

A대학의 B본부는 학생과 교직원, 일반인으로부터 기탁받은 발전기금을 관리한다.
기탁받은 발전기금에 대한 정보는 파일에 저장된다.
발전기금 목록을 single-user 기반으로 관리하는 C++프로그램을 작성하시오.

I. 프로그램 구동

I-1. 각 구성원이 가지는 정보는 다음과 같다.

- 학생은 학번(key), 이름, 소속학과, 휴대전화번호, 총기탁액 정보를 가진다.
- 교직원은 사번(key), 이름, 소속부서, 휴대전화번호, 내선번호, 총기탁액 정보를 가진다.
- 일반인은 이름, 전화번호, 총기탁액 정보를 가진다.
 - 일반인은 프로그램에서 자동 생성하는 key값인 기탁자번호를 추가적으로 가지며, 기탁자번호는 "V00001", "V00002"와 같이 순차번호를 포함한 string 객체로 한다.
 - 삭제되는 일반인이 있더라도 기탁자번호를 다시 채우거나 재사용하지 않는다.
- 모든 전화번호는 반드시 XXX-XXXX-XXXX의 형태로 들어오는 것으로 가정한다.
- 모든 key값들 간에는 중복이 존재하지 않으며, 중복 전화번호도 존재하지 않는다.
입력 파일에서도 제약조건에 맞게 중복값이 존재하지 않는 것으로 가정한다.

I-2. 프로그램은 명령형 인자로 발전기금파일을 읽어들이는 것으로 시작한다.

- 첨부된 예시 파일 FundList.txt와 같이 파일의 각 내용은 공백으로 구분된다.
- 채점시에는 파일명이 다른 여러 파일을 사용하여 평가한다.
- 명령형 인자는 항상 전달되어야 한다.
 - 전달된 파일명에 내용이 없는 경우에는 기탁 정보가 빈 리스트인 것으로 간주한다.

I-3. 프로그램에서는 다음의 기능을 수행한다

- [전체 조회]
 - 전체 기탁 내역을 기탁액 내림차순으로 조회. 기탁액이 같은 경우 정렬기준은 기탁액(내림차순) > 이름(오름차순) > 전화번호(오름차순)으로 한다.
- [신규 기탁자 등록] 새로운 기탁자와 기부액 등록
 - 파일 내용과 같은 순서로 입력하는 것으로 가정.
단 일반인의 경우 key값은 사용자가 입력하지 않고 프로그램에서 자동 생성.
 - 같은 key 값이나 전화번호가 있다면 오류 메시지 출력
- [기존 기탁자 변경] 기탁자의 기탁액 변경
 - key 또는 전화번호를 입력받아 기탁액 추가
- [기존 기탁자 삭제] 기탁 내용을 전부 삭제하여 목록에서 제외

I-4. 프로그램 종료를 선택하면 변경된 내용을 명령형 인자로 전달된 파일에 덮어 쓰지 여부를 확인하여 저장 후 종료 또는 저장하지 않고 종료한다.

- 파일의 구조는 입력되는 형태와 반드시 동일하게 유지.
- 효율성 증가를 위해 파일에 저장할 때도 기탁액 내림차순으로 정렬하여 저장.

