**Requirements**

* Python 3.5
* Anaconda

**How to Use**

1. IDE에서 실행하거나 프롬프트에서 Python power-of-matrix-aid.py를 입력한다.
2. 띄어쓰기를 구분으로 4개의 수를 입력한다.

**제한 사항**

* 2x2 행렬에 대한 계산이므로 입력은 항상 4개의 숫자(정수, 실수)가 입력되어야 한다.
* 4개 이외의 수만큼 입력하거나 숫자가 아닌 타입을 입력하면 예외처리가 되고 다시 데이터를 입력 받는다.
* 입력 받은 행렬의 고유값이 2개의 실근인 경우에만 실행 결과가 출력된다.
* 행렬P의 역행렬이 존재할 경우에만 실행 결과가 출력된다.

**프로그램 작동 원리**

1. 프로그램이 실행되면 ‘Power\_Of\_Matrix\_Aid’ 클래스를 실행한다.
2. ‘get\_input’ 메소드에서 제한 조건에 맞도록 4개의 숫자를 입력 받는다.
3. ‘calculate’ 메소드에서 입력받은 행렬A의 n제곱 계산을 쉽게 하기 위해 형태로 만든다.
4. 이를 위해 행렬A의 고유값과 고유벡터를 이용한다. 고유값을 구하기 위해 2차 방정식에 근의 공식을 사용한다. 이 때, 고유값이 허수가 되거나 중근일 경우에는 제한 사항에 따라 메시지를 출력하고 함수를 종료한다.
5. 위에서 구한 [고유값1, 0, 0, 고유값2] 행렬으로 행렬M을 형성한다.
6. 에서 두 개의 고유벡터 **x**를 구한다.
7. 2x1인 두 개의 고유벡터를 2x2로 합쳐 행렬P를 형성한다.
8. 만일 행렬P의 역행렬이 존재하지 않는다면 메시지를 출력하고 종료한다.
9. 행렬P, M, 을 출력한다.

**기타 사항**

* self.precision 값의 변경을 통해 반올림 할 자리 수를 정할 수 있다.
* 고유벡터를 구한 뒤 ‘refine\_vector’ 메서드에서 간단한 벡터로 다시 초기화 한다. 이 함수는 벡터 원소에 소수부가 없어질 때까지 10을 곱한 뒤, 두 원소를 그들의 GCD로 나눈 배열을 반환한다.
* = 이므로 A의 거듭제곱 형태를 쉽게 구할 수 있다.