

알고리즘 설계과제 1

[환자 관리 프로그램 설계]

1. 과제 개요

인하병원에서는 레드블랙트리(red-black tree)를 이용하여 환자들의 정보를 효율적으로 관리할 수 있는 프로그램을 설계하려 한다. 이 프로그램은 환자 관리를 위한 몇 가지 기능들을 수행한다. 각 기능에 대한 정보를 표준입력으로 입력받은 뒤, 수행한 결과를 표준출력으로 출력하는 프로그램을 구현하시오.

2. 환자정보 형식

- ① 환자번호 (기준키, 유일함): 1,000,000 ~ 1,999,999 사이의 정수.
- ② 이름: 공백 없는 20 bytes 이내의 문자열. 예) "Mary"
- ③ 연락처: 회원의 전화번호(공백 없는 11 bytes 이내의 문자열). 예) "01012345678"
- ④ 주소 좌표 (x, y): x와 y는 각각 0 ~ 10,000 사이의 임의의 정수.
- ⑤ 진료기록: 아래 정보들의 나열
 - 병명: 공백 없는 20 bytes 이내의 문자열. 예) "Headache"
 - 진료비: 10,000 ~ 100,000 사이의 정수

각 줄에 ①~⑤의 정보들이 차례로 공백으로 구분되어 저장된다.

필드 ①	필드 ②	필드 ③	필드 ④		필드 ⑤
환자번호	이름	연락처	주소의 x 좌표	주소의 y 좌표	진료기록

진료기록은 다음의 정보들을 원소로 저장한다.

진료기록의 원소	
병명	진료비

3. 수행할 기능

(1) 신규 가입 (최초 진료)

- 질의형식: "I K N H Ax Ay DI C"

I: 환자를 신규 가입하는 질의를 나타내는 기호

K: 환자번호

N: 이름

H: 연락처

Ax: 주소의 x 좌표

Ay: 주소의 y 좌표

DI: 병명

C: 진료비

- 출력형식: "D S"

D: 트리에서 환자 정보가 저장된 노드의 깊이 (root의 깊이는 0)

S: 신규 가입 처리를 나타내는 정수(0 또는 1) (0: 거절, 1: 승인)

- 설명: 새로운 환자 정보를 입력받아 레드블랙트리에 노드를 삽입하고, 그 노드의 깊이를 출력한다. 만약 동일한 환자번호가 이미 존재한다면, 그 노드의 깊이를 출력하고 신규 가입은 거절된다.

(2) 환자 검색

- 질의형식: “F K”
 - F: 특정 환자를 검색하는 질의를 나타내는 기호
 - K: 환자번호
- 출력형식: “D N H Ax Ay” 또는 “Not found”
 - D: 트리에서 환자 정보가 저장된 노드의 깊이
 - N: 이름
 - H: 연락처
 - Ax: 주소의 x 좌표
 - Ay: 주소의 y 좌표
- 설명: 해당 환자를 탐색하여 존재하면 환자의 정보를 출력한다. 만약 존재하지 않으면, “Not found”(주의: 대·소문자, 띄어쓰기 등 정확히 일치시킬 것)를 출력한다.

(3) 추가 진료

- 질의형식: “A K DI C”
 - A: 특정 환자에 대해 새로운 진료 정보를 추가하는 질의를 나타내는 기호
 - K: 환자번호
 - DI: 병명
 - C: 진료비
- 출력형식: “D” 또는 “Not found”
 - D: 트리에서 환자 정보가 저장된 노드의 깊이
- 설명: 해당 환자를 탐색하여 입력받은 진료 내용을 진료기록에 추가하고, 환자에 대한 노드의 깊이를 출력한다. 만약 환자가 존재하지 않는 경우, “Not found”를 출력한다.

(4) 유행병 조사

- 질의형식 : “E DI”
 - E: 마지막에 진료받은 병명을 기준으로 특정 병명을 수를 집계하는 질의를 나타내는 기호
 - DI: 병명
- 출력형식: “T”
 - T: 모든 환자들의 진료기록에서 마지막으로 추가된 병명이 입력으로 주어진 병명과 같은 환자의 수($T \geq 0$)
- 설명: 트리에 저장된 모든 환자들에 대해, 마지막으로 진단받은 병명이 입력으로 주어진 병명과 동일하면 집계하고, 집계된 수를 출력하여 유행성의 정도를 파악한다. 편의상, 골절 같은 전염성이 없는 병명도 유행병으로 간주한다.

