

CSED 232 Object-Oriented Programing (Spring 2021)

Programming Assignment # 4

- Templates & STL-

Due date : 5월 21일 23시 59분 00초

담당 조교 : 김상기 (kimsangki@postech.ac.kr)

주의사항

1. 모든 문제는 file 입출력이 필수입니다. input파일은 문제 1번에서 input_p1.txt, 문제 2번에서 input_p2.txt입니다. 그리고 output 파일은 문제 1번에서 output_p1_[학번].txt, 문제 2번에서 output_p2_[학번].txt으로 파일명을 결정해주시기 바랍니다.
2. Report의 경우 report_[학번].pdf의 형태로 제출해주시기 바랍니다.
3. 채점은 Windows Visual Studio 2019 환경에서 이루어집니다.
4. 제출 format을 지키지 않을 경우 0점입니다. 제출 format을 지키지 않은 것은 파일명이 잘못되었을 경우, 압축이 잘못되었을 경우 등을 모두 포함합니다.

감점

- 제출 기한이 지나면 얻은 총점의 20% 감점
- 추가로 하루(24시간) 늦을 때마다 20%씩 감점
1일 이내 지연: 20% 감점, 2일 이내 지연: 40% 감점, 5일 이상 지연: 0점
- 컴파일이 정상적으로 이루어지지 않을 경우 0점
- 형식을 맞추지 않을 경우 0점

제출방식

제출하실 때 헤더 파일과 소스파일만을 제출하시기 바랍니다. 문제 1번의 경우 prob1_set_[학번].h, prob1_set_[학번].cpp, prob1_main_[학번].cpp로 파일명을 통일해주시기 바랍니다. 문제 2번의 경우 prob2_matrix_[학번].h, prob2_matrix_[학번].cpp, prob2_main_[학번].cpp으로 파일명을 통일해주시기 바랍니다. 그리고 이것들과 보고서를 assign4_[학번] 폴더에 넣은 후 압축을 해서 제출해주시기 바랍니다. 조교가 딱 한 번만 압축해지를 하면 되게끔 제출하시기 바랍니다. 반드시 PLMS를

통해 제출해주시기 바랍니다. 이메일 제출은 인정되지 않습니다. 5일이 지날 경우 0점입니다.

채점 기준

1. 프로그램 기능 – 70%
 - 프로그램이 테스트 케이스를 몇 개나 통과하는가?
2. 프로그램 요구조건 구현– 20%
 - 각 문제에서 요구하는 조건을 잘 충족시켰는가?
3. 프로그램 보고서 – 10%

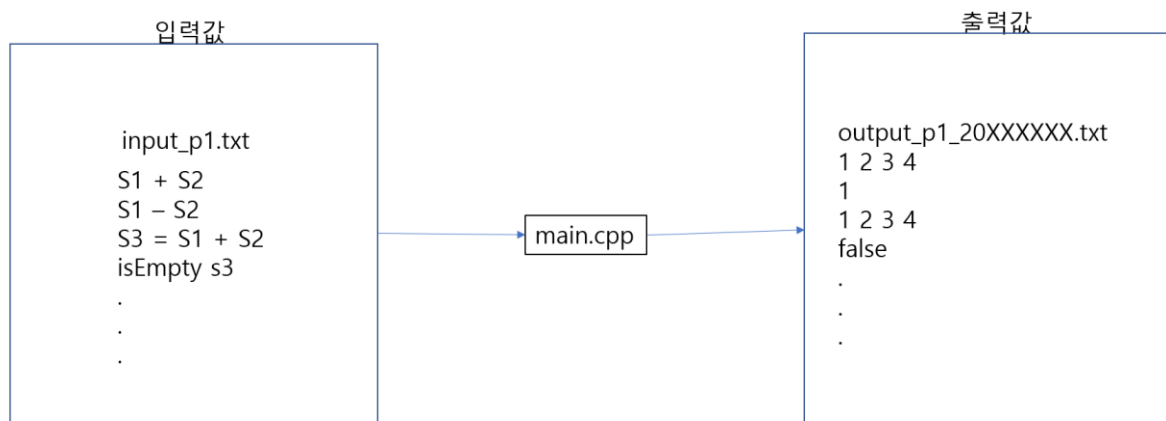
다른 사람의 프로그램이나 인터넷에 있는 프로그램을 복사(copy)하거나 간단히 수정해서 제출하면 학점은 무조건 'F'가 됩니다. 이러한 부정행위가 발견되면 학과에서 정한 기준에 따라 추가의 불이익이 있을 수 있습니다.

