

## 2019-2 컴퓨터 그래픽스 워밍업 프로그램

### 1. 행렬 다루기

3 X 3 행렬의 덧셈, 뺄셈, 곱셈하는 프로그램을 구현한다. 다음의 명령어로 수행한다. 명령어를 연속적으로 수행할 수 있도록 한다.

M/m: 행렬의 곱셈

A/a: 행렬의 덧셈

D/d: 행렬의 뺄셈

R/r: 행렬식의 값 (Determinant)

H/h: 3X3 행렬을 4X4 행렬로 변환하고 행렬식의 값 (4행4열 행렬식 값) 출력  
(4행에 0을, 4열에 0을, 4행 4열의 위치에는 1을 추가한다)

Q/q: 프로그램 종료

\*\* 행렬의 값은 0 또는 1의 값으로 결정한다.

실행 예) 2개의 3X3 행렬을 출력한다.

$$\text{(명령어 입력) M: } \begin{vmatrix} 0 & 0 & 1 \\ 1 & 1 & 1 \\ 0 & 1 & 0 \end{vmatrix} \cdot \begin{vmatrix} 1 & 0 & 1 \\ 0 & 0 & 0 \\ 0 & 1 & 0 \end{vmatrix} = \begin{vmatrix} 0 & 1 & 0 \\ 1 & 1 & 1 \\ 0 & 0 & 0 \end{vmatrix}$$

$$\text{(명령어 입력) A: } \begin{vmatrix} 0 & 0 & 1 \\ 1 & 1 & 1 \\ 0 & 1 & 0 \end{vmatrix} + \begin{vmatrix} 1 & 0 & 1 \\ 0 & 0 & 0 \\ 0 & 1 & 0 \end{vmatrix} = \begin{vmatrix} 1 & 0 & 2 \\ 1 & 1 & 1 \\ 0 & 2 & 0 \end{vmatrix}$$

$$\text{(명령어 입력) R: } \begin{vmatrix} 0 & 0 & 1 \\ 1 & 1 & 1 \\ 0 & 1 & 0 \end{vmatrix} = 1$$

$$\text{(명령어 입력) H: } \begin{vmatrix} 0 & 0 & 1 \\ 1 & 1 & 1 \\ 0 & 1 & 0 \end{vmatrix} \rightarrow \begin{vmatrix} 0 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{vmatrix} = 1$$

### 2. 저장 리스트 만들기 (구조체 데이터 사용)

점 (x, y, z) 값과 원점에서의 거리를 저장하는 리스트를 만든다. (점 데이터는 구조체를 사용하도록 한다.) 최대 10개의 데이터를 저장하도록 한다. 리스트에 데이터를 입력하거나 삭제하고 출력하는 명령어를 실행한다. 각 명령 후 최종 결과 리스트를 아래의 표와 같이 항상 10개의 항목을 가진 리스트로 (인덱스, 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: 리스트의 길이를 출력한다.
- c: 리스트를 비운다.
- m: 원점에서 가장 먼 거리에 있는 점의 좌표값을 출력한다.
- n: 원점에서 가장 가까운 거리에 있는 점의 좌표값을 출력한다.
- q: 프로그램을 종료한다.

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

실행 예)      + 0 1 0                      -> 리스트 1번 출력, (리스트 각 항목 옆에 length 출력)  
                  + 0 1 1                      -> 리스트 2번 출력, (리스트 각 항목 옆에 length 출력)  
                  -                                -> 리스트 3번 출력, (리스트 각 항목 옆에 length 출력)  
                  e 1 1 1                      -> 리스트 4번 출력, (리스트 각 항목 옆에 length 출력)  
                  e 1 0 1                      -> 리스트 5번 출력, (리스트 각 항목 옆에 length 출력)  
                  + 1 1 0                      -> 리스트 6번 출력, (리스트 각 항목 옆에 length 출력)  
                  d                                -> 리스트 7번 출력, (리스트 각 항목 옆에 length 출력)  
                  m                                -> (1, 1, 1) 출력  
                  n                                -> (0, 1, 0) 출력  
                  l                                -> length of list: 2

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

### 3. 파일에서 데이터 읽기

문자와 숫자가 10줄 저장된 파일을 만든다. 파일에서 데이터를 읽어 문자열에 저장한다. 읽은 문자열을 출력하고, 공백을 기준으로 단어의 개수를 세고, 숫자는 따로 출력하고 개수를 센다.

예)      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

2019-2

Korea Polytechnic University

Game and multimedia engineering department

10 line sample data

Monday Tuesday Wednesday Thursday Friday

This is the last sentence.

word count:48

number: 1 2 3 4 5 6 7 8 9 3 10 2019-2 10      total: 13

#### 4. 도형간의 충돌체크

비교할 2개 도형의 좌표값을 [0, 500] 사이의 임의의 값으로 정한다. 두 개의 좌표값으로 표시한다. 키보드 명령을 사용하여 두 번째 도형의 좌표값을 이동시킨 후, 첫 번째 도형과의 충돌 체크 여부를 출력한다. 비교 도형은 사각형과 사각형, 사각형과 원의 조합을 사용한다. 원인 경우는 정원 (타원 X)으로 정한다. 즉, 두 좌표값을 입력하면 그 값을 사용하여 중심과 반지름을 구한다.

(조합 번호: 사각형 x 사각형은 1번, 사각형 x 원은 2번)

출력 예)

(1번- (rectangle x rectangle) / 2번- (rectangle x circle) )

(이동 Command: a(left) / d(right) /s(down) / w(up))

Input Shape Combination: 1

Shape 1: (100, 100) (300, 400)

Shape 2: (400, 400) (500, 500)

Command: a // 두 번째 도형을 왼쪽으로 이동시킨다: x 좌표값을 -40 이동

Shape 1:(100, 100) (300, 400)

Shape 2: (340, 400) (440, 500)

Command: w // 두 번째 도형을 아래쪽으로 이동시킨다: y 좌표값을 -40 이동

Shape 1:(100, 100) (300, 400)

Shape 2: (340, 360) (440, 460)

Command: a // 두 번째 도형을 왼쪽으로 이동시킨다: x 좌표값을 -40 이동

Shape 1:(100, 100) (300, 400)

Shape 2: (280, 360) (380, 460)

Rectangle 1 & Rectangle 2 collide!!

#### 5. 미로 만들기

50x50 크기의 2차원 배열을 만든다. 좌측 상단에서 우측 하단으로 가는 임의의 미로를 만든 후 화면에 출력한다. 0은 길, 1은 벽 (혹은 반대로)으로 표시한다. 이때, 길은 4방향인 좌/우/상/하로 연결되고, 다음의 조건에 맞게 길을 만든다.

조건 1) 미로는 한쪽 방향으로 5칸 이상 계속 이동할 수 없다.

조건 2) 미로는 좌우상하로 최소한 1번 이상 방향을 전환한 적이 있어야 한다.

enter키 입력으로 미로를 계속 바꿔서 화면을 다시 출력할 수 있게 하고, q 입력 시 프로그램 종료한다.

출력 예) (샘플로 10X10으로 나타냄)

0 0 1 1 1 1 1 1 1 1

1 0 0 0 1 1 1 1 1 1

1 1 1 0 1 0 0 0 1 1

1 1 1 0 1 0 1 0 1 1

1 1 1 0 0 0 1 0 1 1

1 1 1 1 1 1 1 0 1 1

1 1 1 1 1 1 0 0 1 1

1 1 1 1 0 0 0 1 1 1

1 1 1 1 0 1 1 1 1 1

1 1 1 1 0 0 0 0 0 0