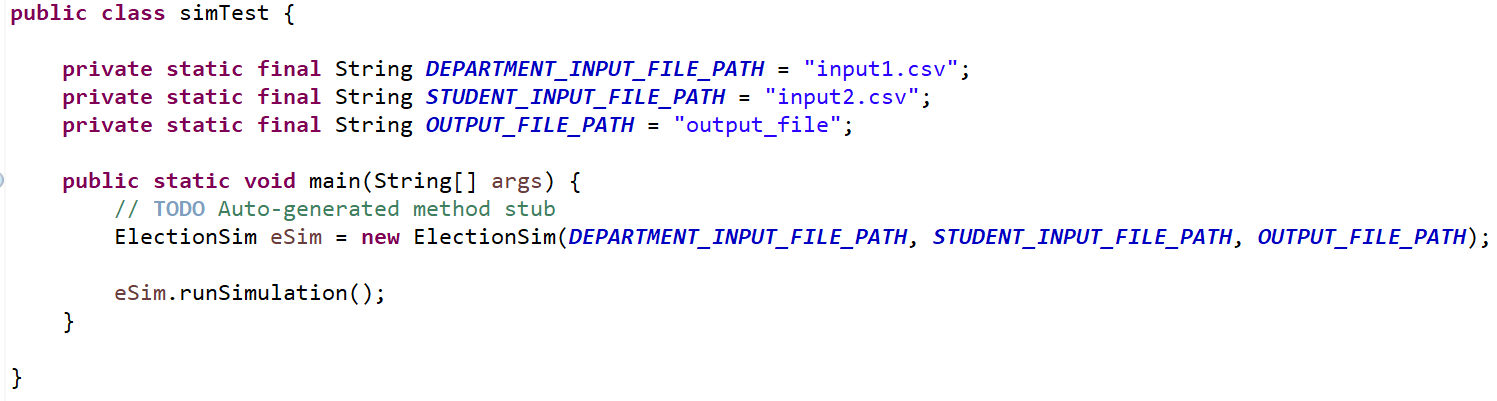
# **Programming Assignment #2 Election Simulation**

2017029807 컴퓨터소프트웨어학부 성창호

본 과제는 각 학과의 대표를 뽑기 위한 가상 선거 시뮬레이션 프로그램으로, 2개의 Input 파일과 simTest, ElectionSim, Student, Candidate and Department 클래스를 이용하여 결과를 출력한다.

**1. simTest Class**



simTest 클래스는 main 함수를 포함하고 있는 클래스로 Input과 Output 파일의 경로와 이름을 설정하고 가상 시뮬레이션 클래스와 메서드인 eSim과 runSimulation를 실행한다.