문제 1번 – (배점 : 40점)

Goal : Set STL의 일부를 구현하는 것을 통해 STL의 이해도를 높인다.

STL 중 자주 쓰이는 Container인 Set을 구현해보자. 실제 Set을 구현하는 것은 매우 어렵고 힘든 일이므로 그 중 일부만을 간단하게 구현해보자.

<Overview>



input_p1.txt는 연산해야 하는 값들을, S#.txt은 Set에 대한 정보를 담고 있다. 이들을 읽고 해당되는 연산을 수행 결과값을 output_p1_[학번].txt에 적는다.

<MySet Class 구현>

```
Class MySet
private:
    학생들이 자유롭게 정의하시오.

public:
    생성자
    소멸자
    MySet operator+(const MySet& S2)
    MySet operator+(const int E1)
    MySet operator-(const MySet& S2)
    MySet operator-(const int E1)
    MySet operator*(const MySet& S2)
    MySet operator=(const MySet& S2)
    bool operator==(const MySet& S2)
    bool contains(const MySet& S2)
```

```
bool contains(const int E1)
bool isEmpty()
int getSize()
string toString()
```

위에 제시된 필수적인 멤버 함수를 정의하기 위해 멤버함수와 멤버변수를 추가로 선언할 수 있다.

<각 멤버 함수 상세 설명>

1. 생성자: 생성자는 2개를 작성한다. 초기화 값이 없는 경우 MySet을 공집합으로 초기화하는 생성자와 초기값이 있을 경우 해당 초기값으로 MySet을 생성하는 생성자 이렇게 2가지를 작성한다. 초기화 매개변수의 형식은 학생들의 재량에 맡긴다.
2. 소멸자: Set을 저장하기 위한 저장공간을 해제하는 동작을 수행한다.
3. MySet operator+(const MySet& S2): MySet S2와의 합집합을 연산 후 return한다.
4. MySet operator+(const int E1): MySet에 원소를 하나 집어넣은 결과 set을 return한다.
5. MySet operator-(const MySet& S2): MySet - S2, 즉 차집합을 계산하고 return한다.
5. MySet operator-(const int E1): MySet에 원소 하나를 뺀 값을 return한다.
7. MySet operator*(const MySet& S2): MySet과 S2와의 교집합을 계산하고 return한다.
8. MySet operator=(const MySet& S2): MySet의 값을 MySet S2로 변경한다.
9. bool operator==(const MySet& S2): MySet과 MySet S2가 같으면 참, 아니면 거짓을 return한다.
10. bool contains(const MySet& S2): MySet이 S2를 포함하면 참, 아니면 거짓을 return한다.
11. bool contains(const int E1): 원소 E1을 MySet이 포함하면 참, 아니면 거짓을 return한다.
12. bool isEmpty(): MySet이 공집합이면 참, 아니면 거짓을 return한다.
13. int getSize(): MySet의 원소 개수를 return한다.
14. string toString(): MySet을 string값으로 return한다. 형식은 각 원소를 ' '로 구분하는 것으로 한다. ex) MySet 값이 1, 2, 3 일 경우 toString()의 return 값은 "1 2 3"으로 한다.

조건)

1. 멤버 변수는 반드시 private, 멤버 함수는 public으로 선언해야 한다.
2. STL set 및 추가적인 STL container를 사용하는 것을 금한다.
3. Class 이름과 멤버 함수 이름은 위에서 언급한 것과 일치시킨다.
4. 연산자는 모두 멤버 함수로 정의한다.
5. MySet은 prob1_set_[학번].h에 선언, prob1_set_[학번].cpp에 정의를 진행하라.
6. 파일 입출력은 prob1_main_[학번].cpp에서 진행하라.

[입력]

입력으로 들어오는 원소의 범위는 0 ~ 2,147,483,647이다.

input_p1.txt는 한 줄마다 요구되는 계산이 존재한다. 매 줄의 계산 수행 후 결과값을 output_p1_[학번].txt에 적어라. 각 Set과 Element들은 별도의 파일에 저장되어 있다. Set의 경우 S#으로 input파일에서 표현되며 해당 파일은 S#.txt이다. Element의 경우 E#으로 input파일에서 표현되며 해당 파일은 E#.txt이다.

아래는 input_p1.txt에 나올 수 있는 연산 목록이다.

