

Problem 1

The file A.dat store each value with its row and column indices by the following format.

```
1 row_idx,column_idx:value
```

The data is stored in unordered sequence and line by line. You need to read all the element into a matrix for the input file for the following problems. An appropriate regular expression is recommended to easily read matrix elements.

```
1 55,0:6 18,4:2 67,2:-9 36,3:1 54,1:1
2 51,2:-3 58,3:1 26,3:9 62,4:5 22,4:-6
3 5,0:6 96,3:-5 21,2:-6 8,4:1 61,3:-8
4 64,3:5 34,1:-6 29,1:-10 44,0:-1 63,1:-3
5 50,3:10 23,1:9 42,2:-5 79,1:-10 99,2:8
6 0,1:7 45,1:-8 50,0:5 32,4:8 73,3:10
7 5,3:0 12,1:-10 94,1:6 59,3:4 13,4:-2
8 77,3:5 15,3:2 33,2:-9 95,4:-4 5,2:9
9 63,0:-4 7,4:4 49,3:-3 34,0:10 65,0:-1
10 :
```

1. (10pt.) Write and test your Python function to find the maximum values of row and column(file name: find_size.py).

코드

```
f= open("a.dat", "r")
```

```
r=[]
```

```
c=[]
```

```
v=[]
```

```
while True:
```

```
    x = f.readline()
```

```
    if not x: break
```

```
    x = x.split()
```

```
    for y in x:
```

```
        r_index=y.split(',')
```

```
        r.append(r_index[0])
```

```
        r_index.remove(r_index[0])
```

```
        c_index=r_index[0].split(':')
```

```
        c.append(int(c_index[0]))
```

```
        v.append(int(c_index[1]))
```

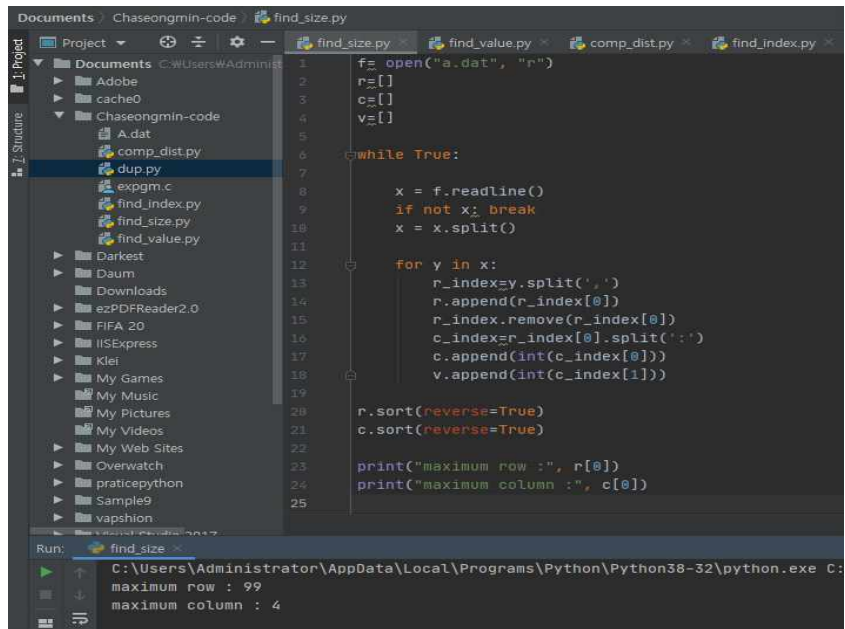
```
r.sort(reverse=True)
```

```
c.sort(reverse=True)
```

```
print("maximum row :", r[0])
```

```
print("maximum column :", c[0])
```

결과



```
1 f = open("a.dat", "r")
2 r = []
3 c = []
4 v = []
5
6 while True:
7
8     x = f.readline()
9     if not x: break
10    x = x.split()
11
12    for y in x:
13        r_index = y.split('.')
14        r.append(r_index[0])
15        r_index.remove(r_index[0])
16        c_index = r_index[0].split(':')
17        c.append(int(c_index[0]))
18        v.append(int(c_index[1]))
19
20 r.sort(reverse=True)
21 c.sort(reverse=True)
22
23 print("maximum row :", r[0])
24 print("maximum column :", c[0])
25
```

Run: find_size

```
C:\Users\Administrator\AppData\Local\Programs\Python\Python38-32\python.exe C:
maximum row : 99
maximum column : 4
```

2. (10pt.) Write and test your Python function to find the maximum and minimum values(filename: find_value.py).

