

CSED 232 Object-Oriented Programming (Spring 2024)

Programming Assignment #2

Due date: 3월 22일 23시 59분 59초

담당 조교: 강병욱 (kbu417@postech.ac.kr)

[안내사항]

1. 모든 문제는 C++의 standard 입출력(i.e. std::cin, std::cout)을 기본으로 합니다.
2. 프로그램 기능 점수 기준은 채점용 testcase 통과 여부입니다.
 - 1) 채점용 testcase들의 점수는 동일하지 않습니다. 문제에서 요구하는 기능별 난이도에 따라 배점이 다릅니다.
 - 2) 예외 처리 미구현에 대한 패널티는 높지 않습니다. 즉, 예외 처리가 필요한 testcase 점수는 필요 없는 testcase에 비해 낮습니다.
 - 3) 채점용 testcase는 비공개입니다.

[감점]

1. 제출 기한이 지나면 얻은 총점의 20% 감점
2. 하루(24시간) 늦을 때마다 추가 20%씩 감점
 - a. 1일 이내: 20% 감점, 2일 이내: 40% 감점, 3일 이내: 60% 감점, 4일 이내 80% 감점
 - b. 4일 이상: 0점
3. 컴파일이 정상적으로 되지 않을 경우 프로그램 기능 점수 0점

[제출방식]

채점 환경은 **Windows Visual Studio 2022**입니다. 파일을 업로드하실 때, **개발환경 파일의 “파일 제출”페이지에 써 있는 대로** 맞춰 올려 주시기 바랍니다. 폴더명은 문제#_학번(e.g. prob1_20230000)으로 만들어 주시기 바랍니다. 또한 문제 폴더 안에 각 문제에 해당하는 Report(e.g. prob1_20230000_report)도 첨부하여 zip파일로 압축한 후 제출해 주시기 바랍니다.

문제마다 따로 프로젝트를 생성하고, 따로 압축하여 제출해 주시기 바랍니다. 즉, **총 2개의 파일을 제출**하셔야 합니다. 제출은 반드시 PLMS를 통해 제출해주시기 바랍니다. 이메일 제출은 받지 않습니다. 제출 기한을 기준으로 4일(3월 26일 23시 59분 59초)이 경과한 이후에는 0점이므로 PLMS에서도 과제 제출 받지 않습니다.

제출파일 예시) prob1_20230000.zip, prob2_20230000.zip

[채점기준]

1. 프로그램 기능 - 50%

- 1) 프로그램이 요구 사항을 모두 만족하면서 올바르게 실행되는가?

2. 프로그램 설계 및 구현 - 35%

- 1) 요구 사항을 만족하기 위한 프로그램(변수, 함수, 알고리즘 등) 설계가 적절한가?
- 2) 각 문제에서 제시한 세부 조건의 유의사항을 모두 만족하였는가?
- 3) 입력과 출력이 주어진 형식에 맞게 프로그램이 잘 작동하는가?

3. 프로그램 가독성 - 5%

- 1) 프로그램이 읽기 쉽고 이해하기 쉽게 작성되었는가?
- 2) 변수 및 함수 명이 무엇을 의미하는지 파악하기 쉬운가?
- 3) 프로그램의 소스 코드를 이해하기 쉽도록 주석을 잘 작성하였는가?

4. 보고서 구성 및 내용, 양식 - 10%

- 1) 보고서는 적절한 내용으로 이해하기 쉽고 보기 좋게 잘 작성되었는가?
- 2) 보고서의 양식을 잘 따랐는가?
- 3) 각 문제에서 제시한 질문이 있다면, 그에 대한 답변이 충분하고 적절한가?

[주의사항]

다른 사람의 프로그램이나 인터넷 등에 있는 프로그램을 단순히 복사(copy)하거나 수정해서 제출하면 부정행위로 간주됩니다. 부정행위 적발 시 'F' 학점을 받을 수 있으며, 학과에서 정한 기준에 따라 추가적인 불이익이 있을 수 있습니다.

