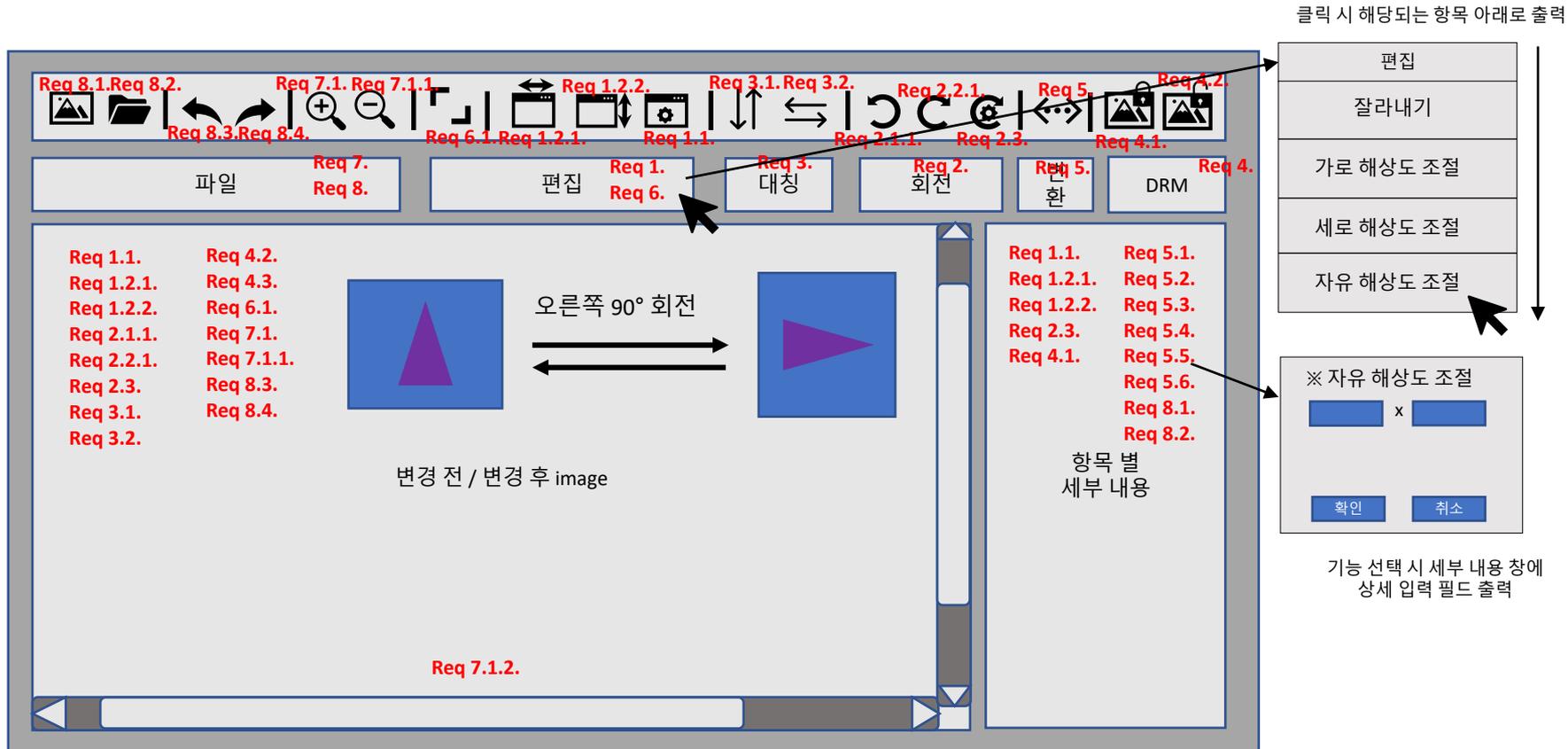




Another Photo Editor

201411257 강정모
201411307 이한강
201611284 이유진

#1. 최종 산출물 모양.