**2. ElectionSim Class**

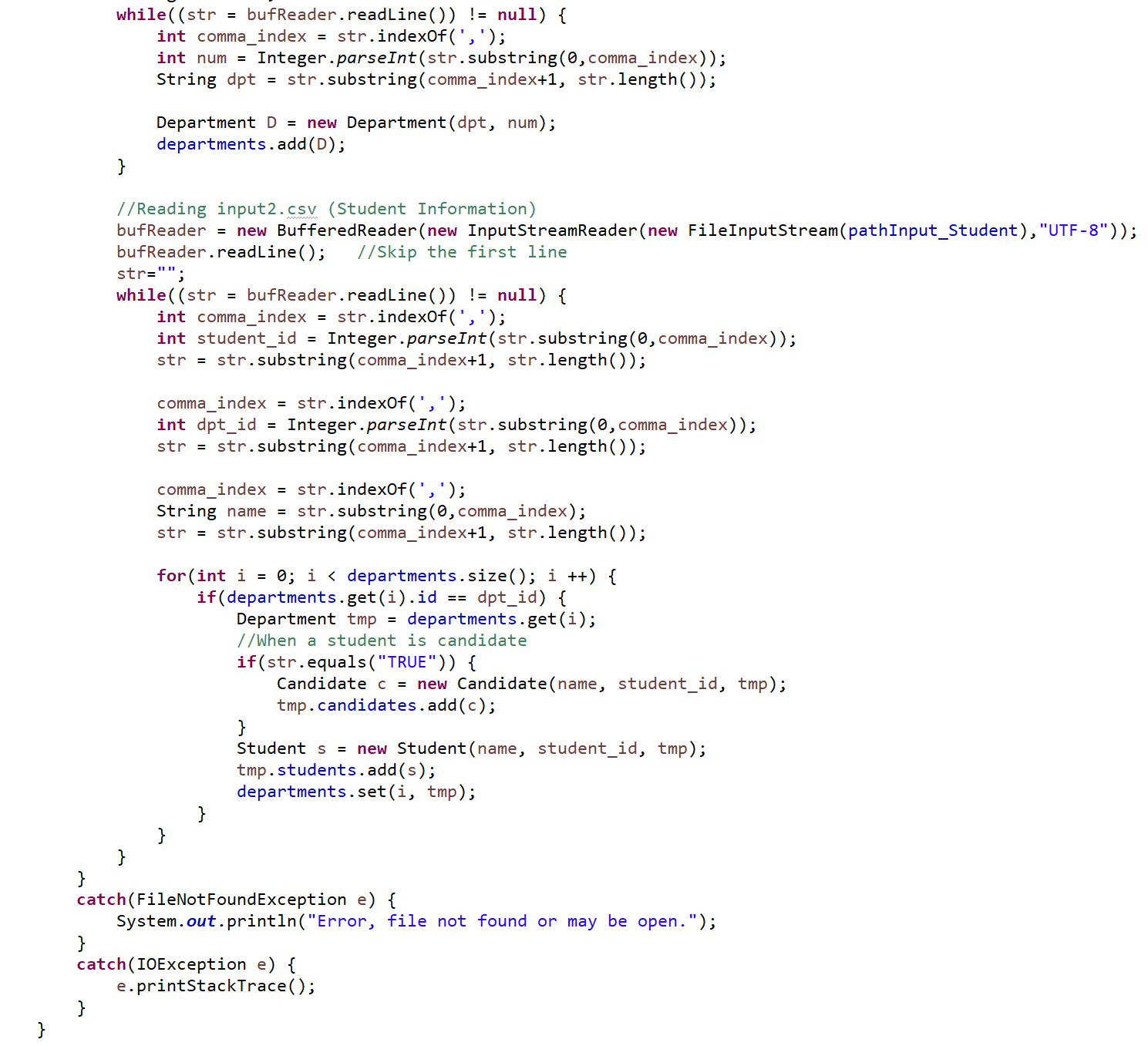


①

④

③

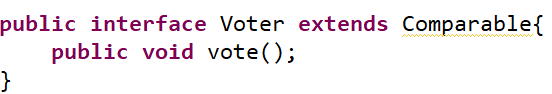
②



ElectionSim 클래스는 학과의 정보와 학생들의 정보를 input1.csv, input2.csv 파일을 통해 입력 받고 Student, Candidate, Department 객체를 생성하여 저장한다.