문제 1 번: 학생 성적표 (배점 35 점)

[문제 설명]

객체지향프로그래밍 조교 포닉스는 성적표 관리 업무를 맡게 되었다. 가난한 대학원생 포닉스는 엑셀과 같은 사무용 소프트웨어를 결제할 돈을 아끼기 위해 성적표 관리 프로그램을 직접 구현하기로 마음먹었다. 포닉스가 무사히 돈을 아낄 수 있도록 성적표 관리 프로그램을 구현해보자.

[학생 정보]

- 1) id: 학번
 - 8 자리 양의 정수이다.
 - 사용자가 중복된 값을 입력하지 않는다고 가정한다.
- 2) name: 이름
 - 알파벳(A~Z, a~z) 10 개 이하로 구성된 공백을 포함하지 않은 문자열이다.
- 3) midterm_exam_score: 중간고사 점수
 - 범위가 0~100 인 정수이다.
- 4) final_exam_score: 기말고사 점수
 - 범위가 0~100 인 정수이다.
- 5) retake: 재수강 여부
 - 재수강이 아닌 학생은 사용자가 정수 0 으로 입력한다.
 - 재수강인 학생은 사용자가 정수 1 로 입력한다.
- 6) 사용자는 학생 수를 최대 100 명까지 입력한다고 가정한다.

[프로그램 기능]

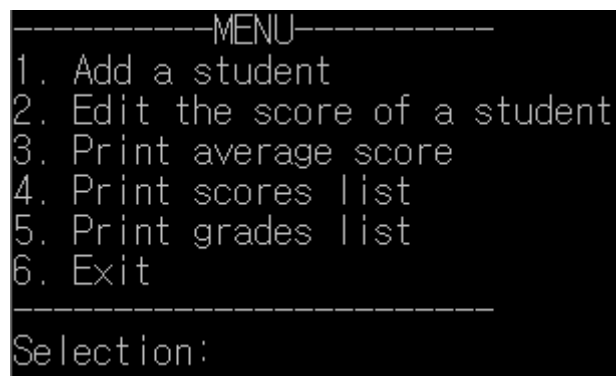
프로그램을 실행하면 그림 1 과 같은 메뉴가 출력되며 사용자로부터 메뉴 번호를 입력 받는다. 이때 선택 가능한 메뉴는 6 개이며, 각 메뉴에 대한 설명은 아래에서 설명된다. 각 메뉴의 기능 수행이 끝난 뒤에는 메뉴를 다시 출력하고 사용자가 다른 메뉴를 선택할 수 있도록 한다. 사용자가 잘못된 메뉴 번호를 입력 하지 않는다고 가정한다.

자동 채점 시스템 이용을 위해, 출력 형식(글자 수 등) 최대한 맞춰주세요. 메뉴 출력 코드는 다음 코드를 사용하면 됩니다.

```
std::cout << "-----MENU-----\n";
```

1. 학생 추가 (직접 구현해보세요.)
2. 학생 점수 수정 (직접 구현해보세요.)
3. 평균 점수 출력 (직접 구현해보세요.)
4. 점수표 출력 (직접 구현해보세요.)
5. 등급표 출력 (직접 구현해보세요.)
6. 프로그램 종료 (직접 구현해보세요.)

```
std::cout << "-----\n";
```



```
-----MENU-----
1. Add a student
2. Edit the score of a student
3. Print average score
4. Print scores list
5. Print grades list
6. Exit
-----
Selection:
```

그림 1

1) 학생 추가 (Add a student)

그림 2,3 을 참고하여 학생 정보(입력 정보 참고)를 사용자로부터 입력 받는 기능을 구현한다. 학생 입력을 성공적으로 마쳤을 경우 "The student is added!"를 출력한다.

다음과 같은 예외처리를 수행하라.