1-1의 소스코드를 살짝만 수정해주니 바로 구할 수 있었다.

코드

```
f = open("a.dat", "r")
```

```
r = []
```

```
c = []
```

```
v = []
```

```
while True:
```

```
    x = f.readline()
```

```
    if not x: break
```

```
    x = x.split()
```

```
    for y in x:
```

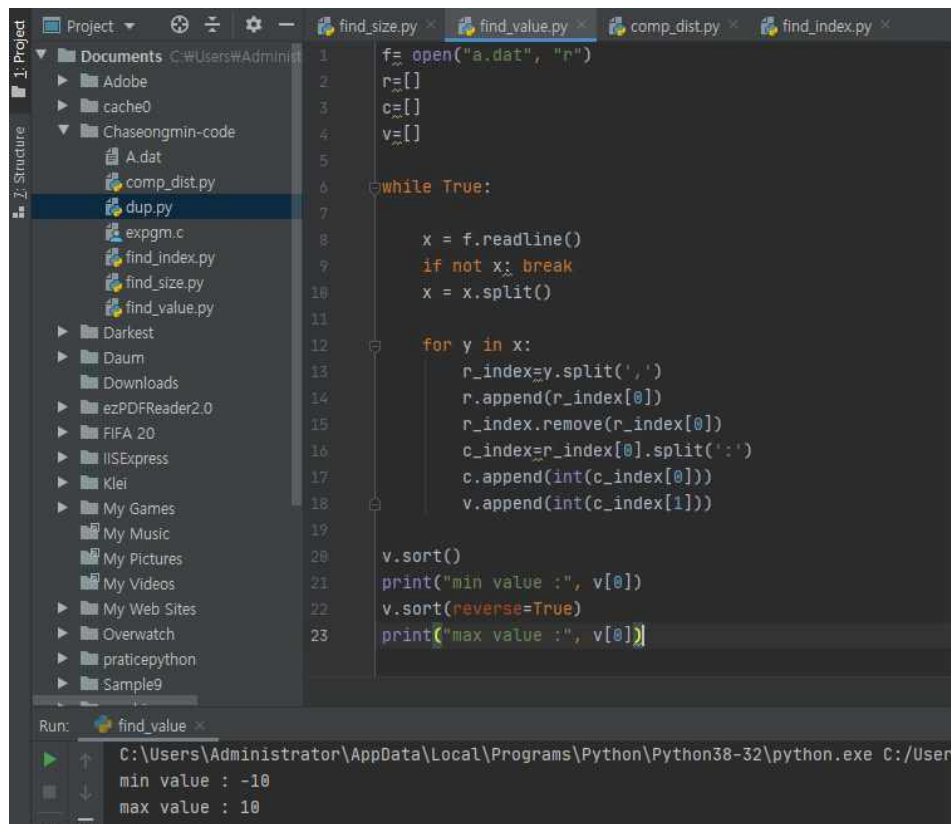
```

r_index=y.split(',')
r.append(r_index[0])
r_index.remove(r_index[0])
c_index=r_index[0].split(':')
c.append(int(c_index[0]))
v.append(int(c_index[1]))

v.sort()
print("min value :", v[0])
v.sort(reverse=True)
print("max value :", v[0])

```

결과



```

1 f = open("a.dat", "r")
2 r = []
3 c = []
4 v = []
5
6 while True:
7
8     x = f.readline()
9     if not x: break
10    x = x.split()
11
12    for y in x:
13        r_index=y.split(',')
14        r.append(r_index[0])
15        r_index.remove(r_index[0])
16        c_index=r_index[0].split(':')
17        c.append(int(c_index[0]))
18        v.append(int(c_index[1]))
19
20 v.sort()
21 print("min value :", v[0])
22 v.sort(reverse=True)
23 print("max value :", v[0])

```

Run: find_value

```

C:\Users\Administrator\AppData\Local\Programs\Python\Python38-32\python.exe C:/User
min value : -10
max value : 10

```

3. (10pt.) Write and test your Python function to calculate the Euclidean distance between 10-th and 27-th rows(file name: comp_dist.py).

