



Object-Oriented Programming Report

Assignment 1-2

Professor	Donggyu Sim
Department	Computer engineering
Student ID	2022202061
Name	Seoeun Yang
Class (Design / Laboratory)	1 / B (미수강시 0로 표기)
Submission Date	2023. 3. 24

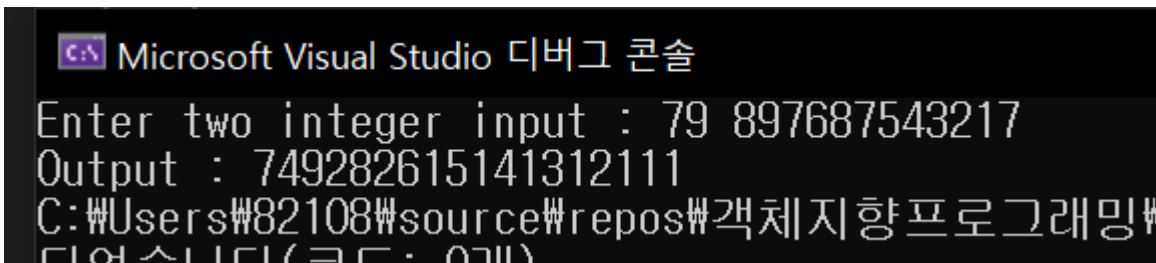
Program 1

□ 문제 설명

먼저 숫자 2 개를 입력 받는다. 첫번째로 입력 받은 숫자의 첫번째 자리 수(A)부터 확인하는데 입력 받은 숫자 2 개에 A 가 몇 번 나오는지 확인해 출력한다. 이후 두번째 자리 수로 이동하여 몇 번 나오는지 확인해. 입력 받은 두 수가 끝날 때까지 출력하며 중복된 숫자는 무시한다. 예를 들어 798, 485 를 입력했다면 출력값은 7191824151 이 된다. 7 - 1 개, 9 - 1 개, 8 - 2 개, 4 - 1 개, 5 - 1 개를 의미한다.

각 숫자마다 최대 20 자리까지 입력받을 수 있도록 변수를 선언해주는데 char 형으로 선언해준다. 숫자의 자릿수 하나하나 배열에 넣어줘야 되기 때문에 int 형으로 선언하지 않는다. 선언된 배열은 쓰레기값이 들어있으므로 -1 로 초기화를 해준다. 굳이 -1 인 이유는 숫자를 입력 받을 때 -1 은 절대 입력받지 않기 때문이다. 정수를 입력 받았다면 각 숫자열의 길이를 구해 두 숫자열을 붙여준다. (배열 att) char 형으로 입력받은 수를 int 형으로 붙일 것이기 때문에 -48 을 해준다. 또한 탐색했던 숫자를 다시 탐색하는 오류를 피하기 위해 10 칸짜리 int 형 배열을 선언해 초기화해준다. (배열 num) 본인의 숫자 탐색 구조는 다음과 같다. 배열 att 의 길이만큼 반복문을 통해 숫자를 탐색한다. 탐색할 숫자가 몇 개 있는지 반복문을 통해 확인한 후 만약 이전에 탐색했던 숫자라면 continue 를 통해 다음 숫자로 넘어간다. 중복이 아니라면 배열 num 에 숫자를 저장하고 탐색한 숫자, 숫자가 나온 횟수를 출력한다.

