

## 감염자 관리 프로그램 만들기

### 1. 문제 정의

- 미래에 발병한 감염병 COVID-59에 대해, 감염자(환자)정보 및 감염 경로를 관리하는 프로그램 구현
- 2222년에는 교통 기술의 발전으로 아래 40개 도시 간에는 하루만에 이동이 가능하게 되었으며, 모든 사람들은 매일 다른 도시로 이동

Seoul, Jeju, Tokyo, LosAngeles, NewYork,  
Texas, Toronto, Paris, Nice, Rome,  
Milan, London, Manchester, Basel, Luzern,  
Munich, Frankfurt, Berlin, Barcelona, Madrid,  
Amsterdam, Stockholm, Oslo, Hanoi, Bangkok,  
KualaLumpur, Singapore, Sydney, SaoPaulo, Cairo,  
Beijing, Nairobi, Cancun, BuenosAires, Reykjavik,  
Glasgow, Warsaw, Istanbul, Dubai, CapeTown

- 2222년에는 감염병 역시 발전을 거듭하면서 전염력이 매우 강한 COVID-59 감염병이 창궐하게 됨
- COVID-59에 대비하기 위해 환자 정보를 관리하고 감염 경로를 파악하기 위한 프로그램이 필요하게 됨
- 환자 정보는 파일로부터 읽어오며, 파일은 아래와 같은 형식으로 정보를 저장

(환자 번호) (나이) (감염 확인일자) (확인4일전 장소) (3일전 장소) (2일전 장소) (하루 전 장소) (확인 당시 장소)

- 모든 정보들은 정수 형태로 저장되어 있음
- 환자 번호는 환자를 식별하기 위해 매겨진 숫자로 0부터 시작
- 감염 확인일자는 환자의 발병이 확인된 시점을 뜻하며, 창궐 후 N일째 형태의 정수로 표현
- 장소는 ifct\_element.c 내 정의된 enum을 기반으로 한 정수로 표현
- 감염자 관리 프로그램은 아래 기능을 제공해야 함

1. 특정 환자에 대한 정보 출력
2. 특정 장소에서 감염이 확인된 환자 관련 정보 출력
3. 특정 범위의 나이에 해당하는 환자 관련 정보 출력
4. 감염 경로 및 최초 전파자 추적
0. 프로그램 종료

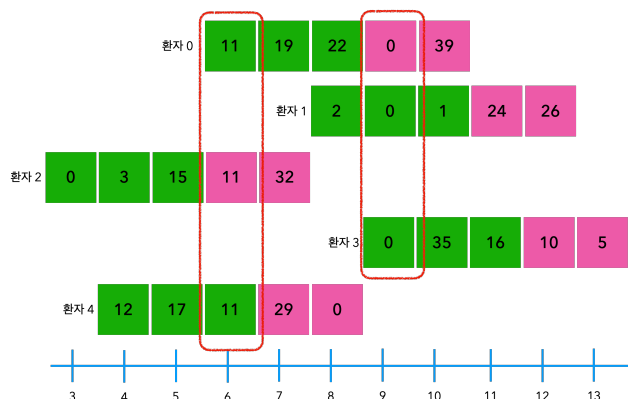
- 프로그램 시작과 함께 파일로부터 환자 정보를 읽어와서 저장한 뒤에 위의 기능을 제공
  - 실행 파라미터를 통해 파일 경로 및 이름을 받아와서 파일을 읽음
  - 주어진 linked list 소스 코드를 활용하여 저장
- 각 기능에 대해 세부적으로 아래를 만족해야 함
  - 1 : 지정된 환자(번호 입력)에 대한 정보 출력은 번호, 나이, 감염 확인일자, 그리고 최근 5개 이동장소를 출력
  - 2 : 지정된 장소(문자열 형태로 입력)에서 발병 확인이 된 환자 모두의 정보 출력
  - 3 : 특정 범위 나이는 최소 및 최대 값을 입력 받으며, 최소값 이상이면서 최대값 이하 나이의 환자 모두의 정보 출력
  - 4 : 지정된 환자를 시작으로 전파자와 감염당한 시점 및 장소를 순차적 출력하고 최초 전파자를 최종적으로 출력
- 감염 경로 추적 시 감염병과 관련된 아래 가정을 활용
  - 전염력이 강력하여 전염 가능시기의 감염자와 같은 시점 같은 도시에 있으면 무조건 감염됨
  - 감염되고 2-4일 만에 반드시 발병하며, 발병 즉시 그 시점 그 장소에서 확인됨

