Object Oriented Programming (Week 3)
Assignment 1-2

2023.

KWANGWOON UNIVERSITY
DEPT. OF COMPUTER ENGINEERING

Contents

ASSIGNMENT 1-2. 1

Assignment 1-2. 1

- Write a program to display the result after applying the specific rule illustrated as below.
  - To elaborate, the program takes **two inputs** and parses the first number that appears at the beginning of the first input.
  - Then, it counts how many times this parsed number appears in both inputs.
  - The parsed number and its count of appearances become the first and second numbers of the output, respectively.
  - If the second number in the first input is different from the first number, it will be the third argument of the output, and the same procedure will be repeated until the end of the first input.
  - Once the program finishes parsing all the way to the end of the first input in the same manner, the same rule is applied to the second input to obtain the final output.

Assignment 1-2. 1

- 예시

Input	Output
79 897687543217	74928261514131211 1
897687543217 89	83927361514131211 1
897687543217 897687543217	84927662524232221 2

Assignment 1-2. 1

- 예시

Input	Output
7 9 897687543217	74928261514131211 1

7의 개수: **4**

Assignment 1-2. 1

- 예시

Input	Output
79 897687543217	74928261514131211 1

```
7                    7989768754321 7
749                  798976875432 17
```

7의 개수: **4**

9의 개수: **2**

Assignment 1-2. 1

- 예시

Input	Output
79 897687543217	74928261514131211

7	7989768754321 7	7의 개수: **4**
749	798 9 7687543217	9의 개수: **2**
74928	7989768 7 543217	8의 개수: **2**

Assignment 1-2. 1

- 예시

Input	Output
79 897687543217	74928261514131211 1

PASS 앞의 나온 숫자와 중복될 경우, pass

7	7989768754321 7	7의 개수: **4**
749	798 9 7687543217	9의 개수: **2**
74928	798976 8 7543217	8의 개수: **2**

Assignment 1-2. 1

- 예시

Input	Output
79 8976875432 17	74928261514131 2111

PASS 앞의 나온 숫자와 중복될 경우, pass

7	7989768754321 7	7의 개수: **4**
749	798976875432 17	9의 개수: **2**
74928	7989768754321 7	8의 개수: **2**

Assignment 1-2. 1

- 예시

Input	Output
79 897687543217	74928261514131211 1

7	7989768754321 7	7의 개수: **4**
749	798 9 7687543217	9의 개수: **2**
74928	7989768 7 543217	8의 개수: **2**
7492826	7989768 7 543217	6의 개수: **1**

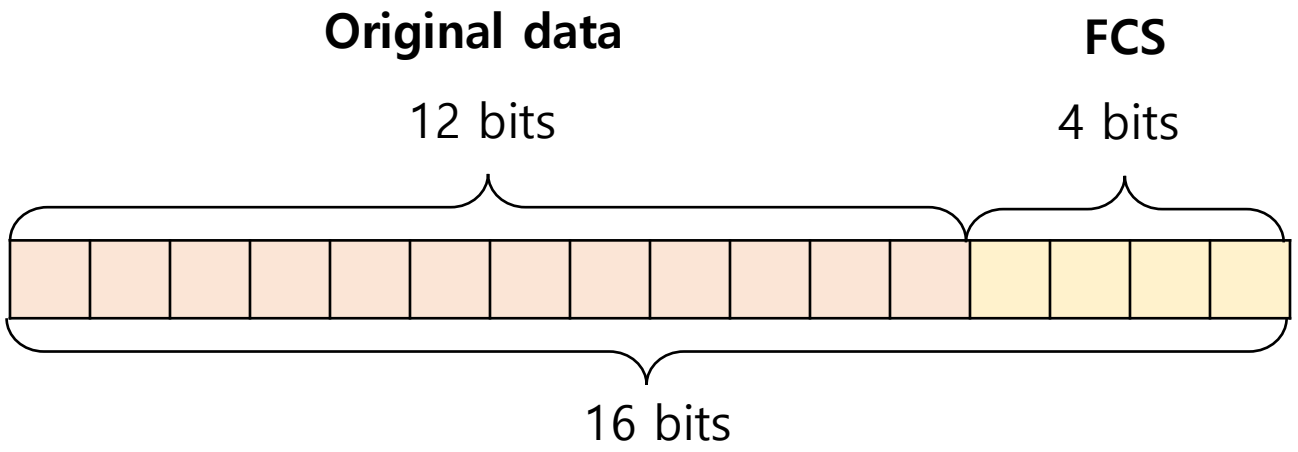
ASSIGNMENT 1-2. 2

Assignment 1-2. 2

- Write a program to add **CRC (Cyclic redundancy check) code** for a **sender**, which in turn it helps the **receiver** find whether errors appeared or not in data during the transmission.
- The program includes **Sender**, **Transmission_Channel** and **Receiver** function.
  - CRC (Cyclic redundancy check, 순환 중복 검사)
    - 네트워크와 저장매체 등을 통하여 데이터를 전송할 때, 전송된 데이터에 오류가 있는지 확인하기 위한 오류 검출 부호
    - 데이터를 전송하기 전에 주어진 데이터의 값에 따라 CRC 값을 붙여 전송하고, 데이터 전송이 끝난 후 전송받은 데이터의 값으로 다시 CRC를 계산함
    - 이 때, CRC 값이 0이 아닐 경우, 데이터 전송 과정 중 오류가 발생한 것으로 판단함

Assignment 1-2. 2

- Sender is supposed to get **12-digit binary data** from a user and **5-digit specified binary divisor** (0b11011).
- After that, Sender calculates frame check sequence (FCS) from a 12-bit block and appends FCS to the block.
- Next, the **16-bit frame**, which is composed of a block (12-bits) and FCS (4 bits for the sake of error detection in data bits), is sent to Transmission_Channel.



