

Warming-up 프로그램 1

2024-2 컴퓨터 그래픽스

(각 문제는 3점으로 채점됩니다.)

1. 행렬 다루기

- 4 X 4 행렬의 덧셈, 뺄셈, 곱셈하는 프로그램을 구현한다.
- 연산을 할 2개의 행렬에는 0, 1, 2의 값 중에서 랜덤하게 설정한다. (사용자 입력 X)
 - 단, 16개의 숫자 중 2의 값은 2개만 있도록 한다. 즉, 14개의 숫자는 0 또는 1, 2개의 숫자는 2로 설정한다.
- 다음의 명령어로 수행한다. 종료 명령어를 입력할 때까지 명령을 연속적으로 수행할 수 있도록 한다.
 - m: 행렬의 곱셈
 - a: 행렬의 덧셈
 - d: 행렬의 뺄셈
 - 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.
 - e: 행렬의 값 중 짝수만 출력한다. → 다시 누르면 홀수만 출력한다. → 다시 누르면 원래대로 출력한다.
 - s: 행렬의 값을 새로 랜덤하게 설정한다.
 - 숫자 (1~9): 입력한 숫자를 행렬에 곱한다.
 - q: 프로그램 종료

실행 예)

(명령어) m: $\begin{pmatrix} 0 & 0 & 1 & 1 \\ 2 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 2 & 1 & 1 & 0 \end{pmatrix} \cdot \begin{pmatrix} 2 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 \\ 0 & 2 & 1 & 1 \\ 1 & 0 & 0 & 0 \end{pmatrix} = \begin{pmatrix} 1 & 2 & 1 & 1 \\ 6 & 3 & 1 & 0 \\ 3 & 1 & 0 & 0 \\ 5 & 5 & 2 & 1 \end{pmatrix}$ (명령어) a: $\begin{pmatrix} 0 & 0 & 1 & 1 \\ 2 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 2 & 1 & 1 & 0 \end{pmatrix} + \begin{pmatrix} 2 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 \\ 0 & 2 & 1 & 1 \\ 1 & 0 & 0 & 0 \end{pmatrix} = \begin{pmatrix} 2 & 1 & 1 & 1 \\ 3 & 2 & 1 & 1 \\ 1 & 2 & 1 & 2 \\ 3 & 1 & 1 & 0 \end{pmatrix}$ (명령어 입력) r: $\begin{vmatrix} 0 & 0 & 1 & 1 \\ 2 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 2 & 1 & 1 & 0 \end{vmatrix} = 2$

(명령어) t: $\begin{pmatrix} 0 & 0 & 1 & 1 \\ 2 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 2 & 1 & 1 & 0 \end{pmatrix}^T = \begin{pmatrix} 0 & 2 & 1 & 2 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{pmatrix}$ (명령어) d: $\begin{pmatrix} 0 & 0 & 1 & 1 \\ 2 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 2 & 1 & 1 & 0 \end{pmatrix} - \begin{pmatrix} 2 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 \\ 0 & 2 & 1 & 1 \\ 1 & 0 & 0 & 0 \end{pmatrix} = \begin{pmatrix} -2 & -1 & 1 & 1 \\ 1 & 0 & -1 & 1 \\ 1 & -2 & -1 & 0 \\ 1 & 1 & 1 & 0 \end{pmatrix}$ (명령어) 3: $3 \cdot \begin{pmatrix} 0 & 0 & 1 & 1 \\ 2 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 2 & 1 & 1 & 0 \end{pmatrix} = \begin{pmatrix} 0 & 0 & 3 & 3 \\ 6 & 3 & 0 & 3 \\ 3 & 0 & 0 & 3 \\ 6 & 3 & 3 & 0 \end{pmatrix}$

2. 파일에서 문자열 읽기

- 문자와 숫자가 10줄 저장된 파일을 만든다. (파일 이름: data.txt)
- 데이터 파일 이름을 입력 받고, 파일에서 문자열을 읽고 저장한다.
- 읽은 문자열을 출력하고, 공백을 기준으로 단어를 구분한다.
 - 1개 이상의 공백이 있는 경우는 공백을 1개로 취급한다.
 - 알파벳이나 숫자 외의 특수문자들은 (예, / , - * 같은 문자) 구분하지 않고 연결된 값으로 카운트 한다.
- 다음의 명령어를 수행한다.
 - c: 대문자로 시작되는 단어를 찾아 그 단어를 다른 색으로 출력하고, 몇 개 있는지를 계산하여 출력한다. 다시 누르면 원래대로 출력한다.
 - d: 문장 한 줄 전체를 거꾸로 출력하기. 다시 누르면 원래대로 출력한다.
 - e: 3문자 후 2개의 @ 문자 삽입하기. 다시 누르면 원래대로 출력한다.
 - f: 공백을 기준으로 모든 단어들을 거꾸로 출력하기. 다시 누르면 원래대로 출력한다.
 - g: 문자 내부의 특정 문자를 다른 문자로 바꾸기 (바꿀 문자와 새롭게 입력할 문자 입력 받음) 다시 누르면 원래대로 출력한다.
 - h: 각 줄의 단어의 개수를 출력한다.
 - r: 한 문장의 단어의 개수에 따라 오름차순으로 정렬하여 문장들을 출력한다. 다시 누르면 내림차순으로 출력한다. 다시 누르면 원래대로 출력한다. (즉, 오름차순 → 내림차순 → 원래대로 → 오름차순 ...)
 - s: 명령어와 단어를 입력하면, 입력받은 문장들을 모두 출력하면서 그 단어를 찾아 다른 색으로 출력하고, 몇 개 있는지를 계산하여 출력한다. (대소문자 구분하지 않는다)
 - q: 프로그램 종료

