Question **1**

Correct

Mark 1.00 out of 1.00

Viết chương trình đọc vào một **đơn đồ thị có hướng, có trọng số**. Áp dụng thuật toán Floyd - Warshall để tìm đường đi ngắn nhất giữa các cặp đỉnh.

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 (0 < n < 100; 0 < m < 500).
- m dòng tiếp theo mỗi dòng chứa 3 số nguyên u, v, w mô tả cung (u, v) có trọng số w (-100 < w < 100).

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

Đầu ra (Output)

- In ra màn hình chiều dài đường đi ngắn nhất giữa các cặp đỉnh theo mẫu bên dưới. Nếu không có đường đi in ra **NO PATH**.

```
u -> v: chiều dài
```

- Liệt kê các cặp theo thứ tự tăng dần của u và v
- Xem thêm ví dụ bên dưới.

For example:

Input	Result
3 3 1 2 9 2 3 4 1 3 4	1 -> 1: 0 1 -> 2: 9 1 -> 3: 4 2 -> 1: NO PATH 2 -> 2: 0 2 -> 3: 4 3 -> 1: NO PATH 3 -> 2: NO PATH 3 -> 3: 0
3 4 1 2 9 2 3 4 1 3 4 2 1 -3	1 -> 1: 0 1 -> 2: 9 1 -> 3: 4 2 -> 1: -3 2 -> 2: 0 2 -> 3: 1 3 -> 1: NO PATH 3 -> 2: NO PATH 3 -> 3: 0

Answer: (penalty regime: 10, 20, ... %)

```c
//Viết chương trình đọc vào một đơn đồ thị có hướng, có tr
//Áp dụng thuật toán Floyd - Warshall để tìm đường đi ngắn

#include <stdio.h>
#define MAX_N 105
#define INF 100000000
#define NO_EDGE -1


int pi[MAX_N][MAX_N]; //đường đi ngắn nhất u -> v
int next[MAX_N][MAX_N];   //đỉnh kế tiếp đỉnh u trên đường
int path[MAX_N];

typedef struct{
    int n,m;
    int W[MAX_N][MAX_N];
}Graph;

//khởi tạo ma trận k/c
void init_graph (Graph *pG, int n){
```

```
21      pG->n = n;
22
```

Precheck Check

	Input	Expected	Got	
✔	3 3 1 2 9 2 3 4 1 3 4	1 -> 1: 0 1 -> 2: 9 1 -> 3: 4 2 -> 1: NO PATH 2 -> 2: 0 2 -> 3: 4 3 -> 1: NO PATH 3 -> 2: NO PATH 3 -> 3: 0	1 -> 1: 0 1 -> 2: 9 1 -> 3: 4 2 -> 1: NO PATH 2 -> 2: 0 2 -> 3: 4 3 -> 1: NO PATH 3 -> 2: NO PATH 3 -> 3: 0	✔
✔	3 4 1 2 9 2 3 4 1 3 4 2 1 -3	1 -> 1: 0 1 -> 2: 9 1 -> 3: 4 2 -> 1: -3 2 -> 2: 0 2 -> 3: 1 3 -> 1: NO PATH 3 -> 2: NO PATH 3 -> 3: 0	1 -> 1: 0 1 -> 2: 9 1 -> 3: 4 2 -> 1: -3 2 -> 2: 0 2 -> 3: 1 3 -> 1: NO PATH 3 -> 2: NO PATH 3 -> 3: 0	✔
✔	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	1 -> 1: 0 1 -> 2: 7 1 -> 3: 9 1 -> 4: 20 1 -> 5: 11 1 -> 6: 20 2 -> 1: NO PATH 2 -> 2: 0 2 -> 3: 10 2 -> 4: 15 2 -> 5: 12 2 -> 6: 21 3 -> 1: NO PATH 3 -> 2: NO PATH 3 -> 3: 0 3 -> 4: 11 3 -> 5: 2 3 -> 6: 11 4 -> 1: NO PATH 4 -> 2: NO PATH 4 -> 3: NO PATH 4 -> 4: 0 4 -> 5: NO PATH 4 -> 6: 6 5 -> 1: NO PATH 5 -> 2: NO PATH 5 -> 3: NO PATH 5 -> 4: NO PATH 5 -> 5: 0 5 -> 6: 9 6 -> 1: NO PATH 6 -> 2: NO PATH 6 -> 3: NO PATH 6 -> 4: NO PATH 6 -> 5: NO PATH 6 -> 6: 0	1 -> 1: 0 1 -> 2: 7 1 -> 3: 9 1 -> 4: 20 1 -> 5: 11 1 -> 6: 20 2 -> 1: NO PATH 2 -> 2: 0 2 -> 3: 10 2 -> 4: 15 2 -> 5: 12 2 -> 6: 21 3 -> 1: NO PATH 3 -> 2: NO PATH 3 -> 3: 0 3 -> 4: 11 3 -> 5: 2 3 -> 6: 11 4 -> 1: NO PATH 4 -> 2: NO PATH 4 -> 3: NO PATH 4 -> 4: 0 4 -> 5: NO PATH 4 -> 6: 6 5 -> 1: NO PATH 5 -> 2: NO PATH 5 -> 3: NO PATH 5 -> 4: NO PATH 5 -> 5: 0 5 -> 6: 9 6 -> 1: NO PATH 6 -> 2: NO PATH 6 -> 3: NO PATH 6 -> 4: NO PATH 6 -> 5: NO PATH 6 -> 6: 0	✔