1. $S1 \ +/-/*/= S2$
 $S1 \ +/- E1$
 - 각종 집합간 연산을 수행 후 output 파일에 적는다.
2. $S1 = S2 \ +/-/* S3$
 $S1 = S2 \ +/- E1$
 $S1 = S2$
 - 집합 update. 이 연산만 유일하게 실제 S1.txt파일이 계산 결과값으로 수정된다. output파일에 적을 내용은 우항의 연산 결과이다.
3. $S1 \text{ contain } S2/E1$
 - set S1이 set S2/element E1을 포함하는지 여부를 output파일에 적는다.
4. isEmpty S1
 - Set S1이 empty인지 판별 후 output파일에 적는다.
5. size S1

- Set S1의 사이즈를 계산 후 output파일에 적는다.

ex)

input_p1.txt
S1 + S2 S1 - S2 S3 = S1 + S2 isEmpty S3 S1 contain S1 S1 contain E1 size S1 isEmpty S1 S1 = S1 - S2 size S1

S1.txt
1 2 3

S2.txt
2 3 4

S3.txt
4 5 6

E1.txt
1

[출력]

매 줄의 연산 결과마다 결과값을 적는다.

output_p1_[학번].txt
1 2 3 4
1

1 2 3 4
false
true
true
3
false
1
1

S1.txt
1

S2.txt
2 3 4

S3.txt
1 2 3 4

E1.txt
1

$S3 = S1 + S2$ 에 의해서 S3.txt값이 1 2 3 4로 변경되었다.

$S1 = S1 - S2$ 에 의해서 S1.txt값이 1로 변경된다. 이에 따라 S1의 size값이 1로 찍히는 상황을 볼 수 있다.

출력 마지막 부분에서 한 줄을 띄지 않도록 한다.

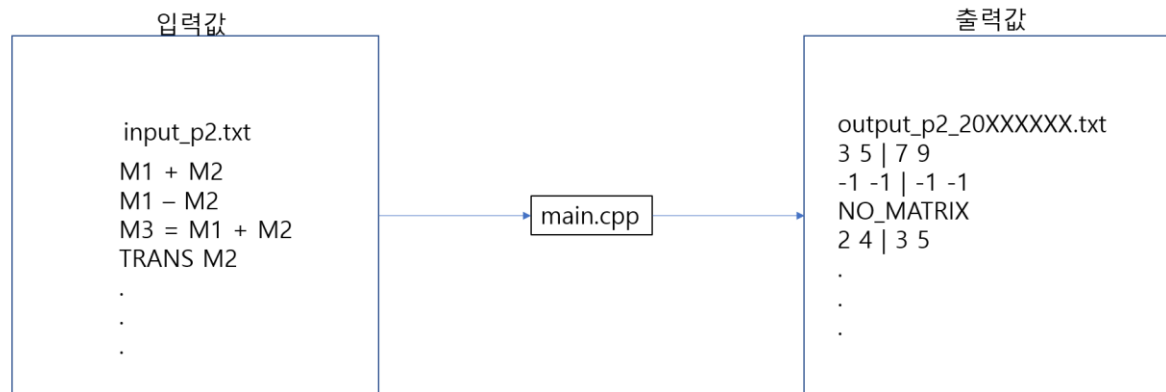
이 문제에서는 Error 핸들링을 생각하지 아니한다. 모든 입력값은 정상적으로 제공될 것이다.

문제 2번 – (배점 : 60점)

Goal : Template을 이용한 **Matrix** 연산을 구현하는 것을 통해 **Template**에 대한 이해도를 높인다.

N X N matrix를 template을 활용하여 만들어보자.

<Overview>



input_p2.txt는 연산해야 하는 값들을, M#.txt은 Matrix에 대한 정보를 담고 있다. 이들을 읽고 해당되는 연산을 수행 결과값을 output_p2_[학번].txt에 적는다.

<MySet Class 구현>

Template 사용

Class MyMatrix

private:

학생들이 자유롭게 정의하시오

public:

생성자

소멸자

MyMatrix operator+(const MyMatrix& M2)

MyMatrix operator-(const MyMatrix& M2)

MyMatrix operator*(const MyMatrix& M2)

MyMatrix operator=(const MyMatrix& M2)

MyMatrix transpose()

string toString()

위에 제시된 필수적인 멤버 함수를 정의하기 위해 멤버함수와 멤버변수를 추가로 선언할 수 있다.

1.생성자: 생성자는 2개를 작성한다. 초기화 값이 없는 경우 MyMatrix을 0으로 초기화하는 생성자와 초기값이 있을 경우 해당 초기값으로 N X N Matrix를 생성하는 생성자 이렇게 2가지를 작성한

다. 초기화 매개변수의 형식은 학생들의 재량에 맡긴다.

2.소멸자: Set을 저장하기 위한 저장공간을 해제하는 동작을 수행한다.