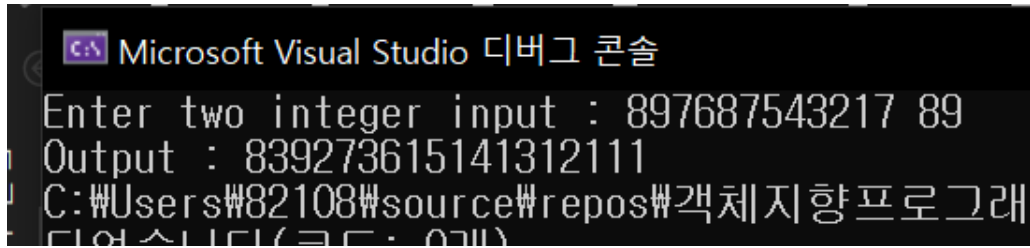
□ 결과 화면



```
Microsoft Visual Studio 디버그 콘솔
Enter two integer input : 79 897687543217
Output : 749282615141312111
C:\Users\82108\source\repos\객체지향프로그래밍\
디버깅합니다(코드: 0개)
```

문제지에 나와있던 첫번째 예시

79, 897687543217 입력, 749282615141312111 출력

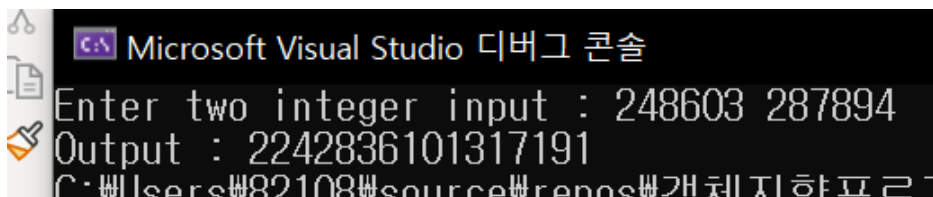


```
Microsoft Visual Studio 디버그 콘솔
Enter two integer input : 897687543217 89
Output : 839273615141312111
C:\Users\82108\source\repos\객체지향프로그래밍\obj\Debug\...
```

문제지에 나와있던 두번째 예시

897687543217, 89 입력, 839273615141312111 출력

첫번째 예시와 같은 입력값이 들어감에도 출력값이 다른 이유는 첫번째 예시에서 2 번째로 입력되었던 수가 두번째 예시에서는 첫번째로 입력되었기 때문이다. 숫자의 순서가 바뀌기 때문에 출력값의 홀수자리의 순서가 바뀐다. 물론 두번째로 입력받은 수도 첫번째의 다른 수와 다르기 때문에 출력값의 숫자 개수도 달라진다. 이 예시를 통해 같은 수를 입력해도 순서가 달라지면 출력값이 달라짐을 알 수 있다.



```
Microsoft Visual Studio 디버그 콘솔
Enter two integer input : 248603 287894
Output : 2242836101317191
C:\Users\82108\source\repos\객체지향프로그래밍\obj\Debug\...
```

248603, 287894 입력, 2242836101317191 출력

입력값에 0 이 있을 경우 0 의 개수 또한 출력값에서 표시해줘야 한다.

□ 고찰

먼저 배열의 길이를 구하는 것이 어려웠다. stl 을 사용할 수 없기 때문에 사이즈를 통해 구하려고 했으나 방법이 잘못된 것인지 본인의 실수였던 것인지 구할 수 없어 다른 방법을 통해 구했다. 먼저 선언된 배열을 -1 로 모두 초기화해준 후 입력을 받는다. 그 후 반복문을 돌며 배열 칸마다 저장된 값이 -1 이 아니라면 길이 변수를 1 씩 증가하게 했다. 반복문이 종료된 후 null 의 크기를 없애주기 위해 -1 을 해주면 숫자 배열의 크기를 구할 수 있었다. 본인은 각 숫자의 개수를 파악하자마자 출력해줬는데 새로운 배열을 선언해 차례대로 저장한 후 출력해줘도 괜찮을 것 같다. 문제지에서 제시해주는 예시에는 0 이 없어 출력값에 0 이 없지만 만약 0 이 포함된 수를 입력한다면 출력값에 0 의 개수 또한 포함시켜야 한다.