Assignment 1-2. 2

- There is a **5%** chance of an error occurring for every single bit in a frame by Transmission_Channel function.
  - - Input : const char* coded_frame
- Receiver must perform data verification about an incoming received frame by checking if 16-bit frame is exactly divisible by the predetermined number, which is the same divisor adopted by Sender.
- Then, it prints the result of whether there is a remainder or not. Note that you should make the best use of **shift operators** and **ExclusiveOR** bit operator.
  - - Input : const char* received_frame

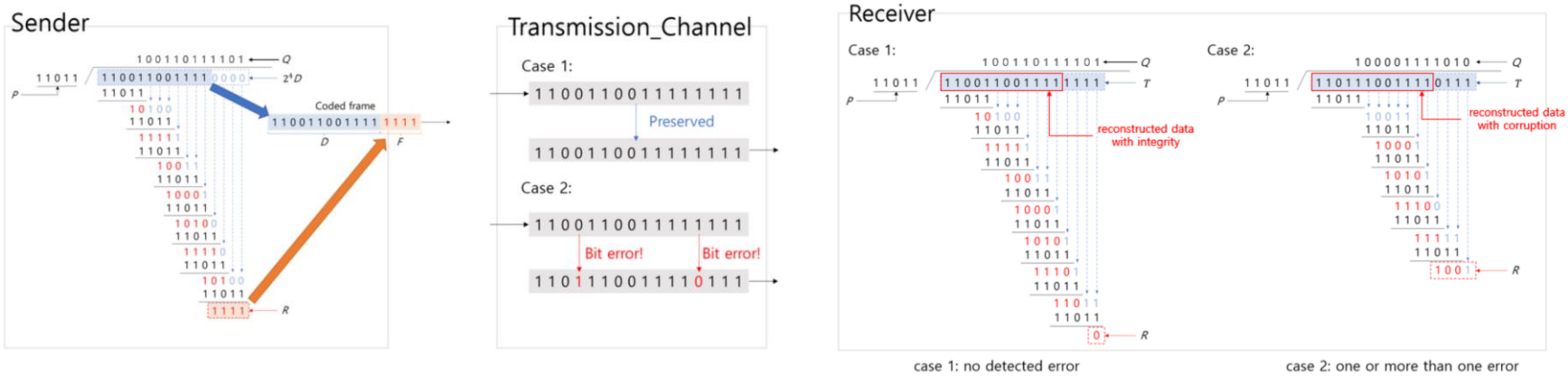
Assignment 1-2. 2

- Adopted modulo 2 arithmetic can be implemented with **shift** and **Exclusive-OR (XOR)** operation.
  - T = 16-bit frame to be transmitted
    - 전송 데이터 (ex. n bits)
  - D = 12-bit block of data; the first 12 bits of T
    - 원본 데이터 (ex. k bits)
  - F = 4-bit FCS; the last 4 bits of T
    - 잉여 데이터 (ex. n-k bits)
  - P = Predetermined divisor of 5 bits (generator)
    - 미리 정의된 CRC 다항식
  - Q = Quotient
    - 나눗셈 몫
  - R = Remainder of 4 bits
    - 나머지

Assignment 1-2. 2

- The illustration above has shown two possible cases of result.
- The case 1 describes the process when no errors occurred by Transmission_Channel function, while the case 2 describes the process when two errors has occurred by Transmission_Channel function.

Input	Output
Data: 110011001111	Coded frame: 1100110011111111 Received frame: 1100110011111111 Reconstructed data: 110011001111 No detected error
Data: 110011001111	Coded frame: 1100110011111111 Received frame: 1101110011110111 Reconstructed data: 110111001111 Receiver has detected error

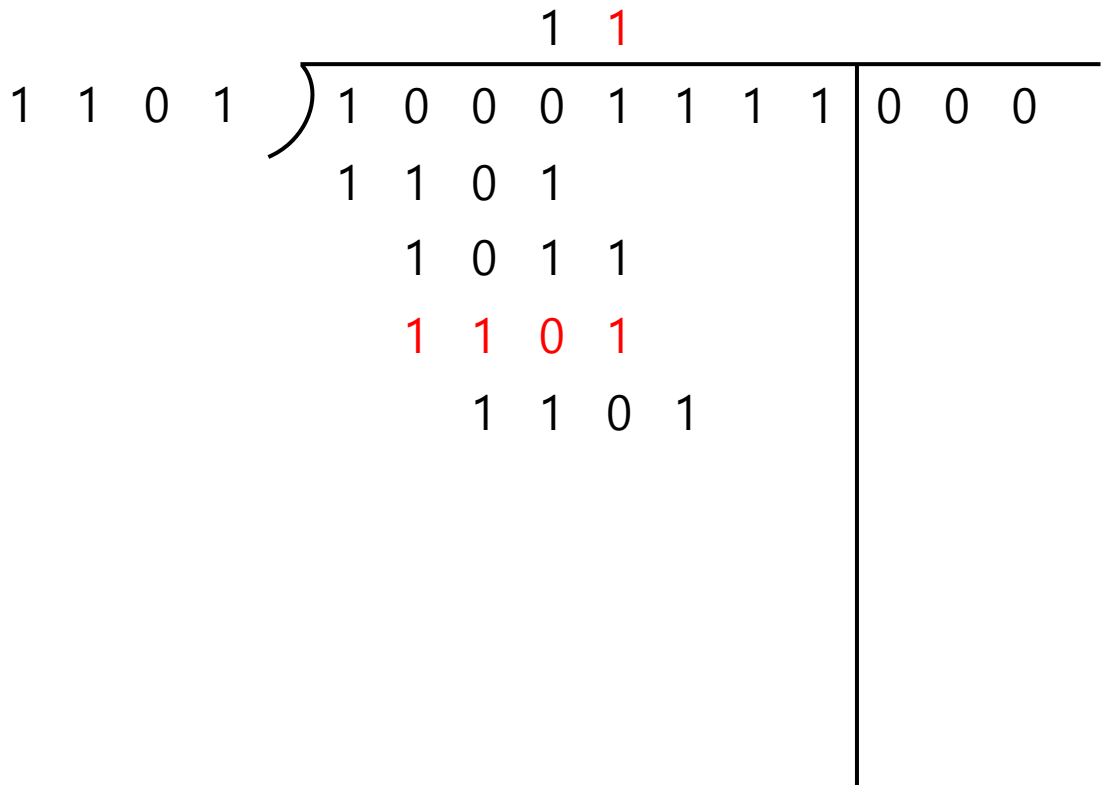


Assignment 1-2. 2 (Sender)

