

파일구조 과제 2(파일 저장구조)을 모두 만족하며 추가로 아래의 기능을 만족하는 프로그램을 C++로 작성하여라. 단, delete는 없다.

1. 다음과 같은 Record를 파일로 저장한다. 여기서, ID가 primary key로 사용된다.

```
Student(char name[20], unsigned ID, float score, char dept[10])
```

2. 다음과 같은 조건을 만족하여야 한다.

1) 검색기능의 향상을 위하여 ID는 Dynamic Hash, score는 B+-tree를 적용하여 검색 속도를 빠르게 한다.

(1) Record 입력:

```
boolean insert(char* name, unsigned *ID, float *score, char* dept)
```

return: true – 입력이 성공, false – 입력이 실패

입력되는 데이터는 각 parameter로 지정하며, ID가 중복되는 것이 있으면 false로 return

student.dat 파일뿐만 아니라, score의 색인을 위하여 score.idx 파일도 binary 로 생성하여 관리한다. idx에 저장되는 B+-tree에는 student.dat에 저장된 record의 block 번호가 함께 저장되어야 한다.

(2) ID로 검색:

```
boolean searchByID(char* name, unsigned ID, float *score, char* dept)
```

return: true – 검색이 성공, false – 검색이 실패

검색한 데이터는 name, score, dept로 반환

사용 예

```
char        name[21];
float        score;
unsigned     ID=202124125;
char        dept[11];
boolean      flag=searchByID(name, ID, &score, dept);
if(flag) cout<<"name: "<<name<<", ID: "<<ID<<", score: "<<score<<"dept: "<<dept<<endl;
else cout<<"Record ID: "<<ID<<" - NOT found"<<endl;
```

(3) score로 검색: [scoreMin, scoreMax] 사이의 모든 record를 검색

```
boolean searchByScore(unsigned *number, unsigned *ID, float scoreMin, float scoreMax)
```

return: true – 검색이 성공, false – 검색이 실패

검색한 데이터는 number로 해당 record 개수와 ID array로 반환

사용 예

```
char        name[21];
char        dept[11];
unsigned     ID[MAX_NUMBER]; // we suppose that MAX_NUMBER is defined somewhere
boolean      flag=searchByScore(&number, ID, 4.1, 4.3);
if(!flag) cout<<"Search Not Found\n";
else {
    for(int i=0;i<number;i++) {
        searchByID(name, ID[i], &score, dept);
        cout<<"name: "<<name<<", ID: "<<ID[i]<<", score: "<<score<<"dept: "<<dept<<endl;
    }
}
```

- 위의 기능은 파일에 직접 입출력으로 이루어져야 한다. 만일 main memory에 데이터를 올리고 처리할 경우, 최소 점수 (10/100)만 주어진다.
- 위의 함수는 main 프로그램에서 호출을 한다. Main 프로그램은 과제 검사할 때 주어질 예정이다.
- 입력 데이터도 과제 검사 때 주어질 것이며 record의 수는 만개 이상 100만개 미만이다.

5) 수행시간(입력부터 ID 및 score 검색까지)이 가장 빠른 프로그램을 3개 선발하여 다음과 같은 보너스를 준다.

1등: 전체 누적점수(과제 및 시험)의 50%

2등: 전체 누적점수의 30%

3등: 전체 누적점수의 15%

3. 제출방법: 위의 함수 프로그램 소스코드 (C++)를 Plato에 upload. 테스트는 C++11을 기준으로 컴파일하여 결과를 확인할 예정이다.

4. 제출마감: 6월 25일 오후 6시