Program 2

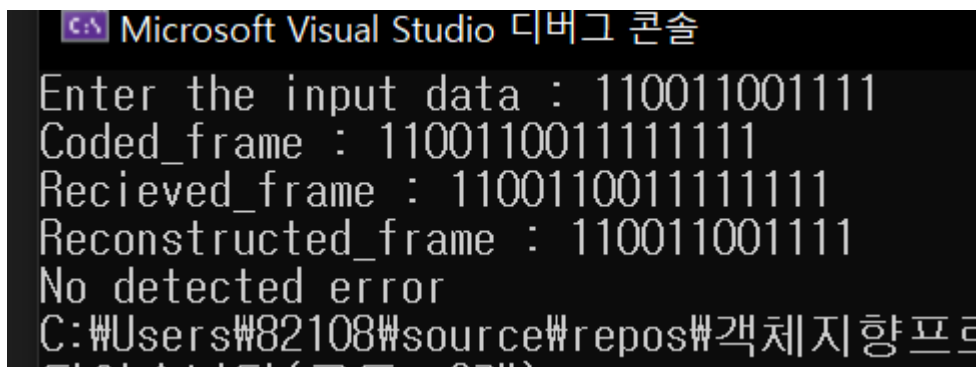
□ 문제 설명

데이터를 전송할 때 데이터에 오류가 생겼는지 확인하는 프로그램이다.

12 비트짜리의 2 진수를 입력 받고 FCS 를 붙여 CRC 를 진행한다. 16 비트 2 진수를 제공된 divisor 와 XOR 연산을 통해 나머지를 구하고 입력 받았던 2 진수와 결합해 전송한다. 전송 과정에서 생기는 오류를 Transmission_Channel 함수를 통해 구현하고 Reciever 에서 전송받은 16 비트 2 진수와 divisor 로 다시 한 번 XOR 연산을 해 오류가 생겼는지 확인한다. 나머지가 0 이면 오류가 생기지 않은 것이며 0 이 아니면 한 개 이상의 오류가 생겼다는 뜻이다.

char 형으로 2 진수를 입력 받아 정수형 배열에 넣어주고 compare 변수에 2 진수 5 자리를 넣어 divisor 와 XOR 연산 후 변수가 0 으로 시작한다면 shift 함수를 호출한다. srand() 함수를 통해 프로그램을 실행할 때마다 다른 경우의 수가 나오도록 구현했다.

□ 결과 화면



```
C:\N Microsoft Visual Studio 디버그 콘솔
Enter the input data : 110011001111
Coded_frame : 1100110011111111
Recieved_frame : 1100110011111111
Reconstructed_frame : 110011001111
No detected error
C:\Users\82108\source\repos\객체지향프로그래밍\obj\Debug\...
```

input: 110011001111

divisor: 11011

Reconstructed_frame 이 input 과 동일하므로 에러가 발생하지 않음을 의미한다.

```
Microsoft Visual Studio 디버그 콘솔
Enter the input data : 110011001111
Coded_frame : 1100110011111111
Recieved_frame : 1101110011111111
Reconstructed_frame : 110111001111
Reciever has detected error
C:\Users\82108\source\repos\객체지향프
```

input: 110011001111

divisor: 11011

Received_frame 과 Coded_frame 이 다르다. 왼쪽으로부터 3 번째 자리의 비트수가 다른데, 이는 Transmission_Channel 함수에서 오류가 발생했음을 의미한다. 비트가 바뀌었으므로 Reconstructed_frame 또한 input 과 달라지며 오류가 발생했음을 알려준다.

❑ 고찰

CRC 프로그램에 대해 전혀 아는 게 없어서 어떤 식으로 문제에 접근해야 할 지 고민이 길었다. XOR 연산 자체가 익숙하지 않아 알고리즘을 짜는 것이 매우 어려웠다. XOR operator, shift operator 을 사용해 구현했으며 그 과정에서 char 형 배열과 int 형 배열을 동시에 사용해야 했다. shift operator 을 사용하기 위해 char 형 배열에 저장되어 있는 2 진수를 10 진수로 변환했다. 이러한 일련의 알고리즘을 생각해내는 데에 상당한 시간이 걸렸다. 처음 코드를 짤 땐 XOR operator, shift operator 없이 구현했었는데 연산자를 사용해 다시 구현해야 했다. 계속 생각했던 알고리즘에서 벗어나 새로운 알고리즘을 짜야 했기에 정말 힘들었다.

