

과제 1. 배열

과제 1

- ◆ 제출 마감일 : 3월 28일 (일요일) 10:59pm
- ◆ eClass 과제방에 제출
- ◆ 개인 과제이나 2인 1팀도 가능
 - 대표 1인만 제출(팀원 명시)
- ◆ 사용 언어
 - C / Java / Python / C++ 중 본인이 선택
- ◆ 가산점 없음
- ◆ 제출 양식
 - 보고서 (표지 + 문제 내용과 해결 방안 + 결과 캡처)
 - 소스코드 (별도 파일로 제출) : .c, .py, .java 등
 - 잘못 제출하는 일이 없도록 각별히 유의
 - 예) .sln, .proj, .class 등
- ◆ 보고서는 Word/HWP/PPT/PDF 로 제출
- ◆ 강의 자료에 있는 코드를 이용하지 않아도 좋

타인의 과제를 복사하지 말 것

(1) 다항식

- ◆ 다항식을 저장하는 2개의 방식을 배웠다. 2개의 방식을 이용하여 각각 다음의 프로그램을 작성하라.
 - 입력된 데이터로부터 다항식을 구성하여 화면에 표시한다.
 - 입력된 2개의 다항식의 합과 곱을 화면에 표시한다.
 - 입력된 2개의 다항식과 합, 곱 다항식으로 총 4개의 다항식이 구성된다.
 - 4개의 다항식 중 하나를 선택하여 x 의 값을 지정하면 결과값을 표시한다.
 - 2개의 방식은 두 개의 다항식을 곱하는 프로그램을 작성하라. 즉 2개의 프로그램이 제출되어야 한다.
 - 입력되는 다항식의 차수는 최대 5를 넘지 않는다. 즉, x^5 이상의 입력이 사용되지 않는다.
 - 다항식의 입력과 결과는 하나의 배열에 모두 담을 필요는 없다(강의 슬라이드 35페이지처럼 하나에 모두 담지 않아도 된다). 즉, 수식 1, 수식 2, 수식 3, 수식 4를 각각 다른 배열에 담아도 좋다.

(1) 다항식 : 방식 1- 모든 차수의 계수를 저장

◆ 실행 예

$$3x^3 + 6x + 3$$

수식 1을 입력하세요 : **3 0 6 3**

수식 2을 입력하세요 : **7 5 1**

$$7x^2 + 5x + 1$$

수식 1 + 2 는 3 7 11 4 (이것을 수식 3이라 하자) $3x^3 + 7x^2 + 11x + 4$

수식 1 * 2 는 21 15 45 51 21 3 (이것을 수식 4라 하자)
$$21x^5 + 15x^4 + 3x^3 + 42x^3 + 30x^2 + 6x + 21x^2 + 15x + 3$$
$$= 21x^5 + 15x^4 + 45x^3 + 51x^2 + 21x + 3$$

수식에 값을 넣으세요 **1 1** (수식 1의 x에 1을 넣겠다)

결과값은 12

수식에 값을 넣으세요 **3 2** (수식 3의 x에 2을 넣겠다)

결과값은 78

수식에 값을 넣으세요 **4 -1** (수식 4의 x에 -1을 넣겠다)

결과값은 -18

이 프로그램에서는 모든 다항식을
방식1로 저장하고 **방식1**로 처리해
야 한다.

(1) 다항식 : 방식 2- 0이 아닌 계수의 차수만 저장

◆ 실행 예

$$3x^3 + 6x + 3$$

수식 1을 입력하세요 : **3 3 6 1 3 0**

수식 2을 입력하세요 : **7 2 5 1 1 0**

$$7x^2 + 5x + 1$$

수식은 차수로 정렬된 상태로만 입력한다. 6 1 3 3 3 0 와 같은 식으로는 입력하지 않는다.

수식 1 + 2 는 3 3 7 2 11 1 4 0 (이것을 수식 3이라 하자) $3x^3 + 7x^2 + 11x + 4$

수식 1 * 2 는 21 5 15 4 45 3 51 2 21 1 3 0 (이것을 수식 4라 하자)

$$\begin{aligned} &21x^5 + 15x^4 + 3x^3 + 42x^3 + 30x^2 + 6x + 21x^2 + 15x + 3 \\ &= 21x^5 + 15x^4 + 45x^3 + 51x^2 + 21x + 3 \end{aligned}$$

수식에 값을 넣으세요 **1 1** (수식 1의 x에 1을 넣겠다)

결과값은 12

수식에 값을 넣으세요 **3 2** (수식 3의 x에 2을 넣겠다)

결과값은 78

수식에 값을 넣으세요 **4 -1** (수식 4의 x에 -1을 넣겠다)

결과값은 -18

이 프로그램에서는 모든 다항식을 **방식2**로 저장하고 **방식2**로 처리해야 한다.

(1) 다항식

◆ 실행 예

$$3x^3 + 6x + 3$$

수식 1은 몇 개의 항으로 구성되어 있나요? **3**

수식 1을 입력하세요 : **3 3 6 1 3 0**

수식 2는 몇 개의 항으로 구성되어 있나요? **3**

수식 2을 입력하세요 : **7 2 5 1 1 0**

$$7x^2 + 5x + 1$$

수식은 차수로 정렬된 상태로만 입력한다. 6 1 3 3 3 0 와 같은 식으로는 입력하지 않는다.

