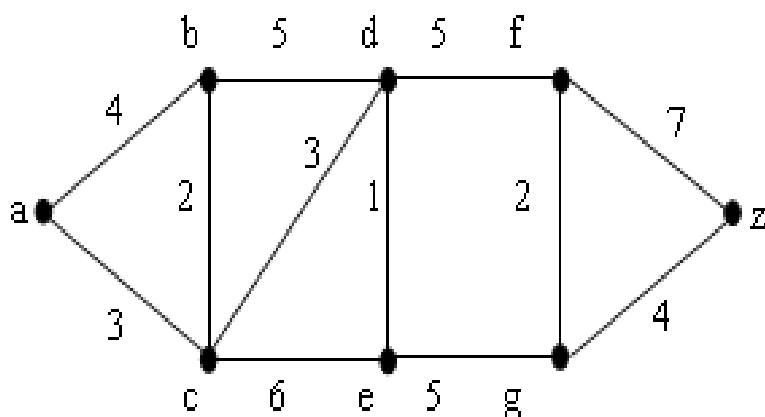
Ví dụ: Đồ thị có 8 đỉnh, đỉnh bắt đầu 4, đỉnh cuối là 8.

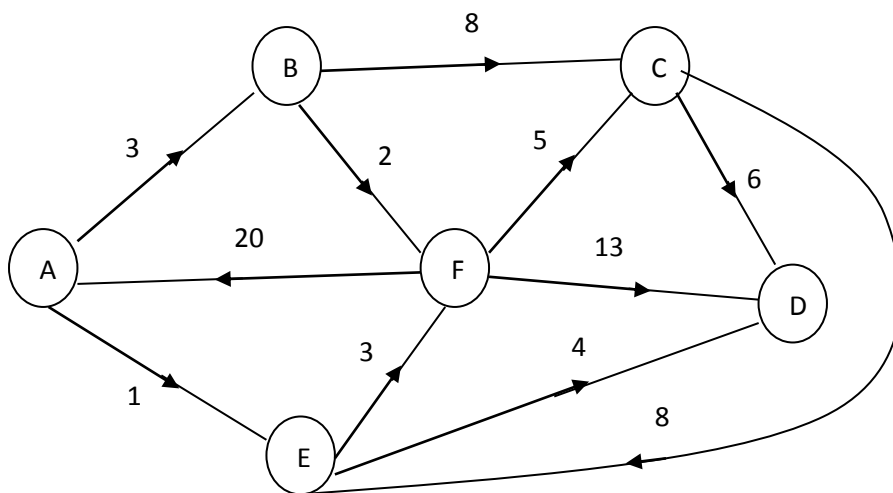
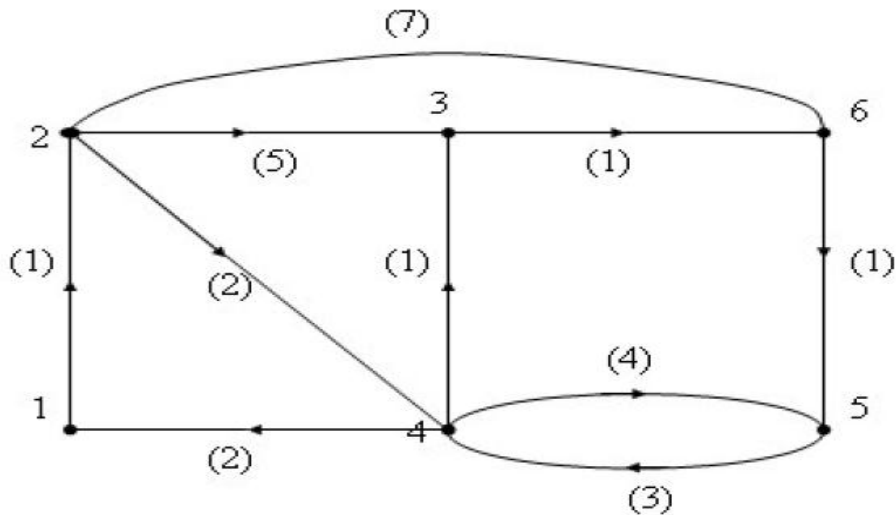
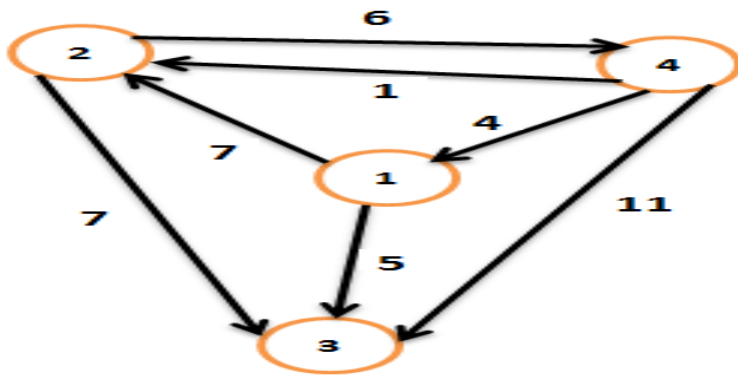
```
8 4 8
0 3 5 2 0 0 0 0
3 0 1 0 7 0 0 0
5 1 0 4 0 1 0 0
2 0 4 0 0 3 6 0
0 7 0 0 0 2 0 3
0 0 1 3 2 0 4 6
0 0 0 6 0 4 0 5
0 0 0 0 3 6 5 0
```