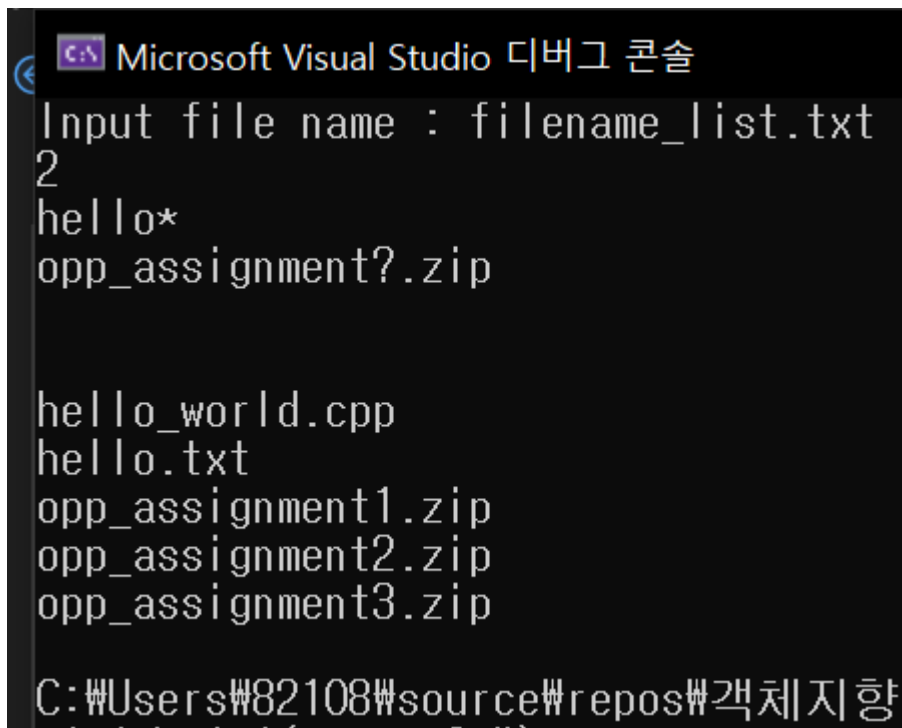
Program 3

□ 문제 설명

텍스트 파일 안에 일련의 파일 이름 리스트를 작성한다. wildcard pattern 에 따라 파일을 검색하고 출력하는 프로그램이다. 열고 싶은 텍스트 파일의 이름을 입력하고, 탐색하고 싶은 파일의 형식, 횟수를 입력한다. 파일의 형식에 *이 있다면, *을 제외한 문자열이 포함되어 있는 모든 파일 이름을 출력한다. 파일의 형식에 ?가 있다면, ?를 제외한 모든 문자열이 같은 파일 이름을 출력한다.

*의 경우, * 앞에 존재한 문자열만 잘라 텍스트 파일 안에 있는 이름들을 비교해 같다면 출력했다. ?의 경우, ? 자리만 공백으로 초기화시키고 이름들을 비교해 공백 자리만 다르다면 출력했다.

