

## 제11주 프로그래밍 과제

- 알라바마 주의 대략 15,000 곳의 지명과 경도 및 위도가 저장된 데이터파일이 주어진다.

```
Irvin Bar      -87.714440    31.386280    // 지명, 경도(longitude), 위도(latitude) 순이다.
Town of Kinsey -85.335400    31.292640
Nunn Branch   -86.974450    34.396210
Melvin Creek  -85.446320    31.402390
McCellan Cemetery -86.196360    33.562600
McCarley Cemetery -88.085600    34.399260
Shrine School -87.288060    33.842050
...
```

그리고 이 지점들 중에서 서로 도로로 연결되어 있는 지점 쌍들을 저장하는 파일이 추가로 주어진다.

```
Mary Ann Hollow Sullivan Ford // 도로로 연결된 두 지점의 명칭. Tab문자로 구분되어 있다.
Mary Ann Hollow  Vaughans Crossroad
Scout Creek      Highlands
Waller Creek     Wiley
Little Chilatchee Creek Parks Landing (historical)
Oak Bayou        Saint Paul United Methodist Church
Oak Bayou        Mudhole Creek
...
```

데이터 파일에서 필드들은 하나의 tab문자로 분리되어 있다. 이 파일에 등장하는 모든 지점들을 정점으로 하고 두 지점간에 도로가 있으면 에지로 연결하는 가중치 그래프의 인접리스트를 구성하라. 각 에지의 가중치는 두 지점간의 거리이다. 다음은 위도 경도 값으로 부터 거리를 계산해주는 함수이다. 지점간의 거리를 계산할 때 이 함수를 이용하라. 이 함수가 반환하는 거리의 단위는 미터(m)이다. C언어로 구현할 경우에는 `math.h`를 include하고 파이( $\pi$ ) 값은 다음과 같이 정의하면 된다.

```
#define PI acos(-1.0)
// 혹은
#define PI 3.14159265358979323846

// 매개변수는 첫번째 지점의 위도(lat1), 경도(lon1), 두번째 지점의 위도(lat2), 경도(lon2) 순서이다.
public double calDistance(double lat1, double lon1, double lat2, double lon2) {
    double theta, dist;
    theta = lon1 - lon2;
    dist = Math.sin(deg2rad(lat1)) * Math.sin(deg2rad(lat2))
        + Math.cos(deg2rad(lat1))
        * Math.cos(deg2rad(lat2)) * Math.cos(deg2rad(theta));
    dist = Math.acos(dist);
    dist = rad2deg(dist);
    dist = dist * 60 * 1.1515;
    dist = dist * 1.609344; // 단위 mile 에서 km 변환.
    dist = dist * 1000.0; // 단위 km 에서 m 로 변환
    return dist;
}

// 주어진 도(degree) 값을 라디안으로 변환
private double deg2rad(double deg) {
    return (double)(deg * Math.PI / (double)180);
}

// 주어진 라디안(radian) 값을 도(degree) 값으로 변환
private double rad2deg(double rad) {
    return (double)(rad * (double)180 / Math.PI);
}
}
```

- 인접 리스트를 구성한 후 임의의 한 지점의 이름을 입력으로 받아서 그 지점으로부터 10 홉(hop) 이내인 모든 지점의 이름과 위도, 경도를 화면에 출력하는 기능을 추가하라. 여기서 10홉 이내라는 말은 10개 이내의 에지를 거쳐서 도달할 수 있다는 의미이다.
- 임의의 한 지점의 이름을 입력으로 받아서 그 지점에서 출발하여 그래프를 DFS로 순회하면서 모든 지점의 이름, 위도, 경도를 파일로 출력하는 기능을 구현하라.
- Dijkstra의 알고리즘으로 최단 경로를 찾는 프로그램을 구현하라. 입력으로 두 지점의 명칭을 받아서 두 지점 간의 최단경로를 구하여 그 경로 상의 지명을 순서대로 출력하고, 그 경로의 길이도 출력하라.