

2023 Spring “Computer Programming”

Exercise 2. <C++ File I/O & Vector>

| 다음 파일을 eTL에서 “과제 > Exercise 2”에 제출합니다. (~ 23/4/18 23:59)

학번-EX2.cpp

ex) 202320131-EX2.cpp

| 주어진 **Skeleton**을 이용해 작성합니다.

| 주어진 ‘filename.txt’와 ‘score.txt’를 이용합니다.

| ‘...’으로 표시된 부분을 수정합니다.

- 각 ‘...’은 단어 하나 혹은 한 줄만 제거된 것입니다.
- 올바른 단어 혹은 문장을 넣으시면 정상 동작하는 코드입니다.
- 여러 Line으로 작성하실 필요 없습니다.
- 함수 prototype에 있는 ‘...’도 수정해야 합니다.

| 상업적으로 판매하는 프로그램이 아닌 개념 공부를 위해 사용하는 코드이므로 Error Case를 따로 잡지 않았습니다.

| Skeleton Code내의 **Comment**와 실습 강의 자료를 함께 활용하시면 되겠습니다.

Problem)

- 학생들의 점수가 들어있는 **score.txt**.
- 학생들 점수를 점수대 별로 나누어 (A, B, C, D, fail) 파일에 저장.
- A~fail grade에 해당하는 점수가 들어갈 **파일의 이름**은 **filename.txt** 내부에 존재.

```
ex2 > ≡ filename.txt
1  A_score.txt
2  B_score.txt
3  C_score.txt
4  D_score.txt
5  Fail_score.txt
```

```
ex2 > ≡ score.txt
1  41
2  67
3  34
4  0
5  69
6  24
7  78
8  58
9  62
10 64
11 5
12 45
```

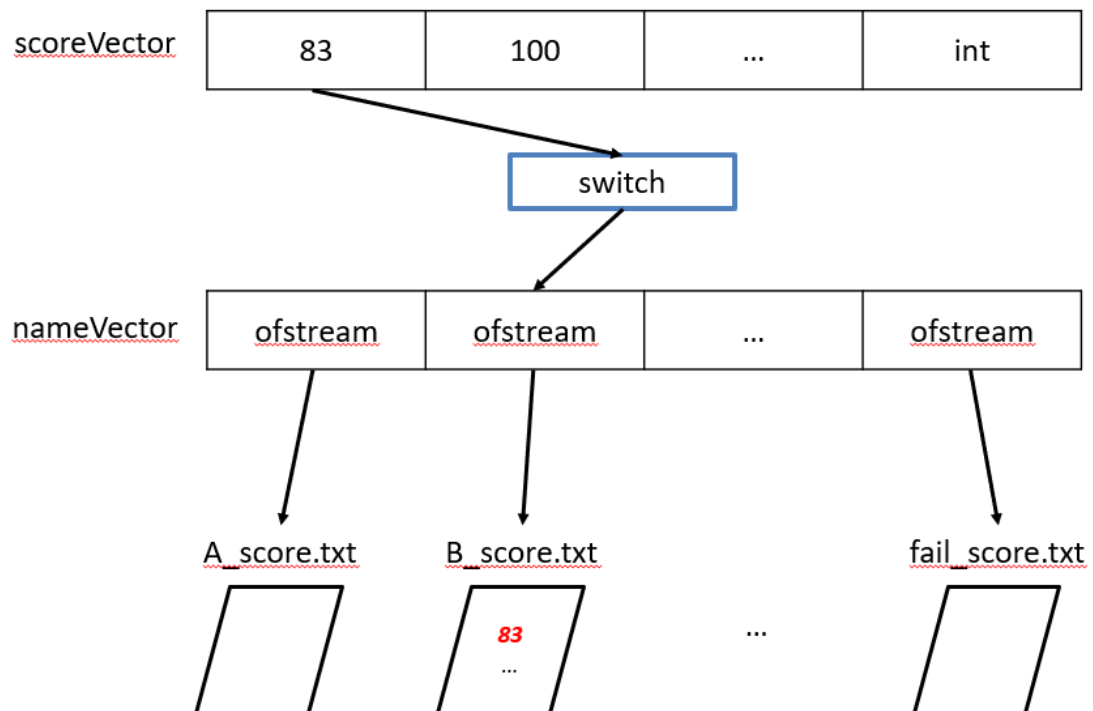
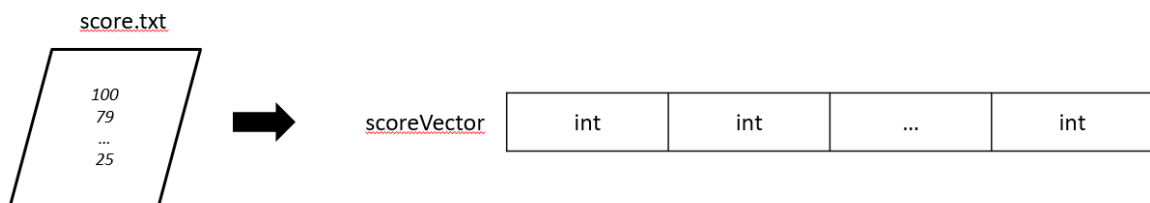
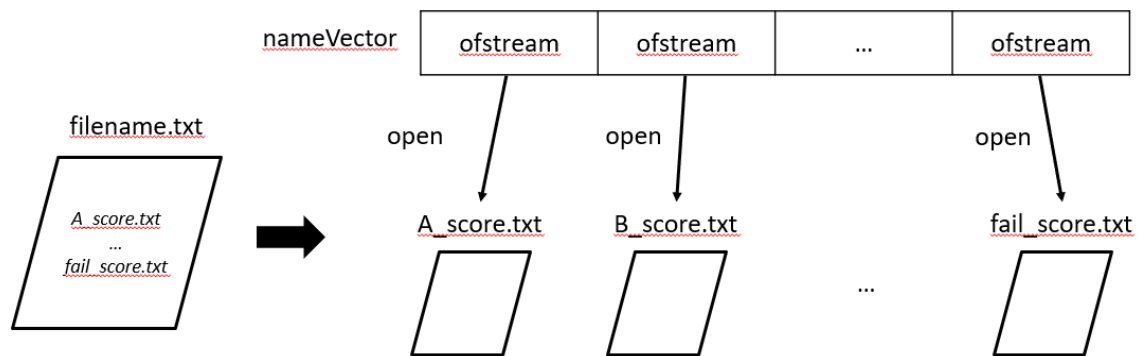
-
- 이 Code에서는 filename.txt에 들어있는 이름들을 이용해 file을 create.
 - 그 후, score.txt를 읽고 점수대에 맞추어 각 file에 write.

```
ex2 > ≡ A_score.txt
1  91
2  95
3  91
4  92
5  95
6  99
7  94
8  90
9
```

```
ex2 > ≡ B_score.txt
1  81
2  82
3
```