□ 결과 화면

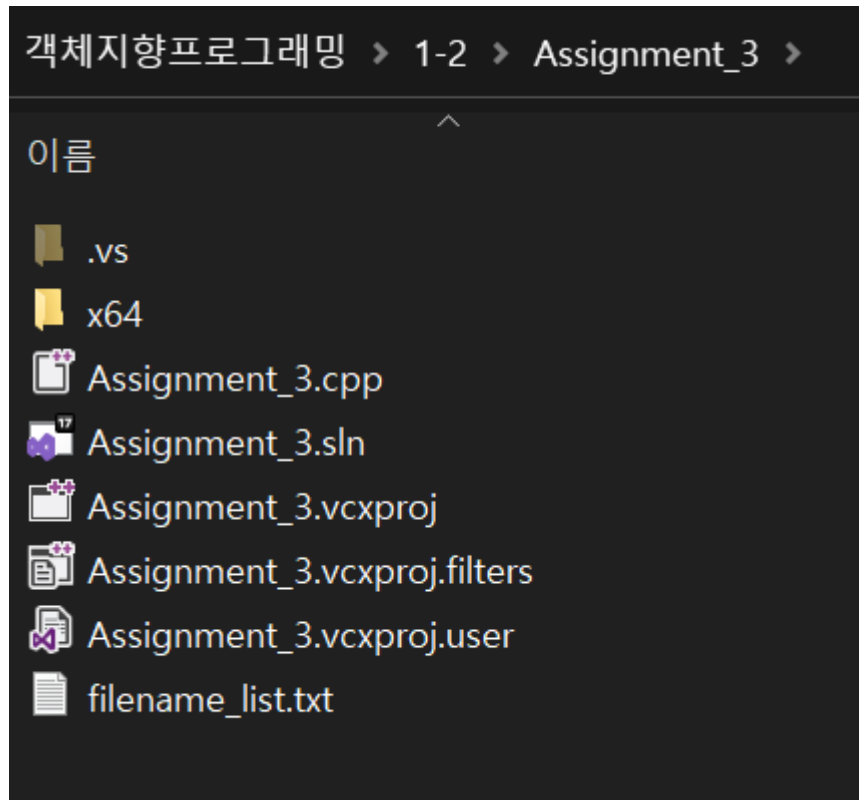


```
C:\> Microsoft Visual Studio 디버그 콘솔
Input file name : filename_list.txt
2
hello*
opp_assignment?.zip

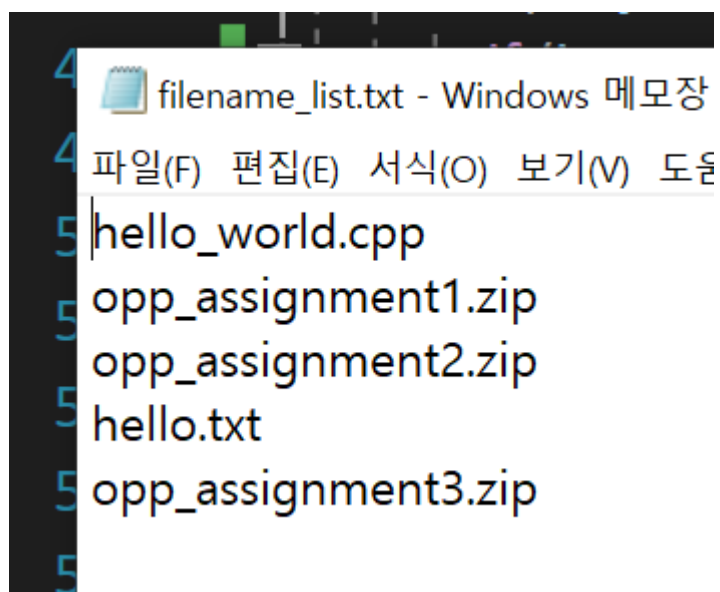
hello_world.cpp
hello.txt
opp_assignment1.zip
opp_assignment2.zip
opp_assignment3.zip

C:\Users\82108\source\repos\객체지향\파일_이름_리스트_검색_프로그램\obj\Debug\
```

문제지 예시



솔루션 파일 안에 저장되어있는 txt 파일



txt 파일 안에 저장되어 있는 문자열

```

.h Microsoft Visual Studio 디버그 콘솔
sig Input file name : filename.txt
10 no file exist
10 C:\Users\82108\source\repos\객체지향프
11 되었습니니다(코드: 0개).
11 디버깅이 중지될 때 콘솔을 자동으로

```

예외 처리

```

Microsoft Visual Studio 디버그 콘솔
Input file name : filename_list.txt
1
opp?assignment?.zip

opp_assignment1.zip
opp_assignment2.zip
opp_assignment3.zip

C:\Users\82108\source\repos\객체지향프
되었습니니다(코드: 0개).