I-5. 구동 예시는 다음과 같다(붉은 색이 사용자 입력).

```
> run.exe TestCase1
***** 발전기금 관리 프로그램 *****
----- 기탁현황 -----
[ 일반 ] 김제니(기탁자코드:V00001) 010-3333-7777 1000000
[ 일반 ] 박채영(기탁자코드:V00002) 010-7895-6587 500000
[교직원] 김남준(사번:F00401, 부서:전자공학부(x7123)) 010-1212-3434 100000
[ 학생 ] 김석진(학번:20180477, 학과:컴퓨터공학과) 010-9999-8888 100000
[ 학생 ] 김지수(학번:20140543, 학과:컴퓨터소프트웨어공학과) 010-1234-5678 100000
[교직원] 민윤기(사번:E09876, 부서:생활관(x7073)) 010-4567-3210 100000
[교직원] 김태형(사번:E01234, 부서:기획협력처(x7070)) 010-4567-9876 50000
[ 학생 ] 박지민(학번:20190777, 학과:신소재공학부) 010-7777-3333 30000
[ 학생 ] 전정국(학번:20211234, 학과:기계시스템공학과) 010-9876-5432 10000

[1] 전체 조회 / [2] 신규 기탁자 등록 / [3] 기탁정보 변경 / [4] 기탁자 삭제 / [5] 종료 : 2
신규 기탁자 정보를 입력하세요: 일반 리사 010-8282-2580 70000
<신규>[ 일반 ] 리사(기탁자코드:V00003) 010-8282-2580 70000

[1] 전체 조회 / [2] 신규 기탁자 등록 / [3] 기탁정보 변경 / [4] 기탁자 삭제 / [5] 종료 : 2
신규 기탁자 정보를 입력하세요: 학생 유재석 20101234 경영학과 010-9999-8888 5000000
!!! 동일한 전화번호가 존재하여 신규 등록 불가 !!!

[1] 전체 조회 / [2] 신규 기탁자 등록 / [3] 기탁정보 변경 / [4] 기탁자 삭제 / [5] 종료 : 3
변경을 원하는 사번이나 학번 또는 전화번호(XXX-XXXX-XXXX)를 입력하세요 : 010-1212-3434
추가 기탁액을 입력하세요 : 500000
<변경> [교직원] 김남준(사번:F00401, 부서:전자공학부) 010-1212-3434 7123 600000

[1] 전체 조회 / [2] 신규 기탁자 등록 / [3] 기탁정보 변경 / [4] 기탁자 삭제 / [5] 종료 : 3
변경을 원하는 사번이나 학번 또는 전화번호(XXX-XXXX-XXXX)를 입력하세요 : 20190777
추가 기탁액을 입력하세요 : 100000
<변경> [ 학생 ] 박지민(학번:20190777, 학과:신소재공학부) 010-7777-3333 130000

[1] 전체 조회 / [2] 신규 기탁자 등록 / [3] 기탁정보 변경 / [4] 기탁자 삭제 / [5] 종료 : 4
삭제를 원하는 사번이나 학번 또는 전화번호(XXX-XXXX-XXXX)를 입력하세요 : 010-3333-7777
<삭제> [ 일반 ] 김제니(기탁자코드:V00001) 010-3333-7777 1000000
```

[1] 전체 조회 / [2] 신규 기탁자 등록 / [3] 기탁정보 변경 / [4] 기탁자 삭제 / [5] 종료 : 1
 [교직원] 김남준(사번:F00401, 부서:전자공학부(x7123)) 010-1212-3434 600000
 [일반] 박채영(기탁자코드:V00002) 010-7895-6587 500000
 [학생] 박지민(학번:20190777, 학과:신소재공학부) 010-7777-3333 130000
 [학생] 김석진(학번:20180477, 학과:컴퓨터공학과) 010-9999-8888 100000
 [학생] 김지수(학번:20140543, 학과:컴퓨터소프트웨어공학과) 010-1234-5678 100000
 [교직원] 민윤기(사번:E09876, 부서:생활관(x7073)) 010-4567-3210 100000
 [일반] 리사(기탁자코드:V00003) 010-8282-2580 70000
 [교직원] 김태형(사번:E01234, 부서:기획협력처(x7070)) 010-4567-9876 50000
 [학생] 전정국(학번:20211234, 학과:기계시스템공학과) 010-9876-5432 10000

[1] 전체 조회 / [2] 신규 기탁자 등록 / [3] 기탁정보 변경 / [4] 기탁자 삭제 / [5] 종료 : 5
 변경 사항을 TestCase1에 저장하려면 W/w, TestCase1을 이전 상태로 유지하려면 C/c : W
 >>> 변경 사항 저장 후 종료 <<<

> more TestCase1

교직원 F00401 김남준 전자공학부 010-1212-3434 7123 600000
 일반 V00002 박채영 010-7895-6587 500000
 학생 20190777 박지민 신소재공학부 010-7777-3333 130000
 학생 20180477 김석진 컴퓨터공학과 010-9999-8888 100000
 학생 20140543 김지수 컴퓨터소프트웨어공학과 010-1234-5678 100000
 교직원 E09876 민윤기 생활관 010-4567-3210 7073 100000
 일반 V00003 리사 010-8282-2580 70000
 교직원 E01234 김태형 기획협력처 010-4567-9876 7070 50000
 학생 20211234 전정국 기계시스템공학과 010-9876-5432 10000

II. C++ 프로그램 요구 조건

II-1. 목록은 MyList 클래스에 저장한다.

o MyList는 Linked List 또는 동적배열을 포함하는 구조로 구현한다.

