

# Warming-up 프로그램 1

2023-2 컴퓨터 그래픽스

# 1. 행렬 다루기

- 3 X 3 행렬의 덧셈, 뺄셈, 곱셈하는 프로그램을 구현한다.
- 연산을 할 2개의 행렬에는 0, 1, 2의 값 중에서 랜덤하게 설정한다. (사용자 입력 x)
- 다음의 명령어로 수행한다. 종료 명령어를 입력할 때까지 명령을 연속적으로 수행할 수 있도록 한다.
  - m: 행렬의 곱셈
  - a: 행렬의 덧셈
  - d: 행렬의 뺄셈
  - 숫자 (1~9): 입력한 숫자를 행렬에 곱한다.
  - r: 행렬식의 값 (Determinant) ➔ 입력한 2개의 행렬의 행렬식 값을 모두 출력한다.
  - t: 전치 행렬(Transposed matrix)과 그 행렬식의 값 ➔ 입력한 2개의 행렬에 모두 적용한다.
  - h: 3X3 행렬을 4X4 행렬로 변환하고 행렬식의 값 (4행4열 행렬식 값) 출력  
(단위 행렬의 값, 즉, 4행에 0을, 4열에 0을, 4행 4열의 위치에는 1을 추가한다)
  - s: 행렬의 값을 새로 랜덤하게 설정한다.
  - q: 프로그램 종료

실행 예)

$$\text{(명령어) m: } \begin{vmatrix} 001 \\ 111 \\ 010 \end{vmatrix} \cdot \begin{vmatrix} 101 \\ 000 \\ 010 \end{vmatrix} = \begin{vmatrix} 010 \\ 111 \\ 000 \end{vmatrix} \quad \text{(명령어) a: } \begin{vmatrix} 001 \\ 111 \\ 010 \end{vmatrix} + \begin{vmatrix} 101 \\ 000 \\ 010 \end{vmatrix} = \begin{vmatrix} 102 \\ 111 \\ 020 \end{vmatrix} \quad \text{(명령어 입력) r: } \begin{vmatrix} 001 \\ 111 \\ 010 \end{vmatrix} = 1 \quad \begin{vmatrix} 101 \\ 000 \\ 010 \end{vmatrix} = 0$$

$$\text{(명령어) t: } \begin{vmatrix} 001 \\ 111 \\ 010 \end{vmatrix} \rightarrow \begin{vmatrix} 010 \\ 011 \\ 110 \end{vmatrix} = 1 \quad \begin{vmatrix} 101 \\ 000 \\ 010 \end{vmatrix} \rightarrow \begin{vmatrix} 100 \\ 001 \\ 100 \end{vmatrix} = 0 \quad \text{(명령어) h: } \begin{vmatrix} 001 \\ 111 \\ 010 \end{vmatrix} \rightarrow \begin{vmatrix} 0010 \\ 1110 \\ 0100 \\ 0001 \end{vmatrix} = 1 \quad \begin{vmatrix} 101 \\ 000 \\ 010 \end{vmatrix} \rightarrow \begin{vmatrix} 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{vmatrix} = 0 \quad \text{(명령어)2: } \begin{vmatrix} 001 \\ 111 \\ 010 \end{vmatrix} * 2 = \begin{vmatrix} 0 & 0 & 2 \\ 2 & 2 & 2 \\ 0 & 2 & 0 \end{vmatrix}$$

## 2. 파일에서 데이터 읽기

- 문자와 숫자가 10줄 저장된 파일을 만든다. (파일 이름: data.txt)
- 데이터 파일 이름을 입력 받고, 파일에서 데이터를 읽어 문자열에 저장한다.
- 읽은 문자열을 출력하고, 공백을 기준으로 단어와 숫자의 개수를 세고, 대문자가 포함된 단어의 개수를 세어 출력한다.
  - 1개 이상의 공백이 있는 경우는 1개로 취급한다.
  - 알파벳이나 숫자 외의 특수문자들은 (예, / , - \* 같은 문자)는 구분하지 않고 연결된 값으로 카운트 한다.
  - 숫자와 문자가 모두 있는 문자열인 경우에는 문자로 카운트 하도록 한다.