2. 파일에서 문자열 읽기

- 콘솔에서 색 출력하기

```
#include <windows.h>
```

```
SetConsoleTextAttribute (GetStdHandle(STD_OUTPUT_HANDLE), attribute_color); //--attribute_color; WORD 타입으로 색상을 나타내는 숫자
```

- 콘솔에서 제공해주는 색상



- 사용 예)

```
#include <stdio.h>
```

```
#include <stdlib.h>
```

```
#include <windows.h>
```

```
int main()
```

```
{
```

```
    int i;
```

```
    for (i = 0; i < 16; i++)
```

```
    {
```

```
        unsigned short text = i;
```

```
        SetConsoleTextAttribute (GetStdHandle(STD_OUTPUT_HANDLE), text);
```

```
        printf("setColor (%d)\n", text);
```

```
    }
```

```
    return 0;
```

```
}
```

```
setColor (1)
setColor (2)
setColor (3)
setColor (4)
setColor (5)
setColor (6)
setColor (7)
setColor (8)
setColor (9)
setColor (10)
setColor (11)
setColor (12)
setColor (13)
setColor (14)
setColor (15)
```

<결과 화면>

2. 파일에서 문자열 읽기

- 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

2024 Fall semester 3D computer graphics

Tech University Korea

Software College Game and multimedia engineering department

It was so hot this summer and I'm glad I was able to endure this extreme heat well.

Computer graphics class this semester is on Mondays, Tuesdays, and Wednesdays.

fall Computer Korea DEPARTMENT multi 3D

Monday Tuesday Wednesday Thursday Friday

2. 파일에서 문자열 읽기

input the command: c //--- 대문자 시작 단어 찾고 다른색으로 출력, 대문자 단어 개수를 맨 뒤에 출력

This is computer graphics warming up program. 1

input the command: d //--- 문장을 거꾸로 출력

.margorp pu gnimraw scihparg retupmoc si sihT

input the command: d //--- 다시 원래 순서로 바뀜, 다음 예제부터는 다시 원래로 바뀌는 샘플은 생략했음

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 //--- 공백 기준으로 단어들을 거꾸로 출력하기

sihT si retupmoc scihparg gnimraw pu margorp.

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

ThKs Ks computer graphKcs warmKng up.

input the command: h //--- 각 줄의 단어의 개수 출력

This is computer graphics warming up program. (7 words)

1 2 3 4 5 6 7 8 9 10 (10 words)

...

input the command: r //--- 단어 개수에 따라 오름차순, 내림차순 원래대로 순서를 바꿔 출력

Tech University Korea (3 words)

This is computer graphics warming up program. (7 words)

input the command: s Computer //--- 단어를 찾아 다른 색으로 출력하고 개수 출력

This is computer graphics warming up program.

...

Computer graphics class this semester is on Mondays, Tuesdays, and Wednesdays.

fall Computer Korea DEPARTMENT multi 3D

Total: 3 Computer

3. 저장 리스트 만들기 (구조체 데이터 사용)

- 점 (x, y, z) 데이터 값을 저장하는 리스트를 만든다. (점 데이터는 각각 정수이고, 구조체를 사용하도록 한다.)
- 최대 20개의 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 19번)까지 차고 아래칸(인덱스 0번)이 비어 있으면 다음 데이터 입력할 때는 0번에 입력된다.