- 사용자는 id 를 항상 10 자리 이하의 정수로 입력한다. 입력 받은 정수가 8 자리 양의 정수가 아닌 경우 "Failed to add: invalid student id!"를 출력하고 학생 추가 작업을 중단한다.
- 사용자는 name 을 항상 20 개 이하로 구성된 공백을 포함하지 않은 문자열을 입력한다. 입력 받은 문자열에 알파벳이 아닌(A~Z,a~z) 문자가 포함되거나 10 자를 초과하는 경우 "Failed to add: invalid name!"를 출력하고 학생 추가 작업을 중단한다.
- 사용자는 midterm_exam_score 를 항상 10 자리 이하의 정수로 입력한다. 입력 받은 정수 범위가 0~100 이 아닐 경우 "Failed to add: invalid midterm exam score!"를 출력하고 학생 추가 작업을 중단한다.
- 사용자는 final_exam_score 를 항상 10 자리 이하의 정수로 입력한다. 입력 받은 정수 범위가 0~100 이 아닐 경우 "Failed to add: invalid final exam score!"를 출력하고 학생 추가 작업을 중단한다.
- 사용자는 retake 를 항상 1 자리의 정수로 입력한다. 입력 받은 정수가 0 이나 1 이 아닐 경우 "Failed to add: invalid retake value!"를 출력하고 학생 추가 작업을 중단한다.

```
-----MENU-----
1. Add a student
2. Edit the score of a student
3. Print average score
4. Print scores list
5. Print grades list
6. Exit
-----
Selection: 1
Student id: 20230001
Name: Ponix
Midterm exam score: 0
Final exam score : 0
Retake: 1
The student is added!
```

그림 2

```
-----MENU-----
1. Add a student
2. Edit the score of a student
3. Print average score
4. Print scores list
5. Print grades list
6. Exit
-----
Selection: 1
Student id: 20230002
Name: TAFriend
Midterm exam score: 1000
Failed to add: invalid midterm exam score!
-----MENU-----
```

그림 3

2) 학생 점수 수정 (Edit the score of a student)

그림 4,5 를 참고하여 사용자가 학생의 중간고사 점수 혹은 기말고사 점수를 수정할 수 있는 기능을 구현한다. 먼저 사용자가 중간고사 점수와 기말고사 점수 중에서 수정할 점수를 선택할 수 있도록 구현한다. 수정할 점수를 선택한 뒤, 사용자가 학번과 점수를 입력한다. 학생의 점수를 성공적으로 수정했을 경우 "Score editing done!"을 출력한다.

자동 채점 시스템 이용을 위해, 출력 형식(글자 수 등) 최대한 맞춰주세요. Function 메뉴 출력은 다음 코드를 사용하면 됩니다.

```
std::cout << "-----Function-----\n";
```

1. 중간고사 점수 수정 선택 (직접 구현해보세요.)
2. 기말고사 점수 수정 선택 (직접 구현해보세요.)

```
std::cout << "-----\n";
```

다음과 같은 예외처리를 수행하라.

- 사용자는 id 를 항상 10 자리 이하의 정수로 입력한다. 입력 받은 정수가 8 자리 양의 정수가 아닌 경우 "Failed to edit: invalid student id!"를 출력하고 학생 추가 작업을 중단한다.
- 입력 받은 학번과 일치하는 학생이 없을 경우 "Can't find the student id: [입력 받은 학번]"을 출력하고 점수 수정 작업을 중단한다.
- 사용자는 수정할 점수를 항상 10 자리 이하의 정수로 입력한다. 입력 받은 정수 범위가 0~100 이 아닐 경우 "Failed to edit: invalid exam score!"를 출력하고 학생 추가 작업을 중단한다.

```
-----MENU-----
1. Add a student
2. Edit the score of a student
3. Print average score
4. Print scores list
5. Print grades list
6. Exit
-----
Selection: 2
-----Function-----
1. Edit the midterm exam score
2. Edit the final exam score
-----
Selection: 1
Student id: 20230001
Score: 100
Score editing done!
```

그림 4

```
-----MENU-----
1. Add a student
2. Edit the score of a student
3. Print average score
4. Print scores list
5. Print grades list
6. Exit
-----
Selection: 2
-----Function-----
1. Edit the midterm exam score
2. Edit the final exam score
-----
Selection: 2
Student id: 20230002
Can't find the student id: 20230002
-----MENU-----
```

그림 5

3) 평균 점수 출력 (Print average score)