예) input data file name: ***data.txt***

This is computer graphics warming up program.

1 2 3 4 5 6 7 8 9 10

File input output sample program

3D computer graphics

2023 Fall semester

Tech University Korea

Game and multimedia engineering department

1001 ada\_ada 1001

Monday Tuesday Wednesday Thursday Friday

abcdefghijklhdcba

word count: 32

number count: 13 (1 2 3 4 5 6 7 8 9 10 2023 1001 1001)

Capital word: 12 (This File Fall Tech University Korea  
Game Monday Tuesday Wednesday Thursday Friday)

### 3. 문자열 다루기 (2번 문제 확장)

- 2번 문제같이 파일에서 문자열들을 입력 받는다.
- 다음의 명령어를 실행한다.
  - d: 문장 한 줄 전체를 거꾸로 출력하기
  - e: 동일한 간격으로 특정 문자를 일정 갯수 삽입하기 (3문자 후 2개의 @ 문자 삽입. 공백도 하나의 문자로 취급하기)
  - f: 공백을 기준으로 문자 거꾸로 출력하기
  - g: 문자 내부의 특정 문자를 다른 문자로 바꾸기 (바꿀 문자와 새롭게 입력할 문자 입력받음)
  - h: 앞에서 읽었을 때, 뒤에서 읽었을 때 어디까지 같은 문자인지 출력하기
  - +: 문장에 있는 모든 숫자에 +1을 한다.
  - -: 문장에 있는 모든 숫자에 -1을 한다. (최소 숫자는 0, 즉 0에 -1을 해도 0으로 출력)
  - q: 프로그램 종료

예) input data file name: **data.txt** (*여기에서는 앞의 예에서 한 문장만을 예제로 적용함. 실제로는 파일에서 읽은 모든 문장에 각각 적용*)

This is computer graphics warming up program.

command: d: 문장 전체를 뒤집기

e: 동일한 간격으로 띄어쓰기를 일정 개수 삽입하기

f: 공백을 기준으로 문자 뒤집기

g: 문자 내부의 특정 문자열을 다른 문자열로 바꾸기

h: 앞뒤에서 읽었을 때 어디까지 같은 문자인지 출력하기

+: 숫자에 +1

- : 숫자에 -1

q: 종료

\* d, e, f 명령어는 기능을 켜다 끄다 하는 토글키처럼 사용됨.  
예) d 명령어인 경우에 문장을 뒤집기/원래대로를 번갈아 가며 적용  
예) e 명령어인 경우에 일정 문자를 삽입/삭제를 번갈아 가며 적용  
예) f 명령어인 경우에 각 단어를 뒤집기/원래대로를 번갈아 가며 적용

//--- 다음 페이지에 계속 소개

### 3. 문자열 다루기 (2번 문제 확장)

input the command: d

.margorp pu gnimraw scihparg retupmoc si sihT

input the command: d //--- 다시 원래 순서로 바뀜

This is computer graphics warming up program.

input the command: e //--- 3글자 뒤에 @ 2개를 삽입함.

Thi@@s i@@s c@@omp@@ute@@r g@@rap@@hic@@s w@@arm@@ing@@up @@pro@@gra@@m.

input the command: f

s@@ihT s@@i r@@etu@@pmo@@c s@@cih@@par@@g pu@@gni@@mra@@w m@@arg@@orp@@.

input the command: f //--- 다시 원래 순서로 바뀜

Thi@@s i@@s c@@omp@@ute@@r g@@rap@@hic@@s w@@arm@@ing@@up @@pro@@gra@@m.

input the command: e //--- e 명령어를 다시 입력하면 의한 @ 문자 삭제

This is computer graphics warming up program.

input the command: g i K //--- i를 K로 바꾸기 (입력 방법은 변경해도 무관)

ThKs Ks computer graphKcs warmKng up.

input the command: + //--- 숫자에 +1 (2번째 줄)

2 3 4 5 6 7 8 9 10 11

input the command: h //--- 8번째 줄 문자열: 1001ada\_ada1001

//--- 10번째 줄 문자열: abcdefghidcba

This is computer graphics warming up program.: 0

...