```

hello* 를 탐색하도록 명령했기 때문에 filename_list.txt 안에 쓰여 있는 파일 이름 중 hello 가 포함되어 있는 모든 파일 이름을 출력한다.

opp_assignment?.zip 의 경우 ?가 한 개 존재하고 텍스트 파일 안에 쓰여 있는 파일 이름들 중 3 개가 ?자리에 숫자만 다르기 때문에 위에서부터 차례대로 opp_assignment1.zip,...2,...3 을 출력한다.


```
Microsoft Visual Studio 디버그 콘솔
Input file name : filename_list.txt
1
opp?assignment?.zip

opp_assignment1.zip
opp_assignment2.zip
opp_assignment3.zip

C:\Users\82108\source\repos\객체지향프
되었습니다(코드: 0개).
```

?가 한 개 이상일 경우

□ 고찰

입력의 경우, 탐색하고 싶은 횟수에 따라 입력하는 파일 형식의 개수가 달라진다. 무작정 많은 변수를 선언해줄 수 없기 때문에 N 번 탐색하고 싶다면 크기가 N*101 인 이차원 배열을 char 형으로 동적 할당해줬다. 따라서 프로그램 종료 전, 동적 할당 해제를 통해 메모리 누수를 방지해야 한다. wildcard pattern 에 의하면 탐색 종류가 4 가지가 나온다.

1. *만 있는 경우

2. ?만 있는 경우

3. ? *가 혼합되어 있는 경우

1, 2 번은 구현했고, 3 번의 경우 *이 먼저 나온다면 뒤의 ?가 필요가 없어진다.

하지만 ?가 먼저 나오고 *이 나온다면 또 다른 경우가 된다. 이 경우 * 전의 문자열을 저장하고 ?의 개수를 파악해 탐색 후 출력한다. 이 외에도 수 많은 예외가 존재하는데 *이 먼저 입력되는 경우 모든 문자열을 출력해야 되고, * 뒤에 문자가 입력되어있다면 텍스트 파일 안에서 * 뒤에 위치한 문자열과 동일한 제목을 가진

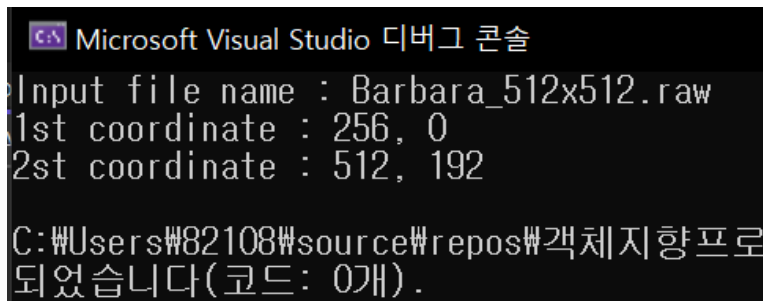
텍스트가 출력되어야 한다. 코드를 짜면서 계속 새로운 예외가 등장해 구현하기 너무 까다로웠다.