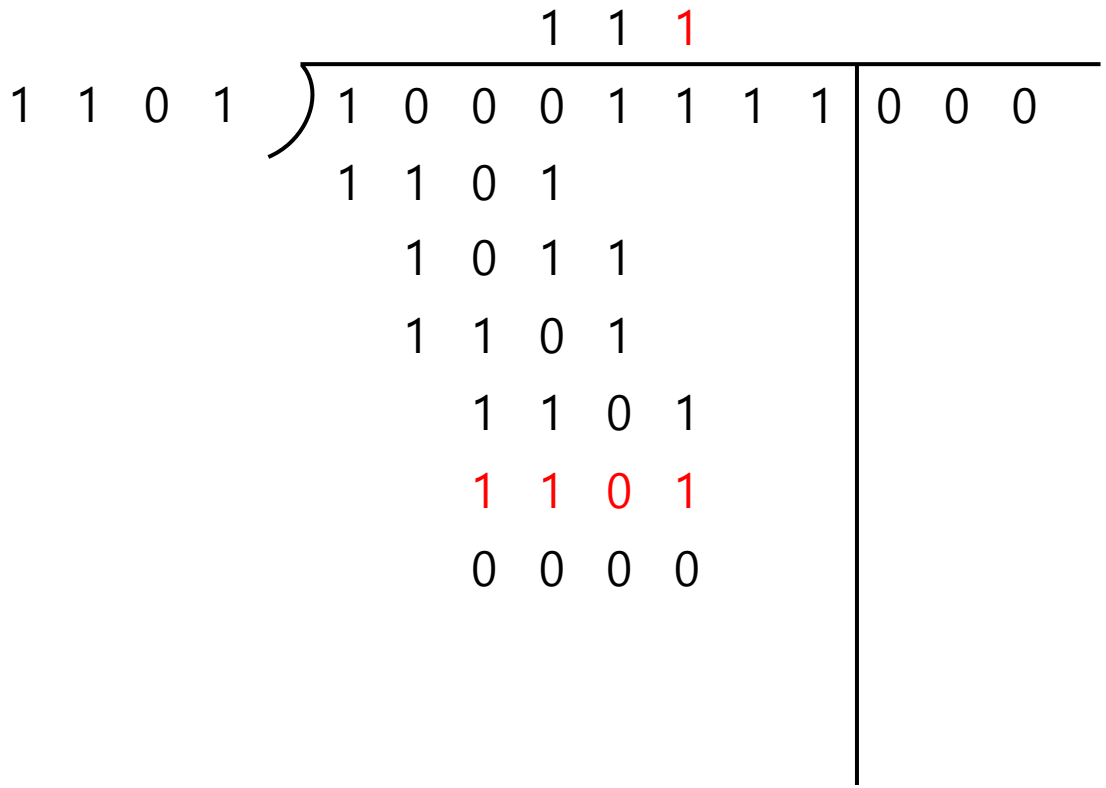
XOR연산

```
                1
          ┌──────────────
1 1 0 1 ) 1 0 0 0 1 1 1 1 0 0 0
          1 1 0 1
          ─────────
            1 0 1 1
```

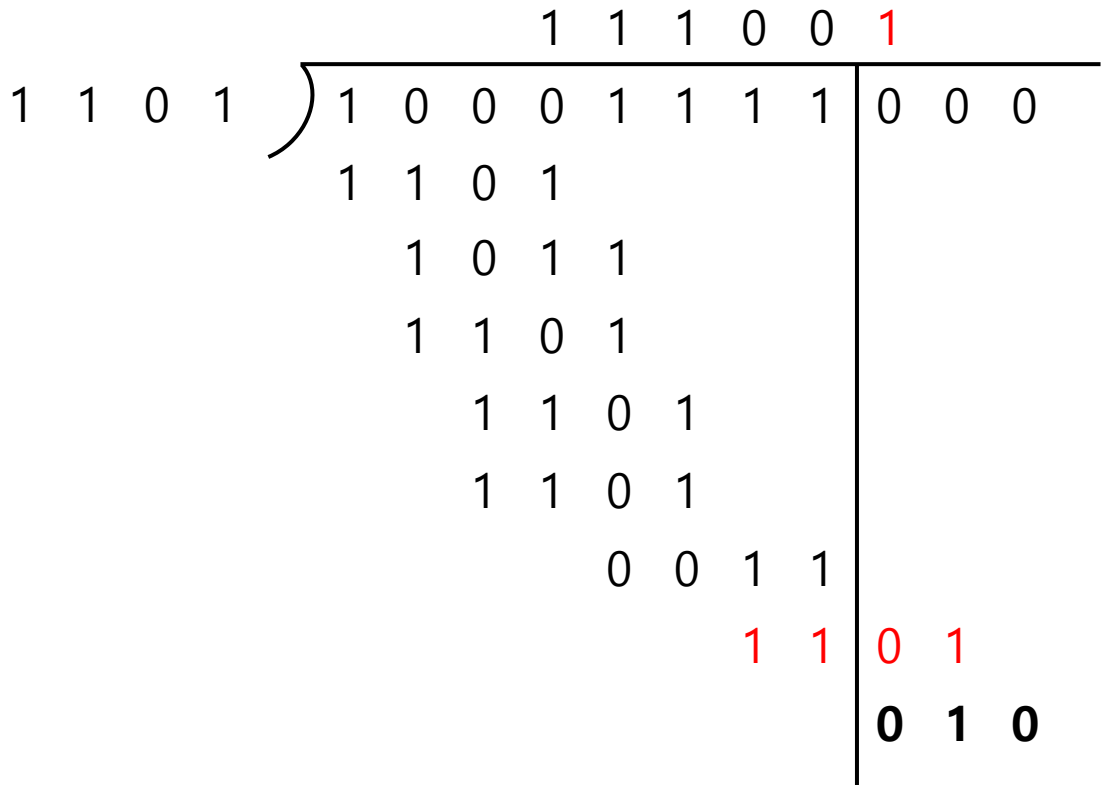
Assignment 1-2. 2 (Sender)