#2. 1차 구현물 데모

(1) BMP decoder

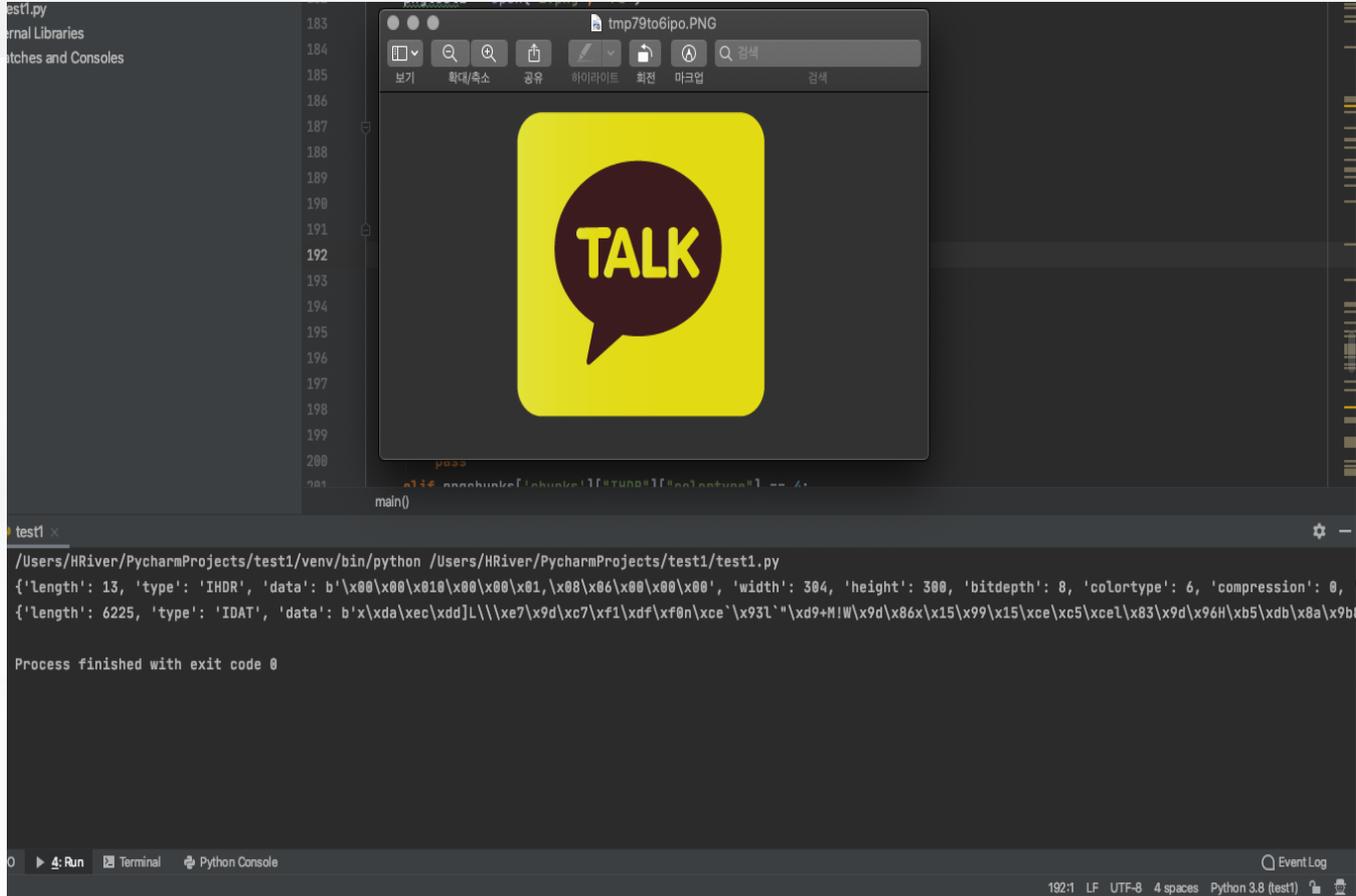
```
(base) C:\Users\이유진\PycharmProjects\im>bmp_decoder.py
file size : 8355894
buffer reserved1 : 0
buffer reserved2 : 0
data start point : 54
DIB header size : 40
width : 2048
height : 1360
color plane : 1
but for pixel : 24
compression : 0
raw size : 0
horizon pixel/meter : 3780
vertical pixel/meter : 3780
number of colors : 0
important number of colors : 0
[[ 7 7 7]
 [ 7 7 7]
 [ 6 6 6]
 ...
 [ 35 35 1]
 [ 37 34 1]
 [ 37 34 1]]

[[ 6 6 6]
 [ 6 6 6]
 [ 5 5 5]
 ...]
```