3.MyMatrix operator+(const MyMatrix& M2): MyMatrix와 행렬 M2와 행렬 덧셈을 진행한다.

4.MyMatrix operator-(const MyMatrix& M2): MyMatrix와 행렬 M2와 행렬 뺄셈을 진행한다.

5.MyMatrix operator*(const MyMatrix& M2): MyMatrix와 행렬 M2와 행렬 곱셈을 진행한다.

6.MyMatrix operator=(const MyMatrix& M2): MyMatrix값을 행렬 M2의 값으로 변경한다.

7.MyMatrix transpose(): 전치행렬을 반환한다.

8.string toString(): 행렬을 string으로 바꾼다. 형식은 다음과 같이 한다.

i -1th row | ith row | i+1 row ...

ex)

1 2 3 4 5 6 7 8 9	1 2 3 4 5 6 7 8 9
-------------------------	-----------------------

왼쪽 행렬에 대해 toString을 진행하면 오른쪽 처럼 나와야 한다.

조건)

1. 반드시 Template을 이용하여 하나의 Class가 여러 타입을 처리할 수 있어야 한다.
2. 멤버 변수는 반드시 private, 멤버 함수는 public으로 선언해야 한다.
3. 모든 STL 사용이 허용된다.
4. 추가적인 설치가 필요한 library는 사용이 불가능하다.
5. Class 이름과 멤버 함수 이름은 위에서 언급한 것과 일치시킨다.
6. 연산자는 모두 멤버 함수로 정의한다.
7. MyMatrix는 prob2_matrix_[학번].h에 선언, prob2_matrix_[학번].cpp에 정의를 진행하라.
8. 파일 입출력은 prob2_main_[학번].cpp에서 진행하라.

[입력]

한 줄마다 동작이 존재한다. 매 줄마다 동작을 수행 후 output_p2_[학번].txt에 결과값을 적어라.

input_p2.txt는 한 줄마다 요구되는 계산이 존재한다. 매 줄의 계산 수행 후 결과값을

output_p2_[학번].txt에 적어라. 각 Matrix는 별도의 파일에 저장되어 있다. Matrix는 M#으로 input 파일에서 표현되며 해당 파일은 M#.txt이다.

아래는 input_p2.txt에 나올 수 있는 연산의 목록이다.

1. M1 +/-/* M2

TRANS M1 +/-/* M2

M1 +/-/* TRANS M2

TRANS M1 +/-/* TRANS M2

- 각종 행렬간 연산을 계산 후 output파일에 적는다.

2. M3 = M1 +/-/* M2

M3 = TRANS M1 +/-/* M2

M3 = M1 +/-/* TRANS M2

M3 = TRANS M1 +/-/* TRANS M2

M3 = TRANS M2

M3 = M1

- 행렬 update. 유일하게 M3에 대응되는 M3.txt값이 변경된다. output파일에 적을 내용은 우항의 연산 결과 값을 적으면 된다.

3. TRANS M1

- 행렬 M1의 transpose 계산 후 결과값을 output파일에 적는다.

input_p2.txt
M1 + M2
M1 - M2
M3 = M1 + M2
TRANS M2
M1 = M1 * M1

M1.txt
1 2
3 4
M2.txt
2 3
4 5

[출력]

output_p2_[학번].txt
3 5 7 9 -1 -1 -1 -1 NO_MATRIX 2 4 3 5 7 10 15 22

M1.txt
7 10 15 22

M2.txt
2 3 4 5

M1 = M1 * M1연산에 의해서 M1.txt값이 변경된 것을 확인할 수 있다.

M3는 해당 M3.txt가 없으므로 NO_MATRIX라는 에러 메시지를 적었다. 자세한 내용은 아래의 Error Handling을 참조하라.

<Error Handling>

이 문제에서는 error handling을 평가한다. error의 종류는 다음과 같다.

1. 두 Matrix간 dimension이 맞지 않아 $+/-/$ 이 진행될 수 없다. 이 경우 UNAVAILABLE_OPERATION을 output 파일에 적어라
2. 입력파일의 Matrix가 정사각행렬이 아니다. 이 경우 NOT_SQUARE를 output 파일에 적어라
3. 입력파일의 Matrix에서 몇 가지 원소가 빠져 있다. 이 경우 LOST_ELEMENT를 output 파일에 적어라.
4. 입력파일에서 M#이라는 Matrix가 나왔다고 가정하자. 해당 Matrix값을 얻기 위해 M#.txt를 찾아보았지만 존재하지 않았다. 이 경우 NO_MATRIX를 output 파일에 적어라.

이외의 예러는 생각하지 아니한다.