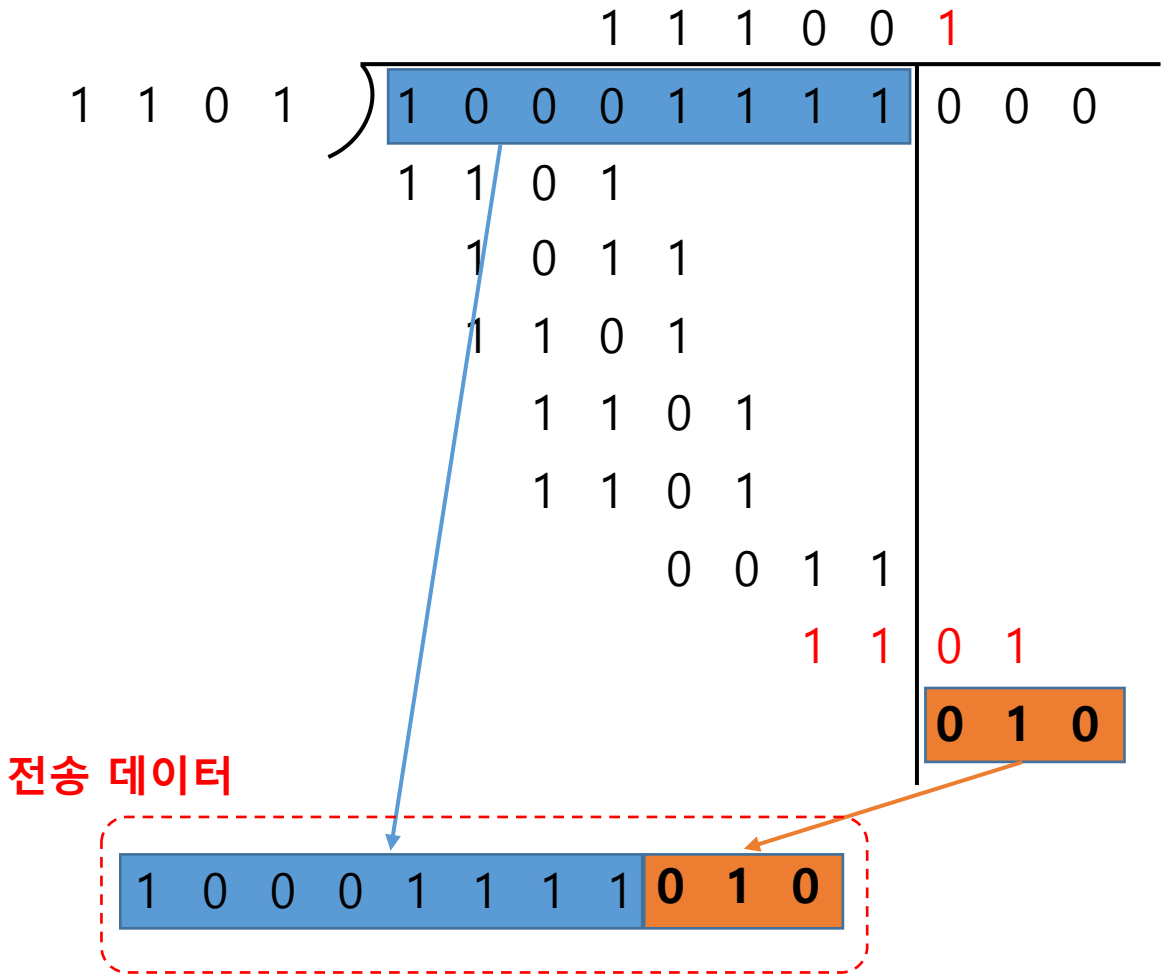
Assignment 1-2. 2 (Sender)