그림 6 을 참고하여 중간고사 점수 + 기말고사 점수를 합산한 점수의 평균과 표준편차를 출력한다. 평균은 "Average score: [평균 점수]" 형식으로 출력하고, 표준편차는 다음 줄에 "Standard deviation: [표준편차]" 형식으로 출력한다. 이 때, 유효숫자는 5 자리까지만 출력한다. 유효숫자에 대한 자세한 설명은 [세부조건]을 참고하라.

```
-----MENU-----
1. Add a student
2. Edit the score of a student
3. Print average score
4. Print scores list
5. Print grades list
6. Exit
-----
Selection: 3
Average score: 120
Standard deviation: 22.672
```

그림 6

학생 수: n

i -번째 학생의 총점: x_i

평균 수식: $\bar{x} = \frac{\sum x_i}{n}$

표준편차 수식: $\sigma = \sqrt{\frac{\sum (x_i - \bar{x})^2}{n}}$

4) 점수표 출력 (Print scores list)

학생 정보를 중간고사 점수 + 기말고사 점수를 합산한 점수를 내림차순으로 정렬한다. 정렬은 Bubble sort 로 구현한다. 점수가 같은 학생들간의 정렬은 학번을 기준으로 오름차순으로 정렬한다. 그림 7 과 같이 학생 정보를 출력한다. (출력 순서: 학번/이름/중간고사점수/기말고사점수/중간고사점수+기말고사점수) 학생 정보 출력을 마치고 "Finished printing the list!"를 출력한다.

```
-----MENU-----
1. Add a student
2. Edit the score of a student
3. Print average score
4. Print scores list
5. Print grades list
6. Exit
-----
Selection: 4
20230005 StudentD 86 57 143
20230002 StudentA 83 54 137
20230003 StudentB 58 76 134
20230001 Ponix 100 0 100
20230004 StudentC 32 54 86
Finished printing the list!
-----MENU-----
```

그림 7

5) 등급표 출력 (Print grades list)

모든 학생들의 등급을 판별하는 기능을 구현한다. 등급 판별 기준은 아래에 서술되어 있다. 모든 학생에 대한 등급 판별을 마치고, 그림 8 과 같이 학생 정보를 출력한다. (출력 순서: 학번/이름/총점(중간고사점수+기말고사점수)/재수강여부/등급) 학생 정보 출력을 마치고 "Finished printing the list!"를 출력한다.

학생 정보 출력 순서에 정해진 규칙은 따로 없습니다. 등급 판별만 정상적으로 수행되면 정답으로 인정됩니다.

등급표 출력 기능 testcase 에서는 동점자가 없는 testcase 로 채점합니다. 동점자 처리는 고려할 필요 없습니다.

첨부한 Problem1-5_testcase 내용을 복사&붙여넣기 하고 5 번 메뉴 선택하면 그림 8 과 같은 결과 쉽게 확인 가능합니다. 단, 채점용 testcase 는 첨부파일과 다르단 점을 참고해주세요.

등급 판별 기준은 다음과 같다.

- ① 다음 조건을 만족한 학생은 A 를 받는다.
 - ◆ 총점 150 점 초과
 - ◆ 총점이 상위 30%이내
 - ◆ 10 명의 학생 중 1,2,3 등만 A 를 받을 수 있다.
 - ◆ 단, '학생 수 * 30%'가 1 미만일 경우 상위 1 등은 30%이내가 아니지만 A 를 받을 수 있다.
- ② 다음 조건을 만족한 학생은 B 를 받는다.
 - ◆ 총점 100 점 초과
 - ◆ 총점이 상위 70%이내
 - ◆ 단, '학생 수 * 70%'가 2 미만일 경우 상위 2 등은 B 를 받을 수 있다.
- ③ 다음 조건을 만족한 학생은 C 를 받는다.
 - ◆ 총점 50 점 초과
 - ◆ 총점이 하위 10%이내가 아님.
 - ◆ 단, '학생 수 * 90%'가 3 미만일 경우 3 등은 C 를 받을 수 있다.
- ④ A,B,C 중 그 무엇도 받지 못한자의 등급은 D 이다.
- ⑤ 재수강인 학생은 원래 성적보다 한 단계 낮은 성적을 부여한다. 단, D 는 그대로 D 다.

```
-----MENU-----
1. Add a student
2. Edit the score of a student
3. Print average score
4. Print scores list
5. Print grades list
6. Exit
-----
Selection: 5
20220001 Ponix 100 1 D
20220002 TAFriend 200 1 B
20230001 StudentB 133 0 B
20230002 StudentC 167 0 A
20230003 StudentD 65 0 D
20230004 StudentE 115 0 B
20230005 StudentF 195 0 A
20230006 StudentG 139 0 B
20230007 StudentH 131 0 B
20230008 StudentI 101 0 C
20230009 Stupid 7 0 D
Finished printing the list!
-----MENU-----
```

