

```

#define NUM_NODE 6
#define INT_MAX 99999 // 무한대를 나타내는 값

int graph[NUM_NODE][NUM_NODE] = {
    {0, 4, INT_MAX, INT_MAX, INT_MAX, 10},
    {4, 0, 3, INT_MAX, INT_MAX, INT_MAX},
    {INT_MAX, 3, 0, 2, INT_MAX, INT_MAX},
    {INT_MAX, INT_MAX, 2, 0, 6, INT_MAX},
    {INT_MAX, INT_MAX, INT_MAX, 6, 0, 1},
    {10, INT_MAX, INT_MAX, INT_MAX, 1, 0}
};

```

요구 사항

- 모든 노드 간 최단 거리 계산을 위해 2차원 배열 `dist[NUM_NODE][NUM_NODE]` 를 사용하세요.
- `dist[i][j]` 는 노드 `i` 에서 노드 `j` 로의 최단 거리를 나타내며, 매트릭스 초기값은 `graph` 배열의 값으로 설정합니다.
- 각 노드 `k` 를 경유하여 `dist[i][j]` 가 더 짧아질 경우 경로를 갱신합니다. 즉, 다음 조건에 따라 업데이트하세요:

```

if (dist[i][k] + dist[k][j] < dist[i][j]) {
    dist[i][j] = dist[i][k] + dist[k][j];
}

```

출력 예시

프로그램을 실행하면 다음과 같은 형식으로 최단 거리 매트릭스를 출력해야 합니다.

```

0 4 7 9 8 9
4 0 3 5 6 7
7 3 0 2 3 4
9 5 2 0 6 3
8 6 3 6 0 1
9 7 4 3 1 0

```

위와 같이 각 노드 간의 최단 거리를 구하는 프로그램을 작성하세요.