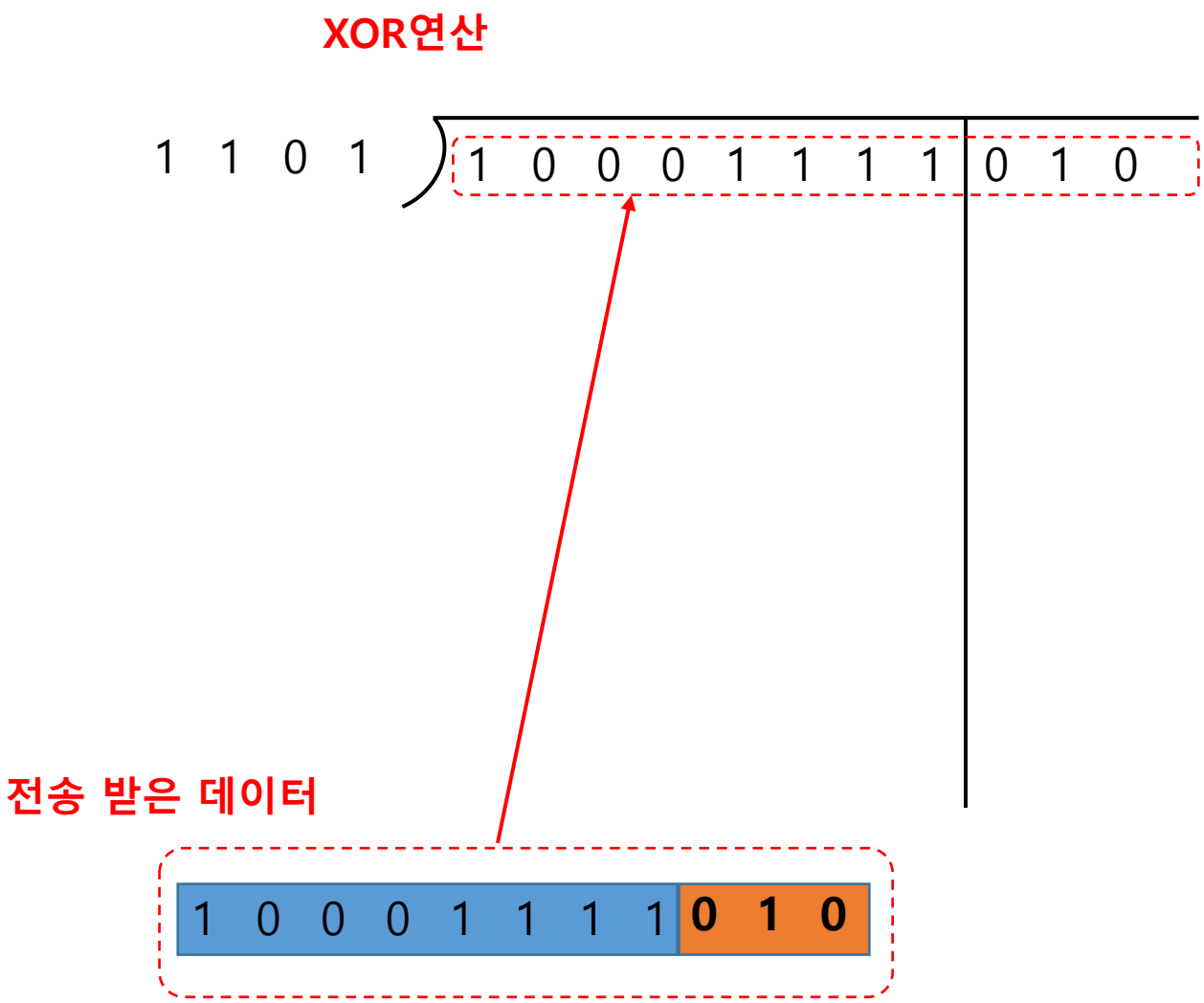
Assignment 1-2. 2 (Sender)



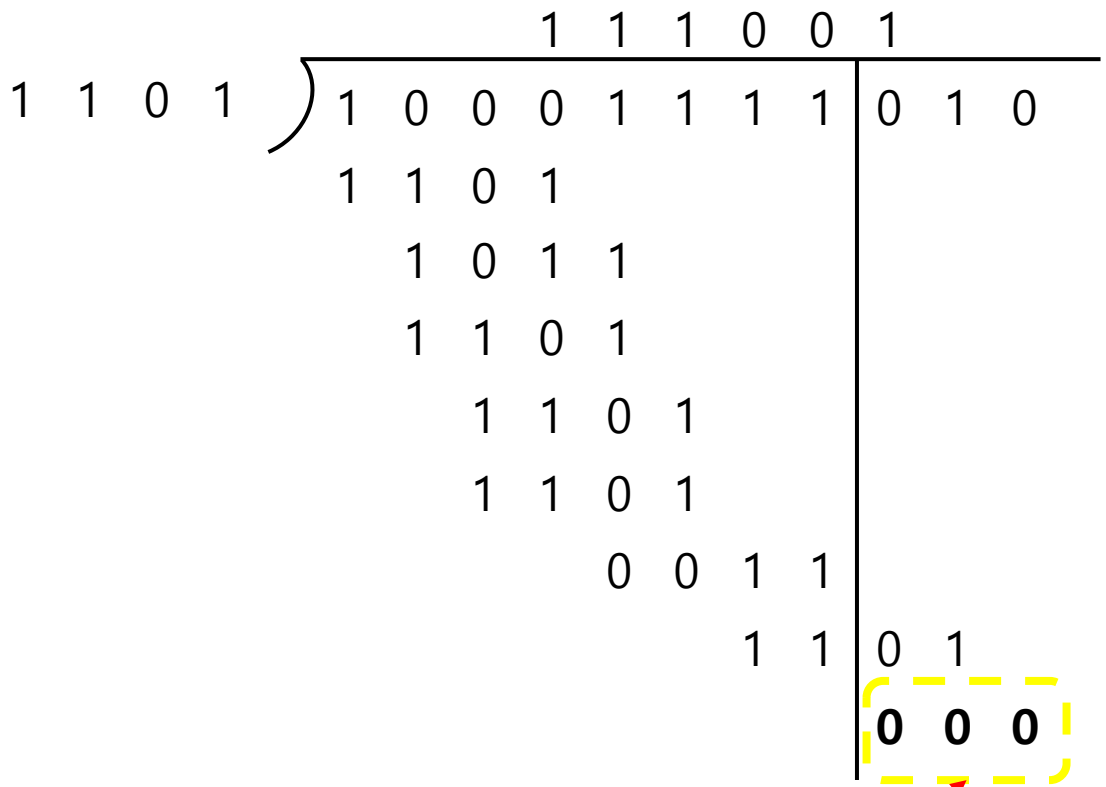
Assignment 1-2. 2 (Sender)



Assignment 1-2. 2 (Receiver)



Assignment 1-2. 2 (Receiver)



나머지가 0인 경우, 에러 발생하지 않음
나머지가 0이 아닌 경우, 에러 발생

ASSIGNMENT 1-2. 3

Assignment 1-2. 3

- 3. Write a program to distinguish filenames that match patterns given a set of filenames along with a wildcard pattern.
  - Wildcard is a way of specifying a filename by using only a part of it, which is widely used in various operating systems.
  - The string used for this purpose is called a wildcard pattern, which is similar to a regular filename but can include special characters such as * or ?.
  - When every character in the pattern matches the corresponding character in the filename, the pattern is said to match the filename.
  - However, the ? in the wildcard pattern matches any single character, and * matches zero or more characters of any type.