그림 8

6) 프로그램 종료 (Exit)

"Good Bye!"를 출력하고 프로그램을 종료한다.

[세부조건]

Struct 그리고 Loop&Branch 문을 활용하여 문제를 해결하는 것을 목표로 하는 문제입니다.

Report 에서 struct 에 어떤 변수들을 저장하였는지 서술해주세요.

등급표 출력 기능에서 branch 문의 개수를 최소화하여 설계해보세요. branch 문의 개수가 평가기준은 아니지만, 프로그램 실행 속도 최적화를 위해 연습해보도록 합시다. Report 에서 branch 문의 개수를 줄이기 위해 혹은 프로그램 실행 속도 최적화를 위해 어떤 방식으로 등급표 출력 기능을 구현했는지 서술해주세요.

1. 유효숫자가 5 자리를 넘을 경우, 반올림하여 출력 (e.g. 122.1458 → 122.15)
 - 다음 코드로 출력 조건을 만족할 수 있다.
 - `#include <iomanip>`
 - `std::cout.precision(5);`
2. struct 사용 가능 / class 사용 불가
3. [STL](#) 사용 불가 (STL 제외한 C++ Standard Library 사용 가능)
4. 출력 형식 불일치 시에 소량의 감점 있습니다.

문제 2 번: 포스맨 (배점 15 점)

[문제 설명]

요원 포닉스는 총학의 대파공작원(대학원 파견 공작원)이고 당신은 그의 전담 연락원이다. 요원 포닉스는 전담 연락원인 당신에게 정기적으로 암호문을 송신한다. 요원 포닉스의 암호문을 무사히 해독할 수 있도록 암호해독 프로그램을 구현해보자.

[입력 정보]

문장이 입력으로 주어진다. 문자열은 아래와 같이 구성되어 있다. 아래에 속하지 않는 문자열은 암호문에 포함되어 있지 않는다고 가정한다.

- 알파벳 (A~Z,a~z)
- 특수문자 ,./()!/@/#/\$/%/&/'/"
- 숫자 (0~9)
- 공백 (띄어쓰기, 개행문자)

[프로그램 기능]

프로그램을 실행하면 그림 9 와 같이 "Enter a cyphertext"이 출력되고 다음 줄에 사용자가 문장을 입력한다. 암호문의 마지막 2 글자는 항상 ".."이며, 암호문 중간에 ".."은 입력되지 않는다. 암호문의 마지막 글자를 인식하여 프로그램이 정상적으로 종료할 수 있도록 구현한다. 암호문을 입력 받아 암호문 노이즈 비율과 해독 결과를 출력한다. 암호문 노이즈 비율 계산 방법과 해독 방법은 아래 내용을 참고하라. 암호문 노이즈 비율과 해독 결과 출력을 마쳤으면 "Decryptor turn off!"를 출력하고 프로그램을 자동으로 종료한다.

```
Enter a cyphertext
I'm really enjoying graduate sc&ool.
professor and seniors are all good people.
I also go to vacation with people in the lab.
Th* vacation date$ are form 5.3 to 5.4 (Note that it is not weekend!)
I get paid well. I earn 415.9 dollars per week.
@fter paying tuition, I still have half my salary left over.
I hope you get into gr*duate school too.
$%#3 mArch 28. ponix..
Ciphertext noise rate: 1.9022%
Hexadecimal digits: 535441593A28
Decryption result: STAY:(
Decryptor turn off!
```