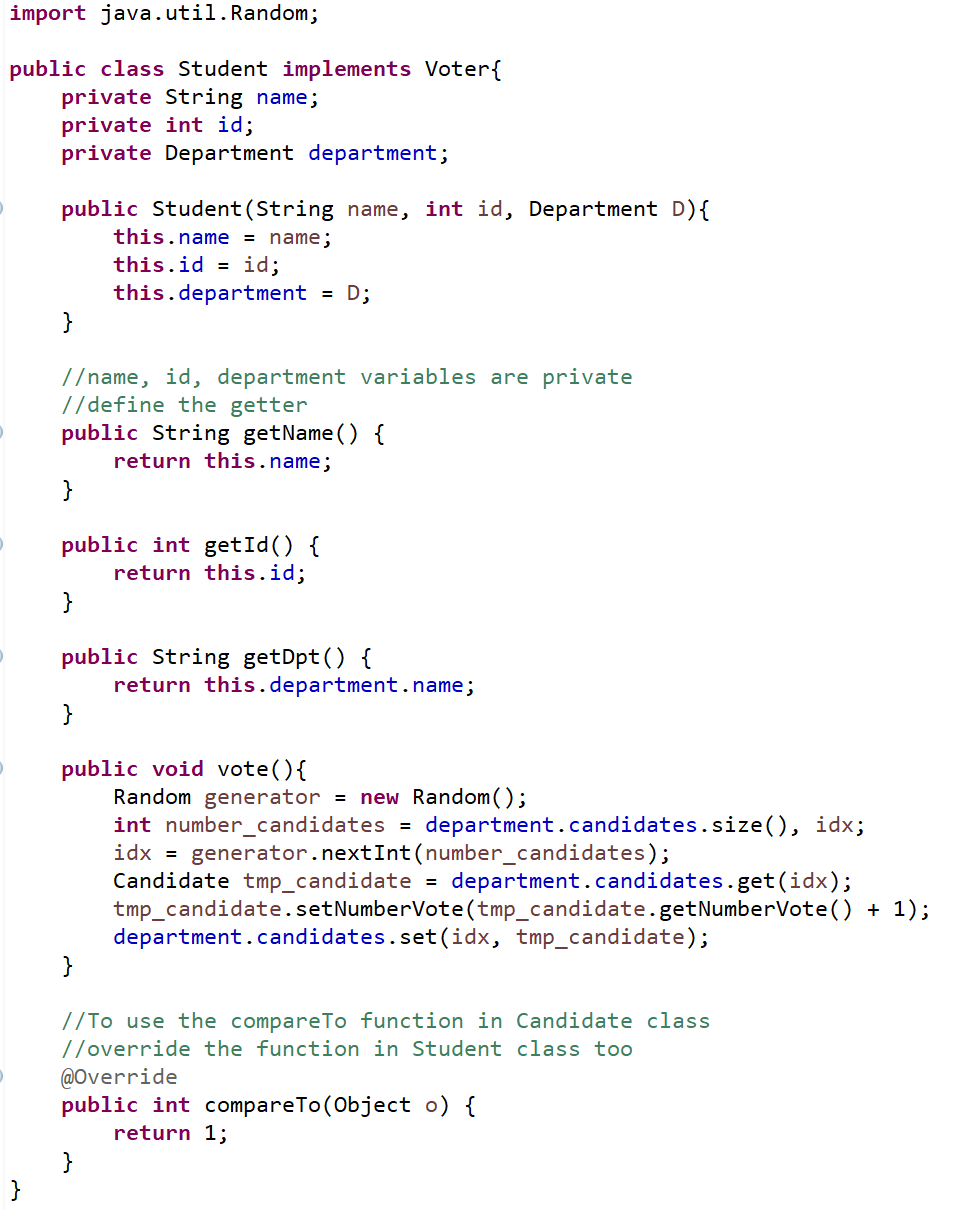
1. Constructor
   1. input1.csv 파일에서 학과의 번호와 이름을 Buffer와 Stream를 이용하여 readLine 메서드로 입력받고, indexOf 메서드를 이용하여 쉼표(,)를 기준으로 번호와 이름을 분리한다. Department 객체인 D를 생성하고 ArrayList인 departments에 추가한다.
   2. input2.csv 파일에서 학생 정보를 Buffer와 Stream를 이용하여 readLine 메서드로 입력받고, indexOf 메서드를 이용하여 쉼표(,)를 기준으로 학생 번호, 학과 번호, 이름과 후보 여부를 분리한다. for문을 이용하여 입력 받은 학생의 학과 번호와 departments ArrayList에 있는 학과 번호가 일치하는지 확인한 후, 후보 여부에 따라 해당 학과의 candidates와 student ArrayList에 추가한다.
2. saveData Method
3. 각 학과의 최종 당선된 후보자가 저장되어 있는 candidates ArrayList를 득표 수에 따라 정렬한다. 그리고 PrintWriter와 OutputStream을 이용하여 for-each문을 통해 각 학과의 이름과 최종 당선인의 id, 득표 수를 출력한다.
4. runSimulation Method
5. 각 학과는 for-each문을 이용하여 참조하고, 각 학과 d에 대해 iterator를 이용하여 students ArrayList의 객체의 vote 메서드를 호출한다. 그리고 candidate ArrayList에 학과 d의 최종 당선인을 getMostVote\_Candidate 메서드를 호출하여 구한다.