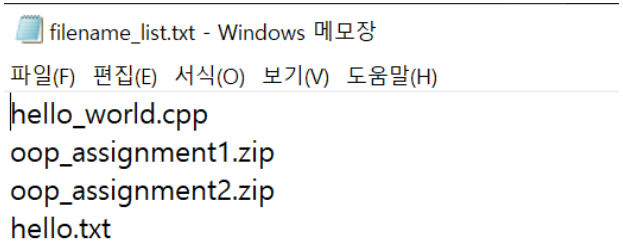
filename_list.txt	Input	Output
hello_world.cpp oop_assignment1.zip oop_assignment2.zip hello.txt oop_assignment3.zip	Input file name : filename_list.txt 2 hello* oop_assignment?.zip	hello_world.cpp hello.txt oop_assignment1.zip oop_assignment2.zip oop_assignment3.zip

Assignment 1-2. 3

- TXT 파일 구성

```
hello_world.cpp
oop_assignment1.zip
oop_assignment2.zip
hello.txt
```

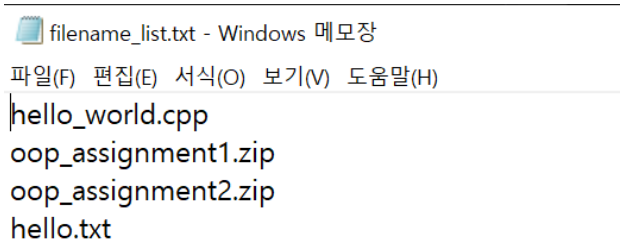
filename_list.txt



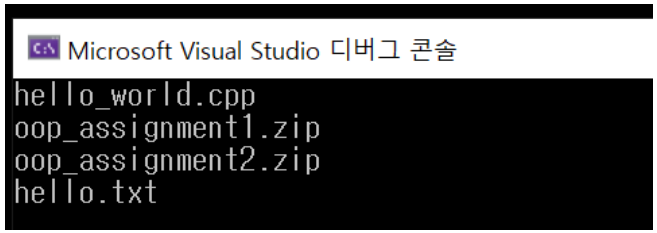
예제 코드

- 텍스트 파일 읽기 예시 코드

```
#include <iostream>
#include <fstream>

using namespace std;

void main() {
    char data[100];
    ifstream fin;
    fin.open("filename_list.txt");

    if (!fin) {
        cout << "no" << std::endl;
    }

    while(!fin.eof()) {
        fin.getline(data, 100);
        cout << data << endl;
    }

    fin.close();
}
```



filename_list.txt



결과화면

예제 코드

- 파일 입력(읽기)
  - 파일 입출력 라이브러리 사용을 위해 헤더파일 포함
    - ifstream: 파일 입력(읽기)
    - ofstream: 파일 출력(쓰기)
    - fstream: 하나의 파일에 읽기와 쓰기 동시에 수행
  - 파일 스트림 객체 생성
    - (읽기의 경우) ifstream 객체명;
  - 파일 열기
    - 파일 스트림 객체.open("파일명");
  - 파일 닫기
    - 파일 스트림 객체.close()

예제 코드

- 파일 입력(읽기)
  - EOF
    - End-of-File
    - 파일의 끝에 도달했을 때, -1 반환
    - ex) ifstream fin("test.txt"); while(!fin.eof())
  - getline(char* s, int n, char delim='₩n')
    - 입력 스트림으로부터 한 줄을 읽어 배열에 저장
    - delim에 지정된 구분 문자까지 또는 n-1개의 문자를 모두 읽어 배열 s에 저장
    - delim에 지정된 구분 문자를 스트림에서 제거
      - ex) char line[50]; cin.getline(line,50);
      - ex) ifstream fin("test.txt"); fin.getline(line,50);

ASSIGNMENT 1-2. 4

Assignment 1-2. 4

- 4. A 2D gray image (NxM) can be stored in a raw image file format.
  - The image consists of NxM points so called 'pixel'. For the **gray** image, each pixel is represented in **1-byte** (0~255 levels).
  - Note that N and M denotes the numbers of row and column of the image.
- Write a program that reads an image file and generates three image files as follows:
  - The program crops the original image with two corners that are entered from a user and stores the cropped image in the file, named '(original image name)_**cropped**_(width)x(height).raw'.
  - Then, the program flips the original image **horizontally** and **vertically**, respectively and stores the **flipped** images in the different files.
    - The names of the flipped image files should be '(original image name)_horizontalflip.raw' and '(original image name)_verticalflip.raw', respectively.
  - Handle all exceptions that you can think.