	Input	Expected	Got	
✔	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	1 -> 1: 0 1 -> 2: 4 1 -> 3: 4 1 -> 4: 4 1 -> 5: 3 1 -> 6: 6 1 -> 7: 8 1 -> 8: 10 2 -> 1: NO PATH 2 -> 2: 0 2 -> 3: NO PATH 2 -> 4: NO PATH 2 -> 5: NO PATH 2 -> 6: NO PATH 2 -> 7: NO PATH 2 -> 8: NO PATH 3 -> 1: 3 3 -> 2: 5 3 -> 3: 0 3 -> 4: 0 3 -> 5: -1 3 -> 6: 2 3 -> 7: 4 3 -> 8: 6 4 -> 1: 3 4 -> 2: 7 4 -> 3: 2 4 -> 4: 0 4 -> 5: 1 4 -> 6: 4 4 -> 7: 6 4 -> 8: 8 5 -> 1: 4 5 -> 2: 8 5 -> 3: 3 5 -> 4: 1 5 -> 5: 0 5 -> 6: 5 5 -> 7: 5 5 -> 8: 7 6 -> 1: 1 6 -> 2: 3 6 -> 3: 0 6 -> 4: -2 6 -> 5: -3 6 -> 6: 0 6 -> 7: 2 6 -> 8: 4 7 -> 1: 3 7 -> 2: 5 7 -> 3: 2 7 -> 4: 0 7 -> 5: -1 7 -> 6: 2 7 -> 7: 0 7 -> 8: 2 8 -> 1: 2 8 -> 2: 6 8 -> 3: 1 8 -> 4: -1 8 -> 5: -2 8 -> 6: 3	1 -> 1: 0 1 -> 2: 4 1 -> 3: 4 1 -> 4: 4 1 -> 5: 3 1 -> 6: 6 1 -> 7: 8 1 -> 8: 10 2 -> 1: NO PATH 2 -> 2: 0 2 -> 3: NO PATH 2 -> 4: NO PATH 2 -> 5: NO PATH 2 -> 6: NO PATH 2 -> 7: NO PATH 2 -> 8: NO PATH 3 -> 1: 3 3 -> 2: 5 3 -> 3: 0 3 -> 4: 0 3 -> 5: -1 3 -> 6: 2 3 -> 7: 4 3 -> 8: 6 4 -> 1: 3 4 -> 2: 7 4 -> 3: 2 4 -> 4: 0 4 -> 5: 1 4 -> 6: 4 4 -> 7: 6 4 -> 8: 8 5 -> 1: 4 5 -> 2: 8 5 -> 3: 3 5 -> 4: 1 5 -> 5: 0 5 -> 6: 5 5 -> 7: 5 5 -> 8: 7 6 -> 1: 1 6 -> 2: 3 6 -> 3: 0 6 -> 4: -2 6 -> 5: -3 6 -> 6: 0 6 -> 7: 2 6 -> 8: 4 7 -> 1: 3 7 -> 2: 5 7 -> 3: 2 7 -> 4: 0 7 -> 5: -1 7 -> 6: 2 7 -> 7: 0 7 -> 8: 2 8 -> 1: 2 8 -> 2: 6 8 -> 3: 1 8 -> 4: -1 8 -> 5: -2 8 -> 6: 3	✔

	Input	Expected	Got	
		8 -> 7: 3 8 -> 8: 0	8 -> 7: 3 8 -> 8: 0	

Passed all tests! ✔

Correct

Marks for this submission: 1.00/1.00.

◄ Bài tập 8 - Extended traffic

Jump to...

Bài tập 10 - Thuật toán Floyd - Warshall (kiểm tra chu trình âm) ►