Program 4

□ 문제 설명

NXM 개의 픽셀로 이루어진 이미지가 있었을 때, 좌표를 지정해 이미지를 자르고, 좌우반전, 상하반전 시켜주는 프로그램이다. 한 픽셀은 1 바이트로 표현될 수 있는데, 0 부터 255 레벨까지 존재한다. 0 레벨은 검정색이고, 255 레벨은 하얀색이다. 이미지를 자르는 경우 두 좌표가 주어지는데 두 좌표가 그리는 사각형 안의 이미지를 저장하면 된다. 각 이미지는 새로운 이름으로 저장된다. 원본 이미지를 불러와 512*512 크기의 배열에 저장한다. 각 경우의 수도 배열을 선언해 원본 이미지의 픽셀 값을 저장하는 형식으로 프로그램을 구현했다.

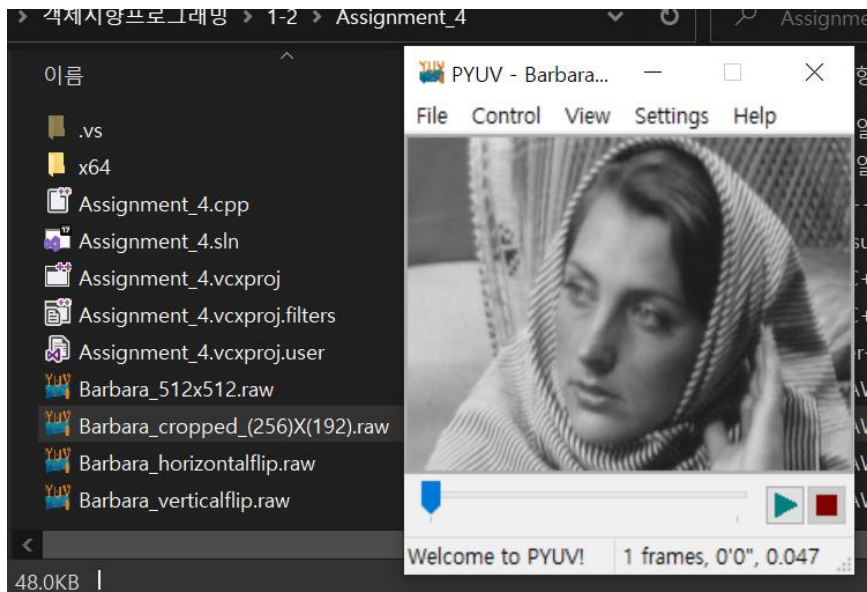
□ 결과 화면



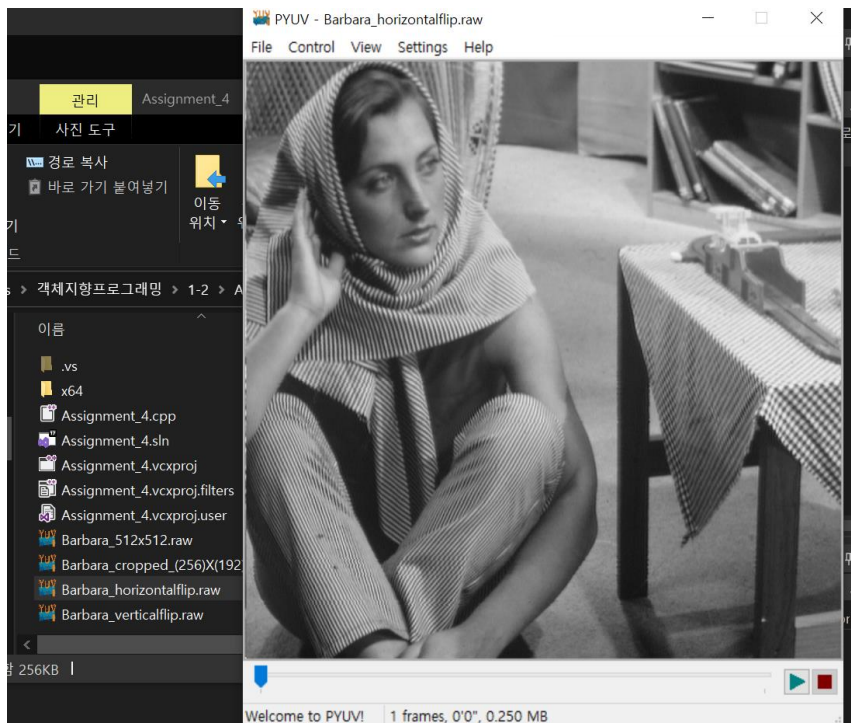
```
Microsoft Visual Studio 디버그 콘솔
Input file name : Barbara_512x512.raw
1st coordinate : 256, 0
2st coordinate : 512, 192

C:\Users\82108\source\repos\객체지향프로
되었습니다(코드: 0개).
```