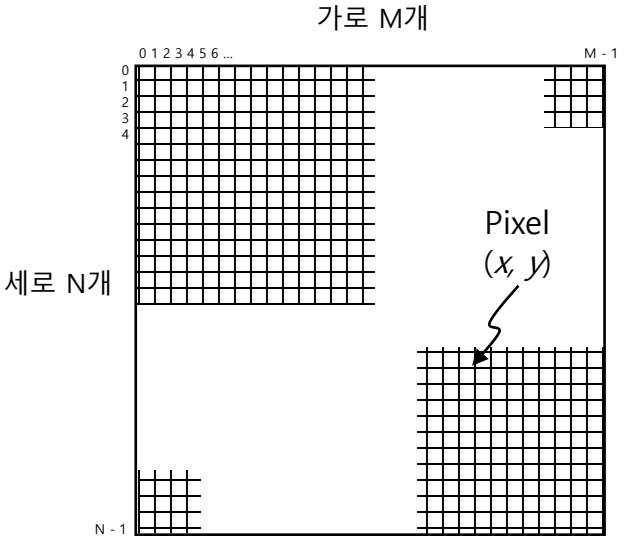
Assignment 1-2. 4

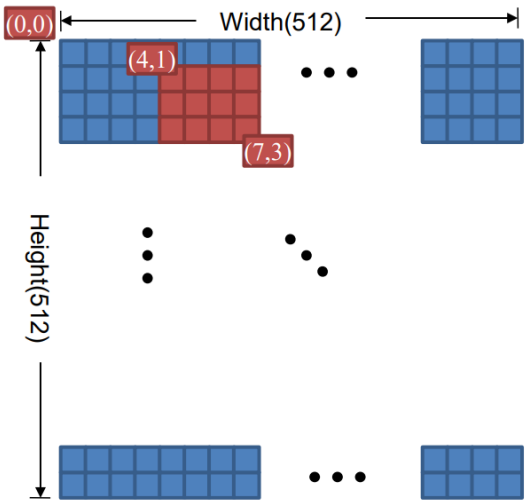
Input	Output
Input file name : barbara_512x512.raw 1st coordinate : 256, 0 2nd coordinate : 512, 192	Output images are stored!

Original image | Cropped image | Horizontally flipped image | Vertically flipped image

Assignment 1-2. 4

- 데이터 포맷
  - 영상의 데이터
    - 가로 M개, 세로 N개의 픽셀로 구성
    - 일반적으로 2차원 배열
    - 각 (x, y)번째 위치한 인덱스의 값은 해당 위치에서의 픽셀 값을 의미
  - RAW 데이터 포맷
    - 별도의 헤더 정보 없이, 영상 데이터(픽셀 값)로만 구성
    - 영상의 크기 정보 필요

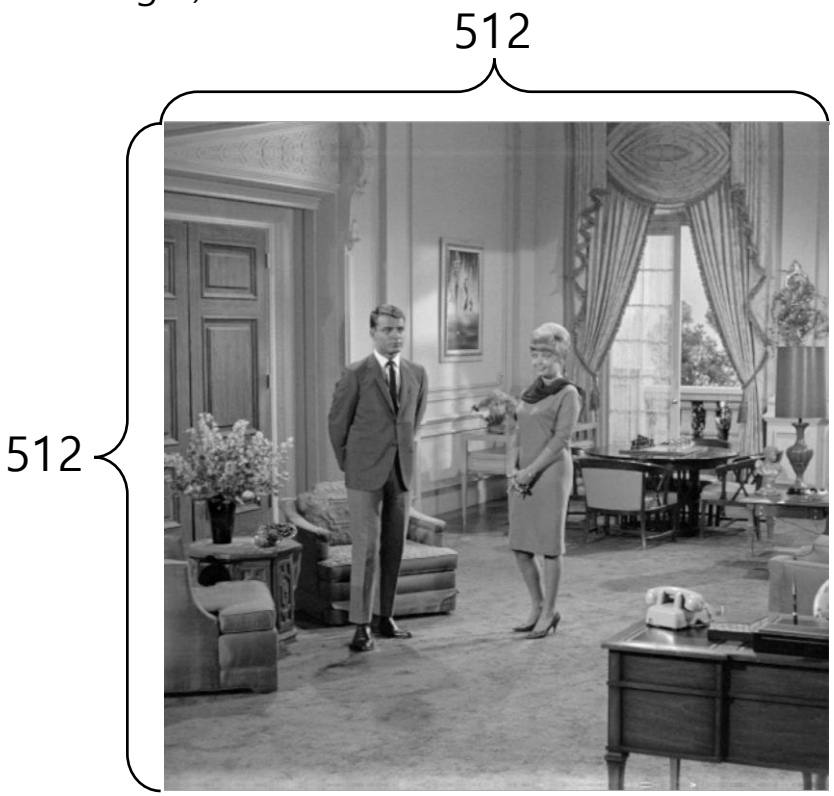




1 pixel 당 1-byte 정보 사용

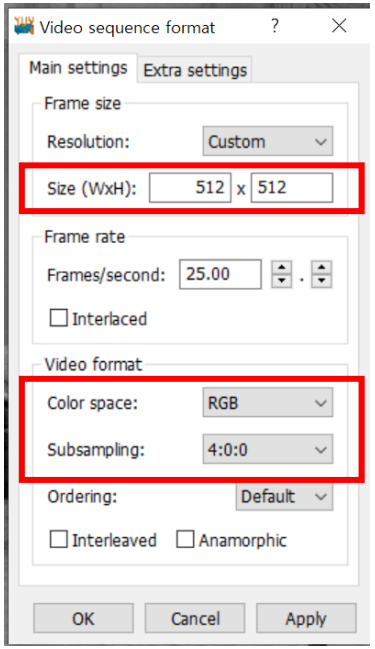
Assignment 1-2. 4

- Test image
  - Couple_512x512_yuv400_8bit.raw
    - Color format: YUV 400
    - Size (width X height): 512 X 512



pYUV

- Raw image or video를 plot하는 툴
  - Index of /~baruffa/dvbt/binaries/player/win32 (unipg.it)
- 사용법
  - 설치, 실행 후 raw image 파일을 drag&drop하면 이미지의 정보를 입력할 수 있는 창이 팝업
  - Test/Reconstruction image의 정보를 아래와 같이 입력 후 OK 버튼을 누르면 이미지가 열림