유클리드 좌표계에서 두 점 사이의 직선거리는 다음과 같다.

$$d(\mathbf{p}, \mathbf{q}) = d(\mathbf{q}, \mathbf{p}) = \sqrt{(q_1 - p_1)^2 + (q_2 - p_2)^2 + \dots + (q_n - p_n)^2}$$

$$= \sqrt{\sum_{i=1}^n (q_i - p_i)^2}.$$

이를 토대로 알고리즘을 먼저 작성해 보자면 다음과 같다.

- 10번째 행들(P로 명명)과 27번째 행들(Q로 명명)의 값들로 구할 수 있는 유클리디안 거리는 아래 식으로 구할 수 있다.

$$\sqrt{(P_1 - Q_1)^2 + (P_2 - Q_2)^2 + \dots + (P_i - Q_i)^2} = \sqrt{\sum_{i=1}^5 (P_i - Q_i)^2}$$

즉 10번째 행의 value 5개와

27번째 행의 value 5개를 구하고, 이들을 위의 식에 넣어주면 해결이 된다.

수도 코드

- 1) f = open("a.dat", "r")등 .dat 파일로부터 데이터를 행렬로 가져오게 세팅한다.
- 2) 10, 27행들의 value 5개씩 구한다.
- 3) 위의 수식을 Numpy 라이브러리 등을 이용하여 적는다.
- 4) 2에서 구한 값들을 3의 수식에 넣어주어 구하고자하는 유클리디안 거리를 구한다.

4. (20pt.) Write and test a Python function that finds row indices whose Euclidean distance is less than or equal to 10.0 to the 37th row(file name: find_index.py).

유클리디안 거리를 구하는 공식은 위와 동일하게 아래 식으로 구할 수 있다.

$$\sqrt{(P_1 - Q_1)^2 + (P_2 - Q_2)^2 + \dots + (P_i - Q_i)^2} = \sqrt{\sum_{i=1}^5 (P_i - Q_i)^2}$$

수도 코드

- 1) f = open("a.dat", "r")등 .dat 파일로부터 데이터를 행렬로 가져오게 세팅한다.
- 2) 37번째 행들의 value 5개를 구한다.
- 3) 위의 수식을 Numpy 라이브러리 등을 이용하여 적는다.
- 4) 수식에 미리 37번째 행들의 value 5개를 넣어두고, 반복문을 이용하여 각 행마다 value를 구하고 이를 37번째 행들의 value가 들어 있는 수식의 비어있는 값 부분에 넣는다.
- 5) 결과로 나온 유클리디안 거리가 10.0이 넘으면 넘어가고, 10.0 이하라면 그 행의 인덱스를 Print로 출력이 되도록 하면 원하고자하는 행의 인덱스를 구할 수 있다.

Problem 2

Modularization is a key for software design and implementation. This will evaluate your understandability on the modularization concept. The code expgm.c has some functions written in

C.

1. (5pt.) Compile and execute the program on a command line.

```
seongmin@ubuntu:~/Desktop$ gcc -o expgm expgm.c
seongmin@ubuntu:~/Desktop$ ./expgm
Way 1 : 83 86 77 15 93 35 86 ==> 15 35 77 83 86 86 93
Way 2 : 92 49 21 62 27 90 59 ==> 21 27 49 59 62 90 92
Way 3 : 63 26 40 26 72 36 11 ==> 11 26 26 36 40 63 72
Way 4 : 68 67 29 82 30 62 23 ==> 23 29 30 62 67 68 82
```

2. (5pt.) Describe what this program does.

Sort1(삽입), sort2(버블), sort3(선택), sort4(퀵) 정렬로 각각의 데이터를 정렬하는 프로그램이다.