Output: In kết quả ra màn hình có định dạng sau: đỉnh đầu – đỉnh cuối: chi phí (đường đi).

Ví dụ: 4 – 8: 8 (4 → 6 → 5 → 8)

**Bài 2.** Hãy kiểm tra chương trình của Bài 1 với tất cả đồ thị dưới đây.





**Bài 3.** Lấy chương trình ở Bài 1, sửa lại nội dung code sao cho chương trình có thể tìm được đường đi ngắn nhất từ đỉnh bắt đầu u tới tất cả các đỉnh còn lại. **Lưu ý: không chấp nhận chương trình khác với Bài 1, phải lấy từ chương trình Bài 1 sửa lại để làm Bài 3.**

Input: như Bài 1, nhưng bỏ đi đỉnh v.

Output: In kết quả ra màn hình, nhưng gồm  $N - 1$  dòng, mỗi dòng là đường đi ngắn nhất từ đỉnh  $u$  tới  $N - 1$  đỉnh còn lại. Mỗi dòng đều có định dạng như *Bài 1*.

**Bài 4.** Cho đồ thị  $G$  (có hướng hoặc vô hướng), có số đỉnh là  $N$  ( $1 \leq N \leq 20$ ). Dựa vào thuật toán Floyd, dùng ngôn ngữ lập trình C (không chấp nhận C++ và ngôn ngữ khác) cài đặt chương trình tìm đường ngắn nhất từ mọi cặp đỉnh  $u, v$  bất kỳ trong đồ thị.

Input: là ma trận kề. Ma trận được chứa trong file có tên “floyd.inp”, có định dạng sau:

- Dòng đầu tiên gồm một số là  $N$ .
- $N$  dòng tiếp theo là ma trận kề.
- Dòng sau ma trận là một cặp đỉnh  $u, v$ . Cặp đỉnh này phục vụ cho phần output của bài toán.

Ví dụ: Đồ thị có 8 đỉnh

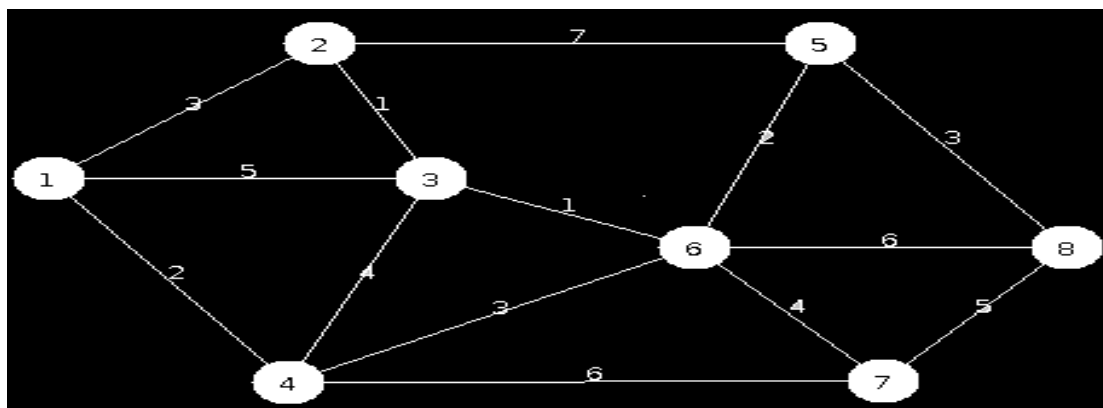
```
8
0 3 5 2 0 0 0 0
3 0 1 0 7 0 0 0
5 1 0 4 0 1 0 0
2 0 4 0 0 3 6 0
0 7 0 0 0 2 0 3
0 0 1 3 2 0 4 6
0 0 0 6 0 4 0 5
0 0 0 0 3 6 5 0
2 5
```