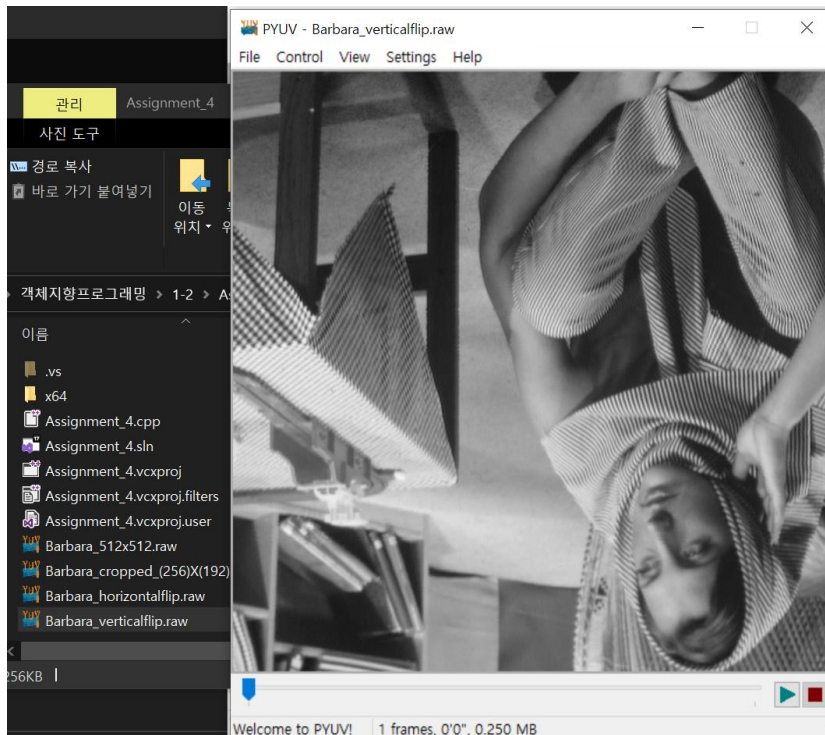
<그림 1>



<그림 2>



<그림 3>



<그림 4>

<그림 1>은 프로그램을 실행했을 때 나오는 cmd 화면이고, <그림 2>부터 차례대로 프로그램에 의해 새로 저장된 cropped image, horizontal image, vertical image 이다.

□ 고찰

이미지를 자르는 부분이 가장 어려웠다. 좌표를 입력 받았을 때, 자를 구간을 어떻게 설정해야 할 지 고민이었다. 따라서 이차원 배열에 어떠한 값을 저장할 때 이중 for 문을 돌며 왼쪽 위부터 오른쪽 대각선 아래로 저장되는 형식을 사용했다. 좌표의 크기를 비교해 스타팅 포인트의 x,y 좌표를 저장해 그 좌표를 시작으로 원본 이미지를 cropped image 에 차례대로 저장했다. 파일 입출력이 익숙치 않아 실습 자료에 나와있던 코드를 이해하는 것조차 쉽지 않았고 생길 수 있는 모든 예외(파일을 찾지 못하거나 파일을 여는 것에 실패하는 경우)를 처리하는 것이 헛갈렸다.