수식 1 + 2 는 3 3 7 2 11 1 4 0 (이것을 수식 3이라 하자) $3x^3 + 7x^2 + 11x + 4$

수식 1 * 2 는 21 5 15 4 45 3 51 2 21 1 3 0 (이것을 수식 4라 하자)

$$\begin{aligned} & 21x^5 + 15x^4 + 3x^3 + 42x^3 + 30x^2 + 6x + 21x^2 + 15x + 3 \\ & = 21x^5 + 15x^4 + 45x^3 + 51x^2 + 21x + 3 \end{aligned}$$

수식에 값을 넣으세요 **1 1** (수식 1의 x에 1을 넣겠다)

결과값은 12

수식에 값을 넣으세요 **3 2** (수식 3의 x에 2을 넣겠다)

결과값은 78

수식에 값을 넣으세요 **4 -1** (수식 4의 x에 -1을 넣겠다)

결과값은 -18

결과 수식도 정렬된 상태로 표시되어야 한다.

(2) 행렬

◆ 행렬은 다음의 2개 방식으로 저장할 수 있다.

- 배열에 그대로 저장하는 방법(Normal)
- 0이 아닌 값만 저장하는 방법(Sparse)

행렬의 합, 곱을 이해해야 한다.

◆ 2가지 방식을 이용하여 다음을 작성하라.

- 정방형 크기($N \times N$)의 행렬을 2개 입력 받는다.
- 방식1로 입력 받은 행렬과 행렬의 합과 곱을 차례로 표시한다.
- 방식2로 입력 받은 행렬과 행렬의 합과 곱을 차례로 표시한다.
- 행렬의 데이터는 10 미만의 정수만 사용한다.
- 행렬의 크기는 최소 2×2 , 최대 9×9
- 파이썬이 제공하는 행렬을 그대로 이용하지 말 것(직접 구현해야 한다)

(2) 행렬

◆ 방식1과 방식2의 행렬 저장 방식을 각각 구성해야 한다.

- 입력 받은 데이터를 방식1의 형태로 1, 2번 행렬을 저장해야 한다.
- 방식 1의 행렬에서 사용할 수 있는 `add_normal`, `multiply_normal` 함수를 각각 구현해야 한다.
- 입력 받은 데이터를 방식2의 형태로 3, 4번 행렬을 저장해야 한다.
- 방식 2의 행렬에서 사용할 수 있는 `add_sparse`, `multiply_sparse` 함수를 각각 구현해야 한다.

소스코드를 확인할 수 있도록
함수 이름을 지킬 것

(2) 행렬 - 방식 1 : 배열을 이용할 때

◆ 실행 예

행렬의 규격을 입력하세요. **3**

행렬 1의 데이터를 입력하세요. **1 0 3 0 0 0 2 0 0**

행렬 2의 데이터를 입력하세요. **2 0 1 0 2 0 0 1 0**

$$\begin{matrix} 1 & 0 & 3 \\ [0 & 0 & 0] \\ 2 & 0 & 0 \end{matrix}$$

$$\begin{matrix} 2 & 0 & 1 \\ [0 & 2 & 0] \\ 0 & 1 & 0 \end{matrix}$$

방식 1:

행렬 1(9)

$$\begin{matrix} 1 & 0 & 3 \\ 0 & 0 & 0 \\ 2 & 0 & 0 \end{matrix}$$

← 행렬 1을 규격대로 표시

행렬 2(9)

$$\begin{matrix} 2 & 0 & 1 \\ 0 & 2 & 0 \\ 0 & 1 & 0 \end{matrix}$$

← 행렬 2을 규격대로 표시

행렬 1+2(9)

$$\begin{matrix} 3 & 0 & 4 \\ 0 & 2 & 0 \\ 2 & 1 & 0 \end{matrix}$$

행렬 1, 2를 입력으로 add_1 함수 실행 결과를 출력

←
$$\begin{matrix} 1 & 0 & 3 \\ [0 & 0 & 0] \\ 2 & 0 & 0 \end{matrix} + \begin{matrix} 2 & 0 & 1 \\ [0 & 2 & 0] \\ 0 & 1 & 0 \end{matrix} = \begin{matrix} 3 & 0 & 4 \\ [0 & 2 & 0] \\ 2 & 1 & 0 \end{matrix}$$

행렬 1*2(9)

$$\begin{matrix} 2 & 4 & 1 \\ 0 & 0 & 0 \\ 4 & 0 & 2 \end{matrix}$$

행렬 1, 2를 입력으로 multiply_1 함수 실행 결과를 출력

←
$$\begin{matrix} 1 & 0 & 3 \\ [0 & 0 & 0] \\ 2 & 0 & 0 \end{matrix} * \begin{matrix} 2 & 0 & 1 \\ [0 & 2 & 0] \\ 0 & 1 & 0 \end{matrix} = \begin{matrix} 2 & 4 & 1 \\ [0 & 0 & 0] \\ 4 & 0 & 2 \end{matrix}$$

(2) 행렬 - 방식 2 : 희소행렬 저장법을 쓸 때

◆ 실행 예