영상 read & write 예제 코드

- Memory 할당 과정

```
uint8_t** memory_alloc2D(uint32_t height, uint32_t width)
{
    uint8_t** ppMem2d = new uint8_t * [height];
    if (ppMem2d == 0)
        return 0;
    ppMem2d[0] = new uint8_t[height * width];
    if (ppMem2d[0] == 0)
    {
        delete[] ppMem2d;
        return 0;
    }

    for (int i = 1; i < height; i++)
        ppMem2d[i] = ppMem2d[i - 1] + width;
    return ppMem2d;
}
```

영상 read & write 예제 코드

- 할당된 memory 해제 과정 **메모리 해제 안할 경우, 감점**

```
int memory_free2D(uint8_t** ppMemAllocated)
{
    if (ppMemAllocated == 0)
        return -1;

    if (ppMemAllocated[0])
        delete[] ppMemAllocated[0];

    delete[] ppMemAllocated;
    return 0;
}
```

영상 read & write 예제 코드

- 영상 read & write 과정

```
#define _CRT_SECURE_NO_WARNINGS
#include <iostream>

using namespace std;

#define IMG_HEIGHT 512
#define IMG_WIDTH  512
```

영상 read & write 예제 코드

- 영상 read & write 과정

```
int main(void)
{
    FILE* fpInputImage = 0;
    FILE* fpOutputImage = 0;
    uint8_t** ppInputImageBuffer = 0;
    const char* IMG_NAME = "Couple_512x512_yuv400_8bit.raw";
    // input file open
    fpInputImage = fopen(IMG_NAME, "rb");
    // memory allocaiton
    ppInputImageBuffer = memory_alloc2D(IMG_HEIGHT, IMG_WIDTH);
    // input file read to memory from the file

    for (int i = 0; i < IMG_HEIGHT; i++)
        fread((uint8_t*)ppInputImageBuffer[i], sizeof(uint8_t), IMG_WIDTH, fpInputImage);
    // output fileopen
    fpOutputImage = fopen("result_512x512_yuv400_8bit.raw", "wb");
    // write the file
    for (int i = 0; i < IMG_HEIGHT; i++)
        fwrite((uint8_t*)ppInputImageBuffer[i], sizeof(uint8_t), IMG_WIDTH, fpOutputImage);
    memory_free2D(ppInputImageBuffer);
    fclose(fpInputImage);
    fclose(fpOutputImage);
    return 0;
}
```

과제 제출 방법

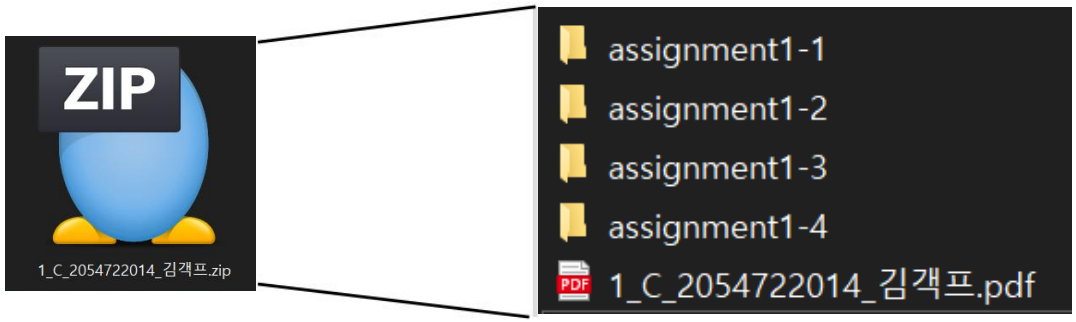
과제 제출 방법

- **FTP Upload (Klas 과제 제출 X)**
  - Address : ftp://223.194.8.1:1321
  - username : IPSL_OBJ
  - password : ipslobj_2023

- **Due date**
  - Soft copy: 마감일 3/24(금) 23:59:59까지 제출 (서버시간 기준)
  - Delay
    - 마감일 이후 +7일까지 제출 가능
    - 단, 1일 초과마다 과제 총점의 10%씩 감점

과제 제출 방법

- Soft copy
  - 과제(보고서, 소스 코드)를 압축한 파일 제출
    - 설계반_실습반_학번_이름.zip
      - 예) 설계1반 수강, 실습 A반: 1_A_학번_이름.zip
      - 예) 설계 수강, 실습 미수강: 2_0_학번_이름.zip
      - 예) 설계 미 수강, 실습 C반: 0_C_학번_이름.zip



  - 과제 수정하여 업로드 시 버전 명시
    - 설계반_실습반_학번_이름_verX.zip

과제 제출 방법

- Soft copy
  - 과제 보고서
    - 영문 또는 한글로 작성
    - **반드시 PDF**로 제출 (PDF 외 파일 형식으로 제출시 0점 처리)
    - 보고서 양식
      - 문제 및 설명(문제 capture 금지) / 결과 화면 / 고찰
      - 보고서 양식은 아래 경로에서 참고
        - https://www.ipsl.kw.ac.kr/post/1%EC%B0%A8-%EA%B3%BC%EC%A0%9C
    - **소스코드 제외**
    - 분량 제한 없음
    - **표절 적발 시 0점 처리**
  - 소스 코드
    - Visual Studio 2022 community 사용 필수
      - https://docs.microsoft.com/ko-kr/visualstudio/install/install-visual-studio?view=vs-2022
    - STL (Standard Template Library) 사용 금지 (vector, map, algorithm 등)
    - Debug 폴더를 제외한 모든 파일 제출
      - .sln 파일 포함(.cpp 만 제출하지 말것)
    - **각 문제마다 프로젝트 파일 생성 필수**
    - **주석 반드시 달기**
    - **소스코드 표절 적발 시 0점 처리**

END OF PRESENTATION

Q&A