Matrix Specifications

1. 기본 연산

1.1 행렬 더하기

Form : C = A + B

Define : C(x,y,..) = A(x,y,..) + B(x,y,..) for all elements.

Assert : size(A) == size(B)

A + B == B + A : 교환 법칙 성립

A + (B + C) == (A + B) + C : 결합 법칙 성립

A + Matrix.Zeros(getSize()) == Matrix.Zeros(getSize()) + A : 항등원 존재

A + (-A) == (-A) + A == 0 : 역원 존재

1.2 행렬 빼기

Form : C = A – B

Define : C(x,y,..) = A(x,y,..) - B(x,y,..) for all elements.

Assert : size(A) == size(B)

A - B == -(B – A)

1.3 행렬 곱하기

Form : C = A \* B

Define : A(x,y,..)\*B(x,y,.) = sigma\_k(A(x,k)\*B(k,y))

Assert : getRow(A) == getCol(B)

A \* (B \* C) == (A \* B) \* C

(A + B)\*C == A\*C + B\*C

1.4 행렬 나누기 (보류)

1.7 스칼라 곱하기

Form : C = A .\* B

Define : C(x,y) = A(x,y) .\* B(x,y)

Assert : size(A) == size(B)

A.\*B == B.\*A

(A.\*B).\*C == A.\*(B.\*C)

1.8 스칼라 나누기

Form : C = A ./ B

Define : C(x,y) = A(x,y) ./ B(x,y)

Assert : size(A) == size(B)

1.9 상수 더하기

Form : A = c + B or B+c where c is constant

Define : A(x,y,..) = c + B(x,y,..) for all elements. Also vice versa

Assert :

A+c == c+A

1.10 상수 빼기

Form : A = c - B or B-c where c is constant

Define : A(x,y,..) = c - B(x,y,..) for all elements. Also vice versa

Assert :

1.11 상수 곱하기

Form : A = c \* B or B \* c where c is constant

Define : A(x,y,..) = c \* B(x,y,..) for all elements. Also vice versa

Assert :

1.12 상수 나누기

Form : A = B/c where c is constant

Define : A(x,y,..) = B(x,y,..)/c for all elements. Also vice versa

Assert :

1.9 역행렬 구하기

Form : A.inv()

1.10 Transpose

Form : A’

Define : A’(x,y) = A(y,x)

1.11 Rank

1.12 행렬식(Determinant)

Form : A.det()

행렬 생성

2.1 바로 생성 A = matrix([3,3,3;4,4,4])

2.2 Eyes

Form : Matrix.Eyes(4)

2.3 Zeros

Form : Matrix.Zeros(4)

2.4 Ones

Form : Matrix.Ones(4)

2.5 대각행렬

Form : Matrix.Digonal(1,2,3,4)

행렬 IO 관련

3.1 특정 위치 값 수정

Form : A(2,4) = 1

3.2 프린트 (행단위, 행렬단위, 원소단위)

Form : A, A(2), A(2,4) …

3.3 Slicing

Form : A(1:2, 1:3)

3.4 뒤에다붙이기

Form : A.concatenate(B)

그밖에

4.1 getSize

Form : A.getSize(), A.getSize(1), A.getSize(2)…

4.4 Gaussian elimination (?)

* C++ googletest 단위테스트 적용방법

1. Googletest 다운받고 Cmake 다운받아서 cmake해야함 : <http://migam.tistory.com/85>
2. 솔루션 밑에폴더에서 오른쪽마우스 – 속성 – 코드생성 – 런타임 라이브러리 : 다중 스레드 디버그(/MTd)로 설정
3. 링커 – 입력 – 추가종속성에 빌드해서 만든 gtest.lib :

( C:\Users\LeeWonKyung\mybuild\Debug\gtest.lib) 추가

1. 구성속성 – VC++ 디렉터리에 포함디렉터리에 다음 추가 :

E:\googletest-master\googletest\include\;

1. **C/C++ - 전처리기 - 전처리기 정의 에**

**\_VARIADIC\_MAX=10** 을 해줘야함.