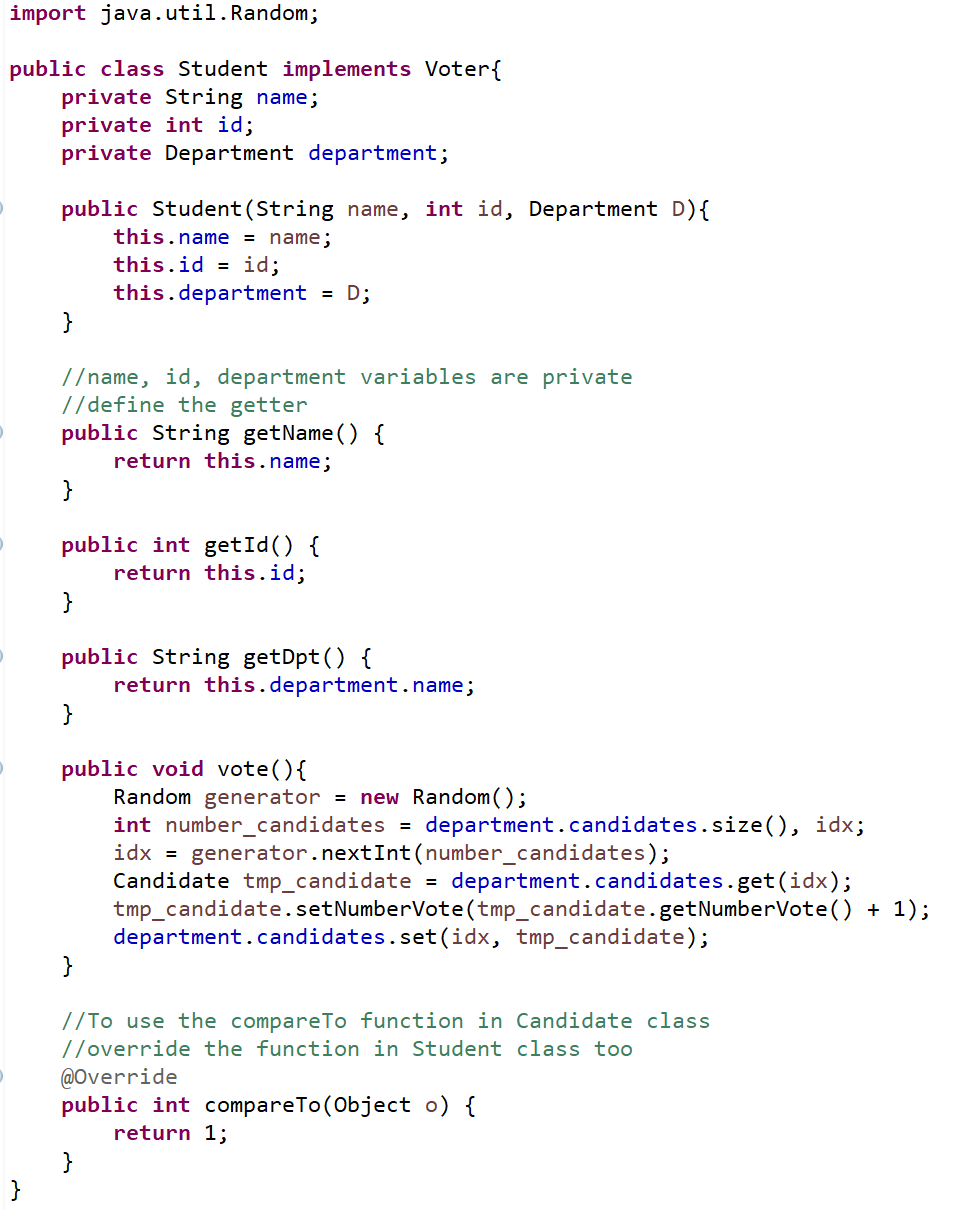
**3. Voter Interface**



Voter 인터페이스는 Comparable 인터페이스를 extends하고, vote 메서드를 선언한다.

**4. Student Class**



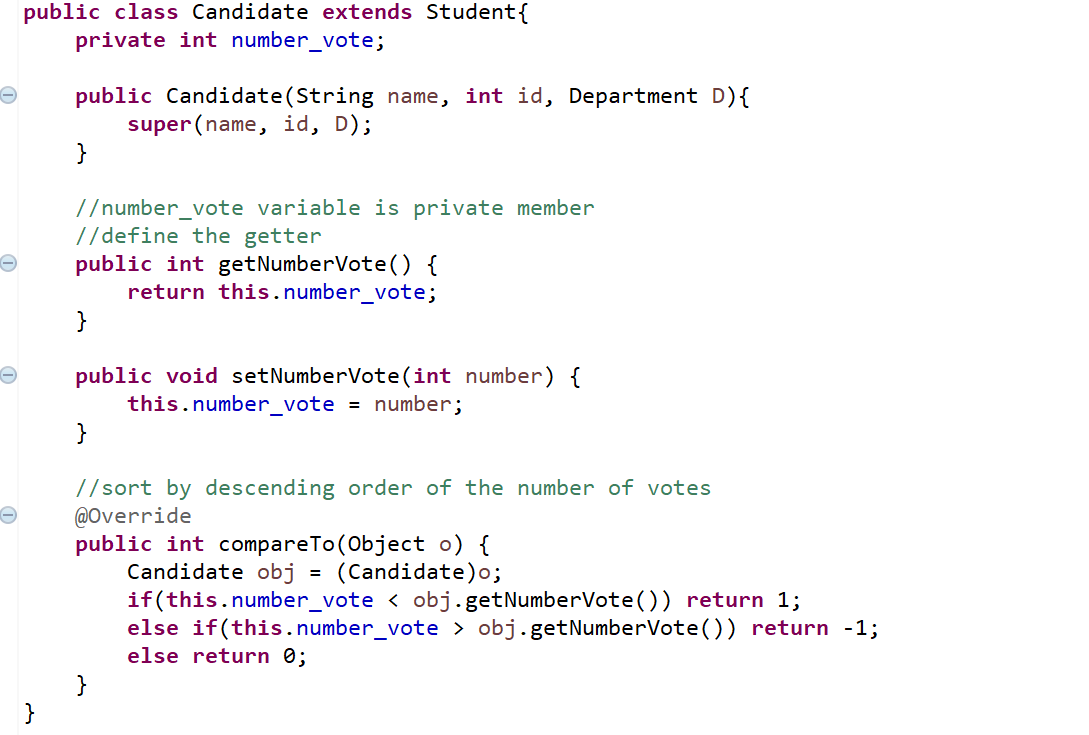


Student 클래스는 Voter 인터페이스를 implements한다. 학생의 정보를 저장하는 클래스이므로 이름과 id, 학과를 private member로 선언한다. Student의 member들이 private member이기 때문에 외부에서 접근하기 위해서는 getter 메서드가 필요하다.

vote 메서드는 소속 학과의 후보 수보다 작은 수를 랜덤하게 설정하여 해당 번호의 후보자의 득표 수를 증가시킨다.

compareTo 메서드는 implements한 Voter 인터페이스에 extends된 Comparable 인터페이스의 것으로 override되고, 후보자의 득표 수를 비교할 것이기 때문에 Candidate 클래스에서도 override 된다.