#2. 1차 구현물 데모

(2) PNG decoder



The screenshot shows a PyCharm IDE with a dark theme. The main window displays a yellow speech bubble with the word "TALK" in white. The background is a dark gray IDE interface. The left sidebar shows a file explorer with "test1.py" selected. The bottom panel shows a terminal window with the following output:

```
test1 <x  
/Users/HRiver/PycharmProjects/test1/venv/bin/python /Users/HRiver/PycharmProjects/test1/test1.py  
{'length': 13, 'type': 'IHDR', 'data': b'\x00\x00\x01\x00\x00\x01,\x00\x06\x00\x00\x00', 'width': 304, 'height': 300, 'bitdepth': 8, 'colortype': 6, 'compression': 0,  
{'length': 6225, 'type': 'IDAT', 'data': b'x\xda\xec\xdd]L\\|xe7\x9d\xc7\xf1\xdf\xfdn\xce'\x93l"'\xd9+M!W\x9d\x86x\x15\x99\x15\xce\xcc5\xceL\x83\x9d\x96H\xb5\xdb\x8a\x9b5  
Process finished with exit code 0
```

The bottom status bar shows "192:1 LF UTF-8 4 spaces Python 3.8 (test1)".

#2. 1차 구현물 데모

(3) 90° 단위 회전, 상하/좌우 대칭
원본



(1)



(2)



(3)



(4)

(1)

```
(base) C:\Users\이유진\PycharmProjects\im>rotate.py  
방향을 입력하세요 : l  
각도를 입력하세요 : 90
```

(2)

```
(base) C:\Users\이유진\PycharmProjects\im>rotate.py  
방향을 입력하세요 : r  
각도를 입력하세요 : 90
```

(3)

```
(base) C:\Users\이유진\PycharmProjects\im>symmetry.py  
방향을 입력하세요 (ud:상하, rl:좌우) : ud
```

(4)

```
(base) C:\Users\이유진\PycharmProjects\im>symmetry.py  
방향을 입력하세요 (ud:상하, rl:좌우) : rl
```

#3. System Test / Pass & Fail

Testcase Set :

(1) 800 * 600 jpg color image	(4) 512 * 512 png color image
(2) 1280 * 1660 jpg color image	(5) 221 * 228 bmp gray-scale image
(3) 64 * 64 png color image	(6) 2048 * 1360 bmp color image

Spec	input	Expected Output	Pass / Fail	comment
1.1	(3), 가로 520, 세로 230 해상도	화면에 520 * 230 해상도를 가진 png color image 출력	F	미구현
1.2.1.	(2), 가로 1600 해상도	화면에 1600 * 2152 해상도를 가진 jpg color image 출력	F	
1.2.2.	(4), 세로 539 해상도	화면에 539 * 539 해상도를 가진 png color image 출력	F	
2.1.	(6), 왼쪽 회전 1번	화면에 왼쪽으로 90° 회전된 1360 * 2048 bmp color image 출력	P	구현 완료
2.2.	(5), 오른쪽 회전 3번	화면에 오른쪽으로 270° 회전된 228 * 221 bmp color image 출력	P	구현 완료
2.3.	(4), 왼쪽 회전 27°	화면에 왼쪽으로 27° 회전된 381 * 381 png color image 출력	△	자유 각도 회전 구현 완료 축소 / cropping 구현 필요
3.1.	(1), 상하 대칭	화면에 상하 대칭된 800 * 600 jpg color image 출력	△	구현 완료 jpg decoding 문제 해결 필요
3.2.	(5), 좌우 대칭	화면에 좌우 대칭 된 221 * 228 bmp gray-scale image 출력	P	구현 완료
4.1.	(2), hello, world! 입력	다른 image editor에서는 열리지 않는 암호화된 (2) image file	△	현재 SHA / AES 256 구현 중
4.2.	4.1.의 expected output, hello, world! 입력	다른 image editor에서도 열리는 복호화된 (2) image file	F	미구현
5.1.	(1), jpg → bmp 선택	800 * 600 해상도의 bmp color image	F	bmp encoding 구현 필요 jpg decoding 구현 중
5.2.	(2), jpg → png 선택	1280 * 1660 해상도의 png color image	F	jpg decoding 구현 중 png encoding 구현 필요