** 즉, 20개의 칸을 다 채울 수 있어야 함.

3. 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
 |

- 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	<table><tr><td>9</td><td></td></tr><tr><td>8</td><td></td></tr><tr><td>7</td><td></td></tr><tr><td>6</td><td></td></tr><tr><td>5</td><td></td></tr><tr><td>4</td><td></td></tr><tr><td>3</td><td></td></tr><tr><td>2</td><td></td></tr><tr><td>1</td><td></td></tr><tr><td>0</td><td>0 1 0</td></tr></table>	9		8		7		6		5		4		3		2		1		0	0 1 0
9																					
8																					
7																					
6																					
5																					
4																					
3																					
2																					
1																					
0	0 1 0																				
2	<table><tr><td>9</td><td></td></tr><tr><td>8</td><td></td></tr><tr><td>7</td><td></td></tr><tr><td>6</td><td></td></tr><tr><td>5</td><td></td></tr><tr><td>4</td><td></td></tr><tr><td>3</td><td></td></tr><tr><td>2</td><td></td></tr><tr><td>1</td><td>0 1 1</td></tr><tr><td>0</td><td>0 1 0</td></tr></table>	9		8		7		6		5		4		3		2		1	0 1 1	0	0 1 0
9																					
8																					
7																					
6																					
5																					
4																					
3																					
2																					
1	0 1 1																				
0	0 1 0																				
3	<table><tr><td>9</td><td></td></tr><tr><td>8</td><td></td></tr><tr><td>7</td><td></td></tr><tr><td>6</td><td></td></tr><tr><td>5</td><td></td></tr><tr><td>4</td><td></td></tr><tr><td>3</td><td></td></tr><tr><td>2</td><td></td></tr><tr><td>1</td><td></td></tr><tr><td>0</td><td>0 1 0</td></tr></table>	9		8		7		6		5		4		3		2		1		0	0 1 0
9																					
8																					
7																					
6																					
5																					
4																					
3																					
2																					
1																					
0	0 1 0																				
4	<table><tr><td>9</td><td></td></tr><tr><td>8</td><td></td></tr><tr><td>7</td><td></td></tr><tr><td>6</td><td></td></tr><tr><td>5</td><td></td></tr><tr><td>4</td><td></td></tr><tr><td>3</td><td></td></tr><tr><td>2</td><td></td></tr><tr><td>1</td><td>0 1 0</td></tr><tr><td>0</td><td>1 1 1</td></tr></table>	9		8		7		6		5		4		3		2		1	0 1 0	0	1 1 1
9																					
8																					
7																					
6																					
5																					
4																					
3																					
2																					
1	0 1 0																				
0	1 1 1																				
5	<table><tr><td>9</td><td></td></tr><tr><td>8</td><td></td></tr><tr><td>7</td><td></td></tr><tr><td>6</td><td></td></tr><tr><td>5</td><td></td></tr><tr><td>4</td><td></td></tr><tr><td>3</td><td></td></tr><tr><td>2</td><td>0 1 0</td></tr><tr><td>1</td><td>1 1 1</td></tr><tr><td>0</td><td>1 0 1</td></tr></table>	9		8		7		6		5		4		3		2	0 1 0	1	1 1 1	0	1 0 1
9																					
8																					
7																					
6																					
5																					
4																					
3																					
2	0 1 0																				
1	1 1 1																				
0	1 0 1																				
6	<table><tr><td>9</td><td></td></tr><tr><td>8</td><td></td></tr><tr><td>7</td><td></td></tr><tr><td>6</td><td></td></tr><tr><td>5</td><td></td></tr><tr><td>4</td><td></td></tr><tr><td>3</td><td>1 1 0</td></tr><tr><td>2</td><td>0 1 0</td></tr><tr><td>1</td><td>1 1 1</td></tr><tr><td>0</td><td>1 0 1</td></tr></table>	9		8		7		6		5		4		3	1 1 0	2	0 1 0	1	1 1 1	0	1 0 1
9																					
8																					
7																					
6																					
5																					
4																					
3	1 1 0																				
2	0 1 0																				
1	1 1 1																				
0	1 0 1																				
7	<table><tr><td>9</td><td></td></tr><tr><td>8</td><td></td></tr><tr><td>7</td><td></td></tr><tr><td>6</td><td></td></tr><tr><td>5</td><td></td></tr><tr><td>4</td><td></td></tr><tr><td>3</td><td>1 1 0</td></tr><tr><td>2</td><td>0 1 0</td></tr><tr><td>1</td><td>1 1 1</td></tr><tr><td>0</td><td></td></tr></table>	9		8		7		6		5		4		3	1 1 0	2	0 1 0	1	1 1 1	0	
9																					
8																					
7																					
6																					
5																					
4																					
3	1 1 0																				
2	0 1 0																				
1	1 1 1																				
0																					
8	<table><tr><td>9</td><td></td></tr><tr><td>8</td><td></td></tr><tr><td>7</td><td></td></tr><tr><td>6</td><td></td></tr><tr><td>5</td><td></td></tr><tr><td>4</td><td></td></tr><tr><td>3</td><td></td></tr><tr><td>2</td><td>1 1 1</td></tr><tr><td>1</td><td>1 1 0</td></tr><tr><td>0</td><td>0 1 0</td></tr></table>	9		8		7		6		5		4		3		2	1 1 1	1	1 1 0	0	0 1 0
9																					
8																					
7																					
6																					
5																					
4																					
3																					
2	1 1 1																				
1	1 1 0																				
0	0 1 0																				