**5. Candidate Class**

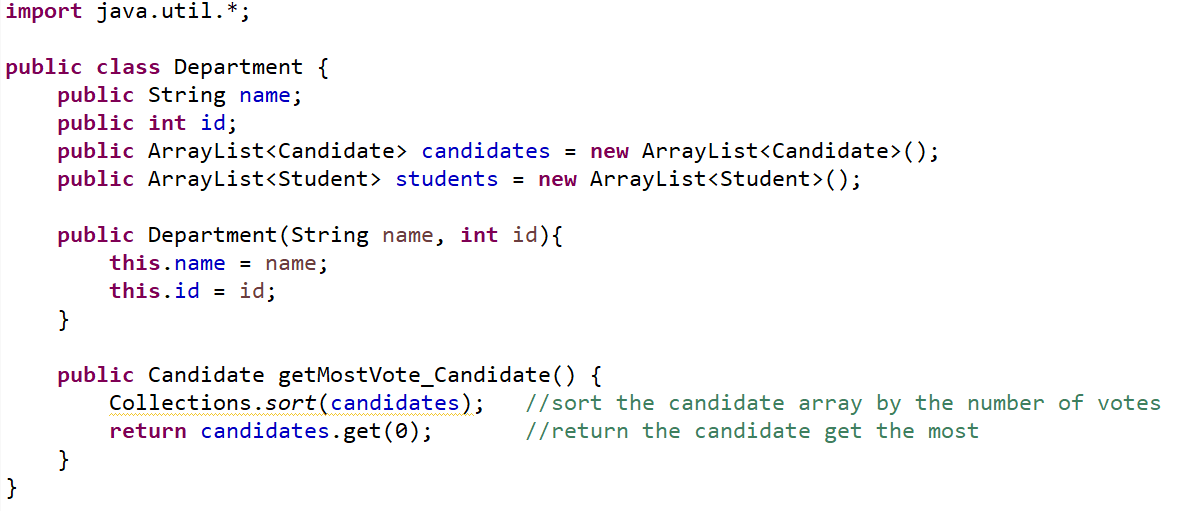


Candidate 클래스는 Student 클래스를 extends한다. 학생이자 후보자이고, 득표 수를 저장하는 클래스이므로 득표 수를 private member로 선언한다. number\_vote이 private member이므로 외부에서 접근하기 위해서는 getter 메서드가 필요하다.

Student 클래스의 vote 메서드에서 후보자의 득표 수를 저장하는 number\_vote를 변경해야 하므로 setter 메서드를 정의한다.

compareTo 메서드는 득표 수에 따라 후보자를 정렬해야 하므로 인수로 받은 o를 Candidate 객체로 casting하여 getNumberVote 메서드를 통해 비교한다.

**6. Department Class**



Department 클래스는 학과의 정보를 저장하는 클래스로, 학과의 이름, id, 소속 학생들과 후보자들의 ArrayList를 member로 가지고 있다.

getMostVote\_Candidate 메서드는 해당 학과의 최종 당선인을 반환하는 메서드이다. sort 메서드를 이용하여 해당 학과의 후보자들을 저장한 candidates ArrayList를 득표 수를 기준으로 내림차순 정렬하고, 정렬 후 가장 맨 앞에 있는, 즉 0의 index를 가지는 candidate 객체를 반환한다.

**\* Program’s Output**

1. Input File : input1.csv, inptu2.csv (sample input)

2. Result