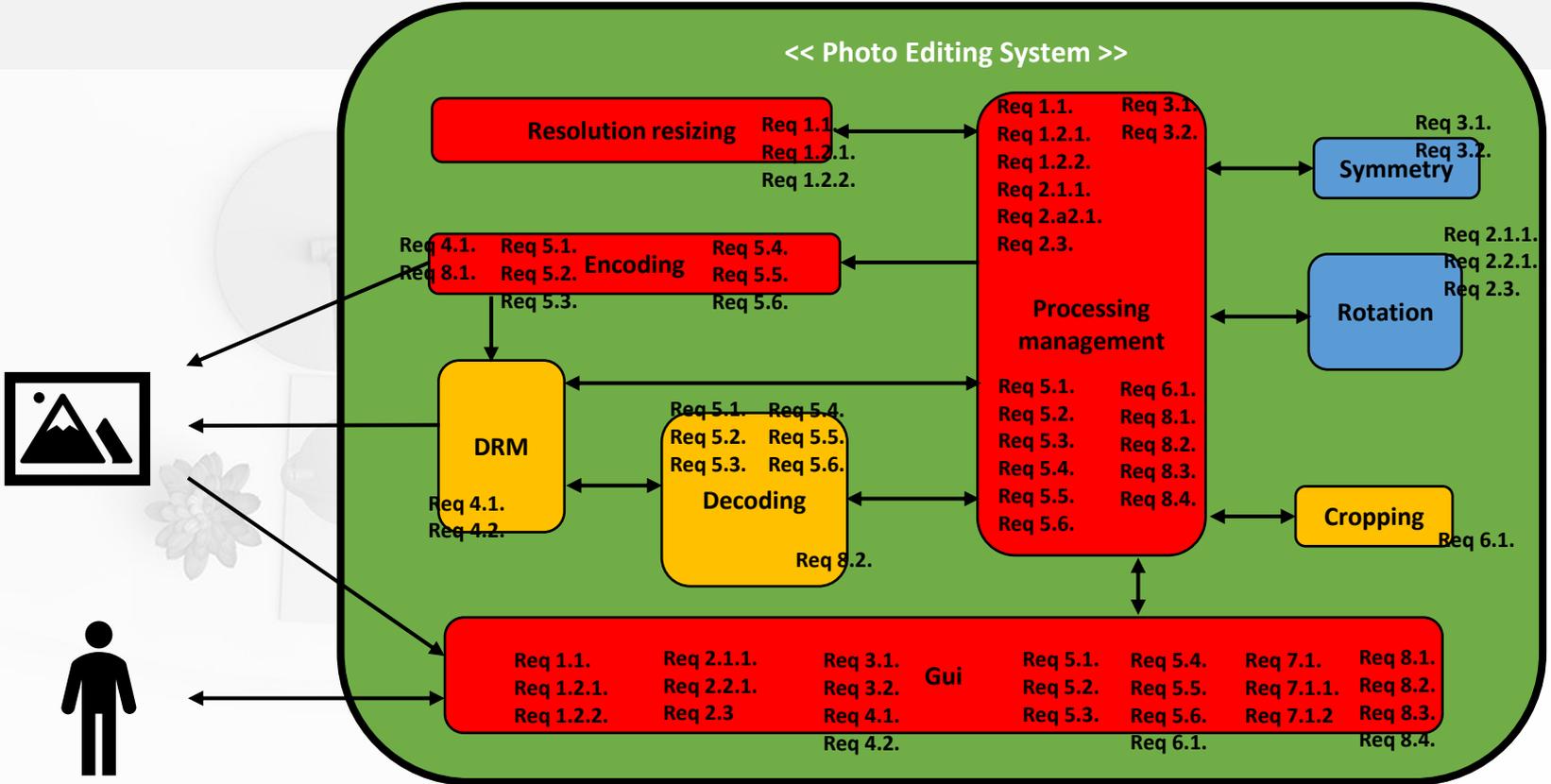
#3. System Test / Pass & Fail

Testcase Set :

(1) 800 * 600 jpg color image	(4) 512 * 512 png color image
(2) 1280 * 1660 jpg color image	(5) 221 * 228 bmp gray-scale image
(3) 64 * 64 png color image	(6) 2048 * 1360 bmp color image

Spec	input	Expected Output	Pass / Fail	comment
5.3.	(3), png → bmp 선택	64 * 64 해상도의 bmp color image	F	bmp encoding 구현 필요
5.4.	(4), png → jpg 선택	512 * 512 해상도의 jpg color image	F	png decoding 구현 중 jpg encoding 구현 필요
5.5.	(5), bmp → png 선택	221 * 228 해상도의 png gray-scale image	F	png encoding 구현 필요
5.6.	(6), bmp → jpg 선택	2048 * 1360 해상도의 jpg color image	F	bmp decoding 구현 완료 jpg encoding 구현 필요
6.1.	(1), (1) 내부를 마우스 커서로 drag 한 영역	화면에 Drag한 영역 만큼의 해상도를 가지는 (1) 내부의 jpg color image 출력	△	cropping 구현 중 GUI 구현 필요
