Cho đồ thị **có hướng** G = <V, E> có n đỉnh và m cung (n < 100, m < 500). Mỗi cung được gán một trọng số w (-100 < w <= 100).

Viết chương trình tìm đường đi ngắn nhất từ đỉnh 1 đến các đỉnh còn lại.

**Đầu vào (Input):**

Dữ liệu đầu vào được nhập từ bàn phím với định dạng:

* Dòng đầu tiên chứa 2 số nguyên n và m.
* m dòng tiếp theo mỗi dòng chứa 3 số nguyên u, v, w mô tả cung (u, v) có trọng số w.

Dữ liệu được đảm bảo không tồn tại chu trình âm.

**Đầu ra (Output):**

In ra màn hình các giá trị pi và p của các đỉnh theo thứ tự 1, 2, ..., n.

Xem thêm ví dụ bên dưới.

**Chú ý:**

* Để chạy thử chương trình, bạn nên tạo một tập tin **dt.txt** chứa đồ thị cần kiểm tra.
* Thêm dòng freopen("dt.txt", "r", stdin); vào ngay sau hàm main(). Khi nộp bài, nhớ gỡ bỏ dòng này ra.
* Có thể sử dụng đoạn chương trình đọc dữ liệu mẫu sau đây:

freopen("dt.txt", "r", stdin); **//Khi nộp bài nhớ bỏ dòng này.**  
 Graph G;  
 int n, m, u, v, w, e;  
 scanf("%d%d", &n, &m);  
 init\_graph(&G, n);  
   
 for (e = 0; e < m; e++) {  
 scanf("%d%d%d", &u, &v, &c);  
 add\_edge(&G, u, v, w);  
 }

**For example:**

| **Input** | **Result** |
| --- | --- |
| 3 3   1 2 9 2 3 4 1 3 4 | pi[1] = 0, p[1] = -1 pi[2] = 9, p[2] = 1 pi[3] = 4, p[3] = 1 |
| 8 13 1 2 4 1 3 4 3 5 4 3 6 2 4 1 3 4 3 2 5 4 1 5 7 5 6 2 3 6 5 -3 7 6 2 7 8 2 8 5 -2 | pi[1] = 0, p[1] = -1 pi[2] = 4, p[2] = 1 pi[3] = 4, p[3] = 1 pi[4] = 4, p[4] = 5 pi[5] = 3, p[5] = 6 pi[6] = 6, p[6] = 3 pi[7] = 8, p[7] = 5 pi[8] = 10, p[8] = 7 |
| 6 9 1 2 7 1 3 9 1 5 14 2 3 10 2 4 15 3 4 11 3 5 2 4 6 6 5 6 9 | pi[1] = 0, p[1] = -1 pi[2] = 7, p[2] = 1 pi[3] = 9, p[3] = 1 pi[4] = 20, p[4] = 3 pi[5] = 11, p[5] = 3 pi[6] = 20, p[6] = 5 |