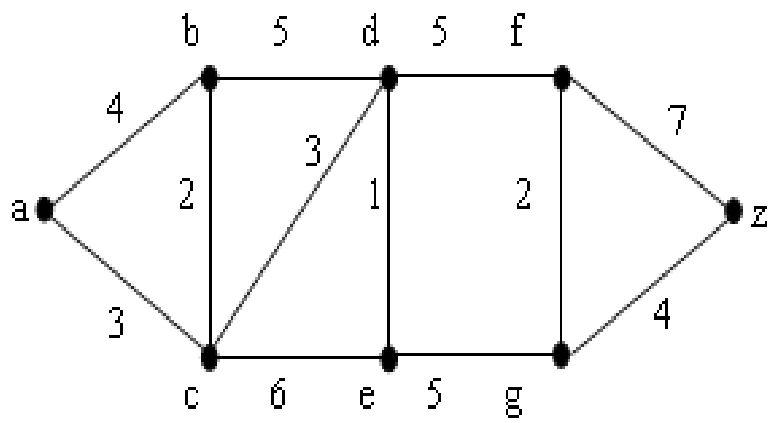
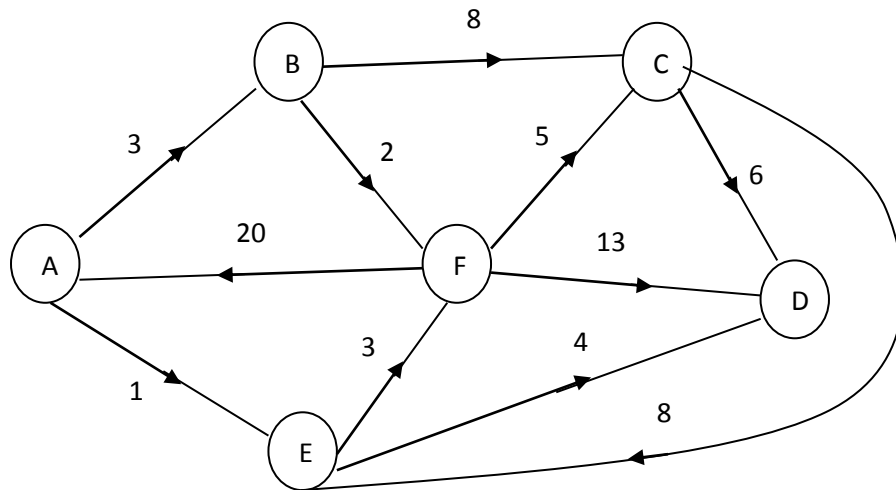
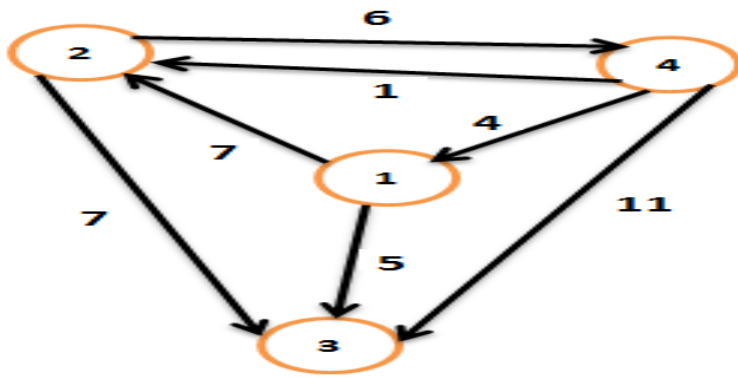
Output: In kết quả ra màn hình có định dạng sau:

- Gồm một dòng, có định dạng sau: đỉnh đầu – đỉnh cuối: chi phí (đường đi). Đỉnh đầu, đỉnh cuối là tùy thuộc vào giá trị của  $u$  và  $v$  trong phần input.

Ví dụ: 2 – 5: 4 (2 → 3 → 6 → 5)

**Bài 5.** Hãy kiểm tra chương trình của *Bài 4* với tất cả đồ thị dưới đây:





---THE END ---