7.1.	(3)을 띄운 출력 창 선택 3번 확대	화면에 (3)이 300% (3배) 확대된 image 출력	F	GUI 구현 필요
7.1.1.	키보드 → 화살표	확대된 image가 출력 창 크기를 초과하는 경우 확대된 (3)의 오른쪽 부분으로 이동하여 출력	F	
7.1.2.	(3)을 3번 확대한 출력 창 선택 1번 축소	화면에 (3)이 200% (2배) 확대된 Image 출력 (확대 단계 1단계 감소)	F	
8.1.	4.2. expected output, 원하는 directory, file name, file extension	4.2.의 expected output을 지정한 directory에 지정한 file extension format과 file name으로 저장	F	
8.2.	(6), (6) file이 있는 directory 경로	(6) Image file을 창에 출력	F	
8.3.	6.1. expected output	화면에 2.3. (1) 출력	F	
8.4.	8.3. expected output	화면에 6.1. expected output	F	

#4. Component / Architecture Diagram



#5. Pass & Fail Revision + 2nd iteration

Testcase Set :

- (1) 800 * 600 jpg color image
- (2) 1280 * 1660 jpg color image
- (3) 64 * 64 png color image

- (4) 512 * 512 png color image
- (5) 221 * 228 bmp gray-scale image
- (6) 2048 * 1360 bmp color image