1001 ada\_ada 1001: 1001 ada

abcdefghidcba : abcd

## 4. 저장 리스트 만들기 (구조체 데이터 사용)

- 점 (x, y, z) 데이터 값을 저장하는 리스트를 만든다. (점 데이터는 각각 정수이고, 구조체를 사용하도록 한다.)
- 최대 10개의 점 데이터를 저장하도록 한다.
- 리스트에 데이터를 입력하거나 삭제하고 출력하는 명령어를 실행한다.
  - 각 명령어를 입력 받으면 결과 리스트를 다음페이지의 그림과 같이 항상 10개의 항목을 가진 리스트로 (인덱스와 데이터 값) 출력한다.
- 구현 함수 프로토타입과 명령어:
  - + x y z: 리스트의 맨 위에 입력 (x, y, z: 숫자)
  - -: 리스트의 맨 위에서 삭제한다.
  - e x y z: 리스트의 맨 아래에 입력 (명령어 +와 반대의 위치, 리스트에 저장된 데이터값이 위로 올라간다.)
  - d: 리스트의 맨 아래에서 삭제한다. (리스트에서 삭제된 칸이 비어있다.)
  - l: 리스트에 저장된 점의 개수를 출력한다.
  - c: 리스트를 비운다. 리스트를 비운 후 다시 입력하면 0번부터 저장된다.
  - m: 원점에서 가장 먼 거리에 있는 점의 좌표값을 출력한다.
  - n: 원점에서 가장 가까운 거리에 있는 점의 좌표값을 출력한다.
  - a: 원점과의 거리를 정렬하여 오름차순으로 정렬하여 출력한다. 인덱스 0번부터 빈 칸없이 저장하여 출력한다.
  - s: 원점과의 거리를 정렬하여 내림차순으로 정렬하여 출력한다. (다시 a 또는 s를 누르면 원래의 인덱스 위치에 출력한다.)
  - q: 프로그램을 종료한다.

\*\* 리스트에서 맨 위(인덱스 9번)까지 차고 아래칸(인덱스 0번)이 비어있으면 다음 데이터 입력할 때는 0번에 입력된다.

\*\* 즉, 10개의 칸을 다 채울 수 있어야 함.

# 4. 저장 리스트 만들기 (구조체 데이터 사용)

실행 예) + 0 1 0  
+ 0 1 1  
-  
e 1 1 1  
e 1 0 1  
+ 1 1 0  
d  
a  
m  
n  
l

- 리스트 1번 출력, (리스트 각 항목 옆에 length 출력)
- 리스트 2번 출력, (리스트 각 항목 옆에 length 출력)
- 리스트 3번 출력, (리스트 각 항목 옆에 length 출력)
- 리스트 4번 출력, (리스트 각 항목 옆에 length 출력)
- 리스트 5번 출력, (리스트 각 항목 옆에 length 출력)
- 리스트 6번 출력, (리스트 각 항목 옆에 length 출력)
- 리스트 7번 출력, (리스트 각 항목 옆에 length 출력)
- 리스트 8번 출력, (리스트 각 항목 옆에 length 출력)
- (1, 1, 1) 출력
- (0, 1, 0) 출력
- 리스트 길이: 3

1

9	
8	
7	
6	
5	
4	
3	
2	
1	
0	0 1 0

2

9	
8	
7	
6	
5	
4	
3	
2	
1	0 1 1
0	0 1 0

3

9	
8	
7	
6	
5	
4	
3	
2	
1	
0	0 1 0

4

9	
8	
7	
6	
5	
4	
3	
2	
1	0 1 0
0	1 1 1

5

9	
8	
7	
6	
5	
4	
3	
2	0 1 0
1	1 1 1
0	1 0 1

6

9	
8	
7	
6	
5	
4	
3	1 1 0
2	0 1 0
1	1 1 1
0	1 0 1

7

9	
8	
7	
6	
5	
4	
3	1 1 0
2	0 1 0
1	1 1 1
0	

8

9	
8	
7	
6	
5	
4	
3	
2	1 1 1
1	1 1 0
0	0 1 0