

# Week 14

- 주의사항: 부정행위금지 (채점 서버 외 인터넷 사용 금지), STL 사용금지(vector는 사용 가능)
- 표준 입출력 사용을 권장(C는 scanf / printf, C++은 cin / cout)

## 문제 1

데이터를 입력 받아 인접리스트표현법(Adjacency List Representation) 기반의 무향그래프를 생성하는 프로그램을 작성하시오. 주어지는 명령어는 다음과 같이 7가지다.

- **Graph GID** : 그래프 ID 가 정수 **GID** ( $10 \leq \text{GID} < 100$ )인 그래프를 생성한다. 같은 **GID**를 갖는 그래프는 존재하지 않음이 보장된다.
- **InsertVertex GID S** : 그래프 **GID**에 정점 ID 가 정수 **S** ( $100,000 \leq S < 1,000,000$ )인 정점을 추가한다. **GID**는 항상 존재하는 그래프로 주어진다. 만약 그래프 **GID**에 정점 **S**가 이미 존재한다면, "ERRORCODE 01"을 출력하고, 해당 명령어를 무시한다.
- **InsertEdge GID S D** : 그래프 **GID**에 서로 다른 두 정점 **S**, **D**를 잇는 간선 (**S,D**)를 추가한다. **GID**는 항상 존재하는 그래프로 주어진다. 만약 그래프에 **GID**에 정점 **S** 또는 **D**가 존재하지 않는다면, "ERRORCODE 03"을 출력하고, 해당 명령어를 무시한다. 만약 그래프 **GID**에 간선 (**S,D**)가 이미 존재한다면, "ERRORCODE 02"를 출력하고, 해당 명령어를 무시한다.
- **EraseVertex GID S** : 그래프 **GID**에서 정점 **S**를 삭제한다. **GID**는 항상 존재하는 그래프에 대해서만 주어진다. 만약 그래프 **GID**에 정점 **S**가 존재하지 않는다면, "ERRORCODE 03"을 출력하고, 해당 명령어를 무시한다.
- **EraseEdge GID S D** : 그래프 **GID**에서 서로 다른 두 정점 **S**, **D**를 잇는 간선 (**S,D**)를 삭제한다. **GID**는 항상 존재하는 그래프로 주어진다. 만약 그래프 **GID**에 정점 **S** 또는 **D**가 존재하지 않는다면, "ERRORCODE 03"을 출력하고, 해당 명령어를 무시한다. 만약 그래프 **GID**에 간선 (**S,D**)가 존재하지 않는다면, "ERRORCODE 04"를 출력하고, 해당 명령어를 무시한다.
- **PrintAdjSum GID S** : 그래프 **GID**에서 정점 **S**의 인접한 정점들의 수와 ID의 합을 공백으로 구분하여 출력한다. **GID**는 항상 존재하는 그래프로 주어진다. 만약 그래프 **GID**에 정점 **S**가 존재하지 않는다면, "ERRORCODE 03"을 출력하고, 해당 명령어를 무시한다. 또는 해당 그래프에서 인접한 정점이 존재하지 않는다면, "0 0"을 출력한다.
- **PrintAdjMin GID S** : 그래프 **GID**에서 정점 **S**의 인접한 정점들 중 최소 ID를 출력한다. **GID**는 항상 존재하는 그래프로 주어진다. 만약 그래프 **GID**에 정점 **S**가 존재하지 않는다면,

"ERRORCODE 03"을 출력하고, 해당 명령어를 무시한다. 또는 해당 그래프에서 인접한 정점이 존재하지 않는다면, "-1"을 출력한다.

## 입력

첫 번째 줄에 명령어의 개수  $N$  ( $1 \leq N \leq 10,000$ )이 주어진다.

두 번째 줄부터  $N$ 개의 줄에 걸쳐 명령어가 한 줄에 하나씩 주어진다.

## 출력

출력이 필요한 명령어가 주어질 때마다 그 결과를 한 줄에 하나씩 출력한다.

### 예제 입출력

| 예제 입력                       | 예제 출력        |
|-----------------------------|--------------|
| 22                          | ERRORCODE 01 |
| Graph 10                    | ERRORCODE 03 |
| Graph 11                    | 2 200005     |
| InsertVertex 10 100001      | 0 0          |
| InsertVertex 10 100002      | ERRORCODE 02 |
| InsertEdge 10 100001 100002 | ERRORCODE 03 |
| InsertVertex 11 100001      | ERRORCODE 03 |
| InsertVertex 10 100001      | 100005       |
| InsertEdge 11 100001 100002 | -1           |
| InsertVertex 10 100003      |              |
| InsertEdge 10 100001 100003 |              |
| PrintAdjSum 10 100001       |              |
| PrintAdjSum 11 100001       |              |
| InsertEdge 10 100001 100002 |              |
| InsertEdge 10 100003 100004 |              |
| EraseVertex 10 100002       |              |
| InsertEdge 10 100001 100002 |              |
| InsertVertex 10 100004      |              |
| InsertVertex 10 100005      |              |
| InsertEdge 10 100004 100005 |              |
| InsertEdge 10 100005 100001 |              |
| PrintAdjMin 10 100004       |              |
| PrintAdjMin 11 100001       |              |