Spec	input	Expected Output	DeadLine	Revision
1.1	(3), 가로 520, 세로 230 해상도	화면에 520 * 230 해상도를 가진 png color image 출력	5/28	Png encoder 구현 완료 후 구현 할 예정
1.2.1.	(2), 가로 1600 해상도	화면에 1600 * 2152 해상도를 가진 jpg color image 출력	5/28	Png encoder 구현 완료 후 구현 할 예정
1.2.2.	(4), 세로 539 해상도	화면에 539 * 539 해상도를 가진 png color image 출력	5/28	Png encoder 구현 완료 후 구현 할 예정
2.1.	(6), 왼쪽 회전 1번	화면에 왼쪽으로 90° 회전된 1360 * 2048 bmp color image 출력	5/18	완료
2.2.	(5), 오른쪽 회전 3번	화면에 오른쪽으로 270° 회전된 228 * 221 bmp color image 출력	5/18	완료
2.3.	(4), 왼쪽 회전 27°	화면에 왼쪽으로 27° 회전된 381 * 381 png color image 출력	5/28	기능과 Processing management 구현이 완료된 후 테스트할 예정
3.1.	(1), 상하 대칭	화면에 상하 대칭된 800 * 600 jpg color image 출력	5/28	기능 구현 완료, Jpg color image 작업이 완료된 후 테스트할 예정
3.2.	(5), 좌우 대칭	화면에 좌우 대칭 된 221 * 228 bmp gray-scale image 출력	5/18	완료
4.1.	(2), hello, world! 입력	다른 image editor에서는 열리지 않는 암호화된 (2) image file	5/28	SHA 256, AES 256 구현 완료 후 테스트 예정
4.2.	4.1.의 expected output, hello, world! 입력	다른 image editor에서도 열리는 복호화된 (2) image file	6/4	GUI의 저장 기능과 함께 구현할 예정
5.1.	(1), jpg → bmp 선택	800 * 600 해상도의 bmp color image	5/28	jpeg의 경우 가장 많이 사용되는 JFIF format에 대해서만 decoding 구현 할 예정, Bmp encoding 구현할 예정
5.2.	(2), jpg → png 선택	1280 * 1660 해상도의 png color image	5/28	jpeg의 경우 가장 많이 사용되는 JFIF format에 대해서만 decoding 구현 할 예정

#5. Pass & Fail Revision + 2nd iteration

Testcase Set :

- (1) 800 * 600 jpg color image
- (2) 1280 * 1660 jpg color image
- (3) 64 * 64 png color image

- (4) 512 * 512 png color image
- (5) 221 * 228 bmp gray-scale image
- (6) 2048 * 1360 bmp color image

Spec	input	Expected Output	DeadLine	Revision
5.3.	(3), png → bmp 선택	64 * 64 해상도의 bmp color image	5/28	bmp encoding 구현할 예정
5.4.	(4), png → jpg 선택	512 * 512 해상도의 jpg color image	5/28	png RGBA 구현 완료 / Png grayscale 구현 예정 jpeg의 경우 JFIF format에 대해서 Encoding 구현 할 예정
5.5.	(5), bmp → png 선택	221 * 228 해상도의 png gray-scale image	5/28	Png grayscale 구현 예정, png encoder 구현 예정
5.6.	(6), bmp → jpg 선택	2048 * 1360 해상도의 jpg color image	5/28	Bmp color-scale 구현 완료, jpeg의 경우 JFIF format에 대해서 Encoding 구현 할 예정
6.1.	(1), (1) 내부를 마우스 커서로 drag 한 영역	화면에 Drag한 영역 만큼의 해상도를 가지는 (1) 내부의 jpg color image 출력	6/4	자르기 기능과 GUI 구현 예정
7.1.	(3)을 띄운 출력 창 선택 3번 확대	화면에 (3)이 300% (3배) 확대된 image 출력	6/4	확대 기능과 GUI 구현 예정
7.1.1.	키보드 → 화살표	확대된 image가 출력 창 크기를 초과하는 경우 확대된 (3)의 오른쪽 부분으로 이동하여 출력	6/4	이동 기능과 GUI 구현 예정
7.1.2.	(3)을 3번 확대한 출력 창 선택 1번 축소	화면에 (3)이 200% (2배) 확대된 Image 출력 (확대 단계 1단계 감소)	6/4	축소 기능과 GUI 구현 예정
8.1.	4.2. expected output, 원하는 directory, file name, file extension	4.2.의 expected output을 지정한 directory에 지정한 file extension format과 file name으로 저장	6/4	저장 기능과 GUI 구현 예정
8.2.	(6), (6) file이 있는 directory 경로	(6) Image file을 창에 출력	6/4	이미지 출력 기능과 GUI 구현 예정
8.3.	6.1. expected output	화면에 2.3. (1) 출력	6/4	Undo 기능과 GUI 구현 예정
8.4.	8.3. expected output	화면에 6.1. expected output	6/4	Redo 기능과 GUI 구현 예정