그림 9

1) 암호문 노이즈 비율 계산

암호송수신 장비의 한계로 수신 과정에서 원본 암호문에 노이즈가 발생할 수도 있다. 암호문의 노이즈 비율 계산 공식은 다음과 같다. '노이즈 문자 수 / 총 문자 수 * 100'. 총 문자 수에는 공백도 포함된다. 노이즈 문자에 해당되는 문자는 아래 설명을 참고하라. 암호문 노이즈 비율은 그림 9 와 같이 "Ciphertext noise rate: [노이즈 비율]%" 형식으로 출력한다. 이때, 노이즈 비율의 유효숫자는 5 자리만 출력한다. 유효숫자에 대한 자세한 설명은 [세부조건]을 참고하라.

노이즈 문자의 목록은 다음과 같다.

- #
- \$
- %
- &
- *

2) 암호문 해독

암호문에 나타나는 순서대로 16 진수 숫자에 해당되는 문자들을 모아서 하나의 문자열로 만든다. 문자열을 그림 9 와 같이 "Hexadecimal digits: [문자열 결과]" 형식으로 출력한다.

만들어진 16 진수 숫자로 구성된 문자열에서 문자들을 2 개씩 묶어서 ASCII 코드로 디코딩한다. 해독된 암호문을 그림 9 와 같이 "Decryption result: [해독결과]" 형식으로 출력한다.

```
Enter a cyphertext
I'm really enjoying graduate sc&ool.
professor and seniors are all good people.
I also go to vacation with people in the lab.
Th* vacation date$ are form 5.3 to 5.4 (Note that it is not weekend!)
I get paid well. I earn 415.9 dollars per week.
@fter paying tuition, I still have half my salary left over.
I hope you get into gr*duate school too.
$%#3 march 28. ponix..
Ciphertext noise rate: 1.9022%
Hexadecimal digits: 53544159328
Undecryptable: the number of hexadecimal digits is odd!
Decryptor turn off!
```

그림 10

그림 10 을 참고하여 다음과 같은 예외처리를 수행하라.

- 암호문에 나타나는 16 진수 숫자의 개수가 홀수일 경우 해독된 암호문 출력 대신 "Undecryptable: the number of hexadecimal digits is odd!"를 출력한다.
- ASCII 코드의 범위를 벗어나는 경우 해독된 암호문 출력 대신 "Undecryptable: invalid hexadecimal digits pair!"을 출력한다.
- 암호문에 나타나는 16 진수의 숫자가 없을 경우 암호문 출력 대신 "Undecryptable: where are the hexadecimal digits!"를 출력한다.

16 진수 숫자 목록은 다음과 같다.

- 숫자 0~9
- 대문자 알파벳 A~F

[세부조건]

Loop&Branch 문을 활용하여 문제를 해결하는 것을 목표로 하는 문제입니다.

Report 에서 branch 문의 개수를 줄이기 위해 혹은 프로그램 실행 속도 최적화를 위해 어떤 방식으로 프로그램을 구현했는지 서술해주세요. 문제 1 과 마찬가지로 branch 문의 개수나, 프로그램 실행 속도가 평가기준은 아닙니다.

ex) 보편적으로 글을 쓸 때, 알파벳의 개수가 다른 문자 종류의 개수에 비해 많은 편이다. 따라서, 현재 문자가 알파벳인지 확인하는 if 문이 다른 else if 문보다 앞에 있으면, 다른 else if 문 테스트가 실제로 실행되는 경우가 적어지므로 프로그램 평균 실행 속도가 더 빠를 것이다.

1. 유효숫자가 5 자리를 넘을 경우, 반올림하여 출력 (e.g. 122.1458 → 122.15)
 - 다음 코드로 출력 조건을 만족할 수 있다.
 - `#include <iomanip>`
 - `std::cout.precision(5);`
2. `<cctype>`, `<string>` 사용 가능
3. 출력 형식 불일치 시에 소량의 감점 있습니다.