- Linked List로 구현하는 경우 더 높은 점수를 얻을 수 있다.
- 정적 배열은 허용하지 않는다. 동적 배열은 반드시 resizable 하게 구현해야 한다.

o MyList를 template으로 구현하면 더 높은 점수를 얻을 수 있다.

II-2. 학생과 교직원, 일반인의 공통 속성만 모은 class Person을 정의하고 상속을 사용한다.

o 상속을 사용하지 않는 설계는 허용하지 않는다.

- * 예를 들어 모든 속성들의 합집합인 class Person을 만들고 Person형 배열만 사용하는 구현은 0점 처리함.

- 학번과 사번, 기탁자코드는 모든 사람이 가지는 속성이 아니므로, 부모 class인 class Person이 아닌 파생클래스가 가지게 구현한다.

II-3. 다형성을 적용하여 구현하면 더 높은 점수를 얻을 수 있다.

- Template으로 구현되는 MyList 는 다형성 전용으로 가정하고 구현한다.
 - 주요 자료형은 MyList<Person *> (또는 Key를 추가한 MyList<Person *, string>) 또는 LinkedList<Person> (또는 Key type을 추가한 LinkedList<Person, string>)으로 선언할 수 있다.
- 다형성을 사용한 구현에서는 자신을 type을 파악할 수 있는 변수를 하나라도 사용하면, 다형성에 의한 가산점은 받을 수 없다.
 - 예를 들어, 멤버 변수 int type 에 0이면 일반, 1이면 학생 등을 저장하는 방식

II-4. 다형성을 적용하지 못하더라도 구현하여 제출하면 부분 점수를 받을 수 있다.

- 전체 목록을 기탁액 내림차순 조회하지 않고, 학생 목록 / 교직원 목록 / 일반인 목록을 각각 별도로 내림차순 조회하고 저장하는 형식으로 구현할 수 있다.

II-5. 정렬을 적용하지 못하더라도 구현하여 제출하면 부분 점수를 받을 수 있다.

- 입력 순서 또는 입력 역순으로 출력하고 저장해도 부분 점수를 받을 수 있다.

II-6. 정렬을 적용하는 경우에는 코드 효율성에 따라 감점이 있을 수 있다.

- 한번 프로그램을 거친 후에는 정렬된 목록이 입력되므로, $O(N\log N)$ 정렬 방식을 적용하는 것보다는 insertion sort 의 insertion을 적용하는 것이 효율이다.
- 목록 자체를 정렬해 두지 않고, 출력시에 매번 정렬하는 방식은 허용하지 않는다.

II-7. 명령형 인자와 파일 입출력은 반드시 구현해야 한다.

III. 채점 기준

III-1. 구동의 정확성과 적용 기술 수준

- [필수 요소] 상속, 명령형 인자의 파일 처리, 동적 배열 또는 Linked List (최대 60점)
- [추가 요소] Template, 다형성, 목록 정렬 (최대 40점. 총 100점)

III-2. 클래스 설계와 가독성/유지보수성의 적절성(감점 최대 30점, 추가 점수 최대 10점)

III-3. 메모리와 속도 효율의 적절성(감점 최대 20점. 추가 점수 최대 10점)

- 동적 배열 또는 Linked List와, 동적 생성되는 객체에 대한 정확한 delete 여부 중요

- o class에 필요없는 변수 포함 최소화
- o 동적배열이나 Linked List 모두에서 정렬key가 존재하는 경우 신규/변경/삭제의 $O(N)$ 의 복잡도를 피할 수 없으므로, 이외의 기본적인 부분에서 비효율적인 코드를 최소화하도록 신경쓸 것

IV. 기타

- o 위에서 명시하지 않은 예외 사항은 발생하지 않는 것으로 가정하고 구현한다.
 - 예를 들어 파일에 잘못된 값이 들어가는 경우 등
- o spec에 명확하지 않은 부분에 대해서는 금오톡톡으로 교수에게 문의한다.
 - spec 변경이 있을 수 있으므로 주기적으로 LMS 공지사항을 확인 바람.
- o 주어진 요구사항을 구현하는 실제 시스템은 DB기반으로 multi-user의 동시 transaction을 처리하도록 구현되어야 한다. 본 과제는 C++의 다양한 기능을 과제 수준에서 종합적으로 구현해 보는 것을 목적으로 하므로, 다소 비효율적인 부분이 있을 수 있음을 양지하고 적절한 수준까지만 최적화를 고민하도록 한다.
 - 예를 들어, key값이나 전화번호의 빠른 탐색 등을 위한 hash-map 등은 가산점 없음.