- 발병 당일 혹은 전날에만 다른 사람에게 전염시킬 수 있음
  - 전파자가 여러명 존재하면 가장 이른 시점에 감염시킨 전파자를 추적
- 주어진 base code의 구조 하에서 구현하며, 각 소스코드파일은 아래와 같은 역할을 가짐
- ifct\_element.c : 감염자 정보 구조체를 다루며, 구조체 instance 생성과 내용 분석 역할을 가짐
  - ifct\_database.c : 감염자 정보들을 모아서 관리하며, linked list 형태의 자료 구조로 저장
  - main.c : 프로그램의 주요 동작 흐름 실현
- Data encapsulation 개념으로 구현하며, 각 소스코드파일의 역할
- 소스코드파일 간 전역 변수 공유 사용 금지 (즉, extern 문법 사용 금지)
  - ifct\_ele\_t 구조체 정의는 ifct\_element.c에서 이루어지며, 이 파일 내에서만 사용 가능
  - linked list는 ifct\_database.c 내에서만 접근 가능하며 다른 파일에서는 ifct\_database.h 내 prototyping된 함수를 통해서만 접근 가능
  - 표준 입출력은 main.c에서만 사용 (즉, 키보드 키보드 및 모니터 출력은 main.c에서만 다루어야 함)
  - main.c에서만 ifct\_database.h 및 ifct\_element.h를 include할 수 있음  
(ifct\_database.c에서 ifct\_element.h를 include하거나 ifct\_element.c에서 ifct\_database.h를 include하면 안 됨)

- 환자 정보 예시 : patientInfo\_sample.txt
- 5명의 환자 정보가 존재하며 아래와 같은 내용임

번호	나이	확인 시점	이동 장소				
0	36	10	London (11)	Madrid (19)	Oslo (22)	Seoul (0)	CapeTown (39)
1	51	12	Tokyo (2)	Seoul (0)	Jeju (1)	Bangkok (24)	Singapore (26)
2	27	7	Seoul (0)	LA (3)	Munich (15)	London (11)	Cancun (32)
3	23	13	Seoul (0)	Glasgow (35)	Frankfurt (16)	Milan (10)	Texas (5)
4	44	8	Manchester (12)	Berlin (17)	London (11)	Cairo (29)	Seoul (0)

- 시점을 기준으로 환자들의 이동 경로를 도식화하면 아래와 같음



- 따라서 전파자 추적을 해보면 다음과 같음
  - ▶ 0번 -> 2번 (시점 6, London)
  - ▶ 1번 -> 0번 (시점 9, Seoul) -> 2번 (시점 6, London)
  - ▶ 2번 (최초 전파자)
  - ▶ 3번 -> 0번 (시점 9, Seoul) -> 2번 (시점 6, London)
  - ▶ 4번 -> 2번 (시점 6, London)

---

## 2. 코딩 방향

- Basecode로 주어진 main.c 및 ifct\_element.c 파일을 채움
  - ifct\_element.c 내에 환자 정보 구조체 생성 및 구조체 내부 변수를 접근할 수 있는 함수 구현
  - ifct\_element.c 내에 환자 정보를 출력하는 함수 구현
  - main.c 내에 파일 입출력을 통해 환자 정보 구조체 생성 및 linked list 저장 코드 구현 (ifct\_database.h 함수 활용)
  - main.c 내에 특정 조건에 맞는 환자의 정보 출력 코드 구현
  - main.c 내에 환자의 전파자를 추적하고 최초 전파자를 알아내는 코드 구현
- Base code로 주어진 ifct\_database.c는 있는 그대로 활용하며, 제공된 함수를 header include를 통해 활용
- 함수/변수에 대한 이름을 잘 정의하고, 각종 정의 및 코드 흐름에 대해 다른 사람들이 알아볼 수 있도록 주석 삽입
- 코드를 보기 쉽게 들여쓰기를 적절히 삽입
- 기능 혹은 일부 코드를 구현하거나 디버깅 과정에서 코드를 수정할 때마다 Github에 올려서 변화를 추적할 수 있도록 함
- 본 구현에 필요한 입출력과 문자열, 구조체, 알고리즘 등의 요소들은 추후 관련 이론을 배우는 주차에 실습으로 다룰 예정임

---

## 3. 채점 기준

- 프로그램 동작 점수 : 100점
  - 문제 정의에 명시된 기능들이 충실하게 잘 동작하는 지 여부
  - 주어진 감염자 정보 파일 5개를 실제로 돌려서 각 기능에 대한 동작 여부 확인
  - 정상적으로 동작하면 만점, 부분적으로 동작하면 구현 코드 및 표면적인 동작 상태에 따라 부분점수 부여
  - 배점 : 1-3번 기능 - 각 20점, 4번 - 40점
- 코드 관리 점수 : 50점
  - Data encapsulation 개념으로 잘 구현되어있는가
  - 변수 및 함수 이름을 잘 지정했는가
  - 주석을 충실하게 달았는가 (함수 및 변수 정의에 대한 설명 및 코드 흐름에 대한 설명)
  - 들여쓰기를 잘했는가
  - Github에 코드를 단계적으로 구현한 이력이 남아있는가
- **copy 여부에 대해 집중적으로 볼 예정이며, 적발 시 관련자 모두 0점으로 처리**

---

## 4. 결과물 제출 방법

- Github 계정에 별도의 repository로 올림
  - infectionPath라는 이름으로 repository 생성
  - 제출 기한 시간 직전에 올라간 소스코드 파일들을 기준으로 채점
  - Github에 infectionPath라는 repository가 없으면 제출하지 않은 것으로 간주할 예정이므로, 제출 후 웹 페이지 상에서 해당 repository가 보이는지 반드시 확인 요망
  - 가급적 미리 올리거나 지속적으로 여러번 submit을 하는 형태로 진행 요망

**- 제출 기한 : 12월 14일 수요일 23:59까지**