3. (10pt.) Split expgm.c into more than 6 files. Each file should include related functions or macros. Write and test your Makefile script to build the executable file.

sort1.c sort2.c sort3.c sort4.c function.c print.c, header.h, program.c로 분할하였다. 각 순서대로 코드는 아래와 같다.

```
#include "header.h"
void sort1(int *a, int n){
    int i, j, move;
    int val;

    for(i=1; i<n; i++){
        val = a[i];
        j = i;
        if( a[j-1] > val )
            move = true;
        else
            move = false;

        while( move ) {
            a[j] = a[j-1];
            j = j - 1;
            if( j > 0 && a[j-1] > val )
                move = true;
            else
                move = false;
        }
        a[j] = val;
    }
}
```

Sort1.c

```
#include "header.h"

void sort2(int *a, int n){
    int i, j;
    int val;
    for(i=n-1; i > 0; i--){
        for(j=0; j<i; j++){
            if( a[j] > a[j+1] ) {
                val = a[j];
                a[j] = a[j+1];
                a[j+1] = val;
            }
        }
    }
}
```

Sort2.c

```
#include "header.h"

void sort3(int *a, int n){
    int i, j, least;
    int val;

    for(i = 0; i < n-1; i++){
        least = i;
        for( j = i + 1; j < n; j++)
            if( a[j] < a[least] ) least = j;
        val = a[least];
        a[least] = a[i];
        a[i] = val;
    }
}
```

Sort3.c

```
#include "header.h"

void sort4(int arr[], int low, int high)
{
    if (low < high)
    {
        int pi = part(arr, low, high);

        sort4(arr, low, pi - 1);
        sort4(arr, pi + 1, high);
    }
}
```

Sort4.c

```
#include "header.h"

void get_value(int *a, int n) {
    for(int i=0; i < n; i++)
        a[i] = rand() % 100;
}

void swap(int* a, int* b)
{
    int t = *a;
    *a = *b;
    *b = t;
}

int part(int arr[], int low, int high)
{
    int pivot = arr[high];
    int i = (low - 1);

    for (int j = low; j <= high- 1; j++)
    {
        if (arr[j] <= pivot)
        {
            i++;
            swap(&arr[i], &arr[j]);
        }
    }
    swap(&arr[i + 1], &arr[high]);
    return (i + 1);
}
```

function.c

```
#include "header.h"

void print_array( int *a, int n) {
    for( int i=0; i<n; i++)
        printf("%d ", a[i]);
}
```

print.c

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

#define true 1
#define false 0

void print_array( int *a, int n);
void get_value(int *a, int n);
void sort1(int *a, int n);
void sort2(int *a, int n);
void sort3(int *a, int n);
void swap(int* a, int* b);
int part(int arr[], int low, int high);
void sort4(int arr[], int low, int high);
```

header.h

```

#include "header.h"

int main(){
    int a[7];
    int n = 7;

    get_value(a, n);
    printf("Way 1 : ");
    print_array(a, n);
    sort1(a, n);
    printf(" ==> ");
    print_array(a, n);
    printf("\n");

    get_value(a, n);
    printf("Way 2 :");
    print_array(a, n);
    sort2(a, n);
    printf(" ==> ");
    print_array(a, n);
    printf("\n");

    get_value(a, n);
    printf("Way 3 :");
    print_array(a, n);
    sort3(a, n);
    printf(" ==> ");
    print_array(a, n);
    printf("\n");

    get_value(a, n);
    printf("Way 4 :");
    print_array(a, n);
    sort4(a, 0, 6);
    printf(" ==> ");
    print_array(a, n);
    printf("\n");

    return 1;
}

```

program.c

아래는 makefile과 실행결과이다.

```

program : sort1.o sort2.o sort3.o sort4.o function.o print.o program.o
gcc -o program sort1.o sort2.o sort3.o sort4.o function.o print.o program.o
sort1.o : sort1.c header.h
gcc -c sort1.c
sort2.o : sort2.c header.h
gcc -c sort2.c

sort3.o : sort3.c header.h
gcc -c sort3.c

sort4.o : sort4.c header.h
gcc -c sort4.c

function.o : function.c header.h
gcc -c function.c

print.o : print.c header.h
gcc -c print.c

program.o : program.c header.h
gcc -c program.c

```


4. (10pt.) Based on your split sources, write and test your CMake script to build the executablefile.

Cmake가 우분투에서 에러가 계속 발생하여 설치가 되지 않았고(아래 에러 첨부), 그래서 차선책으로 Make를 실행하였다.