*맨 마지막의 newline은 있어도 괜찮습니다 / 정렬 및 중복 고려하지 않고 작성합니다.

-
- 각 file에 access 할 수 있는 object 5개를 nameVector로 관리
 - score.txt 의 점수들을 scoreVector로 관리
 - nameVector와 scoreVector를 이용해 open, IO, close 를 수행



- **Functions**

- **Void openGradeFiles()**

- Arguments: input file stream의 pointer, output file stream vector의 pointer
- Local variable인 Output file stream vector의 iterator와 string 변수를 이용
- Argument로 들어온 vector pointer와 iterator를 이용해 해당 vector의 처음부터 끝까지 loop을 돌
- 한 번의 iteration마다 input file stream의 pointer가 point하고 있는 file stream에서 한 line씩 읽음
- 읽은 line과 iterator를 이용해 iterator가 point 하는 output file stream을 open

- **void readScores()**

- Arguments: input file stream의 pointer, integer vector의 pointer
- Local Variable인 string 변수를 이용
- Argument로 들어온 input file stream pointer를 이용해 string 변수에 한 line 씩 read
- 한 line을 read할 때마다 argument로 들어온 integer vector에 맨 끝에 해당 line을 push. (이때 string to integer는 stoi()을 이용해 구현)

- **void writeScores()**

- Arguments: integer vector의 pointer, output file stream vector의 pointer
- Local variable인 integer vector의 iterator를 이용
- Argument로 들어온 integer vector의 pointer와 iterator를 이용해 해당 vector의 처음부터 끝까지 loop을 돌
- 한 번 iteration마다 switch-case를 이용해 integer vector의 원소 값을 확인하고 점수 영역에 맞게 나눔
- Case문을 통해 나뉜 각 점수 영역은 output file stream pointer와 integer vector iterator를 이용해 적절한 file에 integer 값을 write함 (이때, Access하는 file은 comment로, write하는 형식은 skeleton에 구현되어 있음)
- **Case 문 범위에 있는 “...” 은 범위를 표현하기 위한 것이므로 수정하지 않음**
e.g. case 90 ... 100 ← 90에서 100까지 라는 의미

- **void closeInputFile()**
 - Argument로 들어오는 input file stream pointer를 이용해 해당 file을 close
- **void closeGradeFiles()**
 - Argument: output file stream vector의 pointer
 - Local variable인 output file stream vector의 iterator를 이용
 - Argument로 들어온 output file stream vector의 pointer와 iterator를 이용해 해당 vector의 처음부터 끝까지 loop을 돌
 - 한 번 iteration마다 iterator를 이용해 output file stream vector의 원소 하나를 close
- **main()**
 - "filename.txt"와 "score.txt"를 각각 읽을 수 있는 input file stream object를 heap 영역에 2개 만듦
 - 해당 object의 주소를 담을 수 있는 input file stream type pointer 변수 2개를 선언 및 각각 주소 할당
 - Output file stream 5개를 담을 수 있는 vector를 heap 영역에 만듦
 - 해당 vector의 주소를 담을 수 있는 output file stream vector type pointer 변수 1개를 선언 및 주소 할당
 - 비어 있는 Integer vector를 heap 영역에 만듦
 - 해당 vector의 주소를 담을 수 있는 integer vector type pointer 변수 1개를 선언 및 주소 할당
 - 위 함수 prototype 및 code를 확인하여 함수 argument에 각종 main내의 local 변수를 적절히 전달하며 call