4. 입출력 제한사항

- (1) 질의는 최대 100,000개가 입력된다.
- (2) 전체 질의에 대해 2초의 제한시간 이내에 수행되어야 한다.
- (3) 제시한 입출력 형식대로 표준 입출력을 사용하여 처리한다.
- (4) 문제에서 설명되지 않은 예외처리할 질의는 입력되지 않는다.

5. 프로그램 입출력 예

(1) 표준입출력 예시

첫 번째 줄은 질의의 수가 주어지고, 이후 각 줄에는 질의 정보가 주어진다.

파란색: 프로그램 표준입력 내용

빨간색: 프로그램 표준출력 내용

```
=====
16
I 1005691 Mary 01012345678 1123 90 Pneumonia 50000
0 1
I 1024129 Dorothy 01014832345 3453 6660 Diabetes 10000
1 1
I 1009711 Frank 01090123141 5453 5678 Fracture 10000
0 1
I 1008353 Athur 01065461752 23 2365 Measles 10000
2 1
I 1012317 Anna 01048713158 111 2234 Flu 100000
2 1
I 1014748 Edward 01097123455 3245 1234 Bruise 10000
1 1
I 1011062 Nancy 01078954184 766 445 Fracture 10000
3 1
I 1028522 Henry 01015648964 4346 6567 Fracture 10000
3 1
F 1005691
1 Mary 01012345678 1123 90
F 1003200
Not found
I 1014748 Susan 01093223455 322 124 Fracture 10000
1 0
E Fracture
3
E Headache
0
A 1008000 Pneumonia 30000
Not found
A 1011062 Pneumonia 30000
3
E Fracture
2
(프로그램 종료)
```

(2) 표준입출력 예시에서의 각 입/출력 내용 (구현을 검증할 때, 활용할 것)

Input
16 I 1005691 Mary 01012345678 1123 90 Pneumonia 50000 I 1024129 Dorothy 01014832345 3453 6660 Diabetes 10000 I 1009711 Frank 01090123141 5453 5678 Fracture 10000 I 1008353 Athur 01065461752 23 2365 Measles 10000 I 1012317 Anna 01048713158 111 2234 Flu 100000 I 1014748 Edward 01097123455 3245 1234 Bruise 10000 I 1011062 Nancy 01078954184 766 445 Fracture 10000 I 1028522 Henry 01015648964 4346 6567 Fracture 10000 F 1005691 F 1003200 I 1014748 Susan 01093223455 322 124 Fracture 10000 E Fracture E Headache A 1008000 Pneumonia 30000 A 1011062 Pneumonia 30000 E Fracture

Output
0 1 1 1 0 1 2 1 2 1 1 1 3 1 3 1 1 Mary 01012345678 1123 90 Not found 1 0 3 0 Not found 3 2

6. 주의 사항 (지키지 않으면, 0점 또는 감점)

(1) 개발언어 및 채점서버 환경

- ① 허용된 개발언어: C, C++
- ② OS: Ubuntu 18.04 (64-bit)
- ③ gcc 버전: gcc (Ubuntu 7.3.0-16ubuntu3) 7.3.0 (c++14 지원)

(2) 제출 파일

- ① 보고서 (I-class에 제출)
 - (a) 파일형식: 아래아한글 문서(.hwp), MS Word 문서(.doc, .docx), PDF 문서(.pdf)
 - (b) 양식: 첨부된 파일 참조
 - (c) 파일명: "학번_이름"으로 통일 (예: 12059876_홍길동)
 - ② 소스코드 (채점서버에 제출)
 - (a) 하나의 소스코드 파일로 구현하여 서버에 제출 및 채점 (예: 12059876_GildongHong.cpp)
(채점서버에서 파일명에 대한 한글 인식이 되지 않기 때문에, 파일명은 영문이름으로 제출)
 - (b) 소스코드에 반드시 주석이 기재돼 있어야 함
- ※ 파일명 형식 어길 시, 감점

(3) 기타

- ① 제출 마감: 5월 19일 수요일 오전 11:59까지 (마감 후에 제출되는 과제는 받지 않음)
- ② 마감 직전에는 I-class 및 채점 서버에 학생들이 많이 몰리기 때문에 혼잡할 수 있으니, 마감 시간보다 1시간 정도 이전에 여유 있게 제출하는 것을 권장
- ③ 제출 후에는 제대로 제출되었는지 반드시 확인
- ④ 부정행위가 적발될 경우, 베낀 학생과 원본을 제공한 학생 모두 0점 처리 및 성적의 총점에서 -10점 감점. 인터넷이나 책(교재, 참고서 등)의 소스코드, 예전에 제출했던 과제 등을 베껴서 제출해도 부정행위로 처리