```
root@ubuntu:/home/seongmin/Desktop# sudo apt-get install cmake
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  cmake-data libjsoncpp1 libhash0
Suggested packages:
  cmake-doc ninja-build
The following NEW packages will be installed:
  cmake cmake-data libjsoncpp1 libhash0
0 upgraded, 4 newly installed, 0 to remove and 9 not upgraded.
Need to get 4,929 kB/5,115 kB of archives.
After this operation, 25.8 MB of additional disk space will be used.
Do you want to continue? [Y/n] y
Ign:1 http://us.archive.ubuntu.com/ubuntu disco/main amd64 cmake-data all 3.13.4-1
Err:2 http://us.archive.ubuntu.com/ubuntu disco/main amd64 cmake amd64 3.13.4-1
404 Not Found [IP: 91.189.91.39 80]
Err:1 http://us.archive.ubuntu.com/ubuntu disco/main amd64 cmake-data all 3.13.4-1
404 Not Found [IP: 91.189.91.39 80]
E: Failed to fetch http://us.archive.ubuntu.com/ubuntu/pool/main/c/cmake/cmake-data_3.13.4-1_all.deb 404 Not Found [IP: 91.189.91.39 80]
E: Failed to fetch http://us.archive.ubuntu.com/ubuntu/pool/main/c/cmake/cmake_3.13.4-1_amd64.deb 404 Not Found [IP: 91.189.91.39 80]
E: Unable to fetch some archives, maybe run apt-get update or try with --fix-missing?
root@ubuntu:/home/seongmin/Desktop# cmake

Command 'cmake' not found, but can be installed with:

snap install cmake # version 3.17.3, or
apt install cmake # version 3.13.4-1

See 'snap info cmake' for additional versions.
```

Cmake 설치 시 발생한 에러

아래가 Make를 실행한 결과이다.

```
seongmin@ubuntu:~/Desktop$ vi Makefile
seongmin@ubuntu:~/Desktop$ make
gcc -c sort1.c
gcc -o program sort1.o sort2.o sort3.o sort4.o function.o print.o program.o
seongmin@ubuntu:~/Desktop$ ls
expgm.c      function.o  Makefile   print.o    program.c  sort1.c    sort2.c    sort3.c    sort4.c
function.c   header.h   print.c    program    program.o  sort1.o    sort2.o    sort3.o    sort4.o
seongmin@ubuntu:~/Desktop$ ./program
Way 1 : 83 86 77 15 93 35 86 ==> 15 35 77 83 86 86 93
Way 2 : 92 49 21 62 27 90 59 ==> 21 27 49 59 62 90 92
Way 3 : 63 26 40 26 72 36 11 ==> 11 26 26 36 40 63 72
Way 4 : 68 67 29 82 30 62 23 ==> 23 29 30 62 67 68 82
seongmin@ubuntu:~/Desktop$
```

Problem 3

(10pt.) Write a shell script to extract file name and size in the current directory. The output should display the file name first, followed by the file size like this example(file name: script.sh).

```
1 dsscript.sh 64
2 expgm 13032
3 expgm.c 2459
4 total 111811
```

코드

```
#!/bin/bash
for i in $(ls)
do
    echo $i $(stat -c%s $i)
done
```

결과

```
root@ubuntu:/home/seongmin/Desktop# ./script.sh
expgm 16872
expgm.c 2455
function.c 488
function.o 1968
header.h 328
Makefile 465
print.c 116
print.o 1592
program 17072
program.c 674
program.o 3160
script.sh 61
sort1.c 436
sort1.o 1456
sort2.c 288
sort2.o 1432
sort3.c 291
sort3.o 1432
sort4.c 206
sort4.o 1528
root@ubuntu:/home/seongmin/Desktop#
```

Problem 4

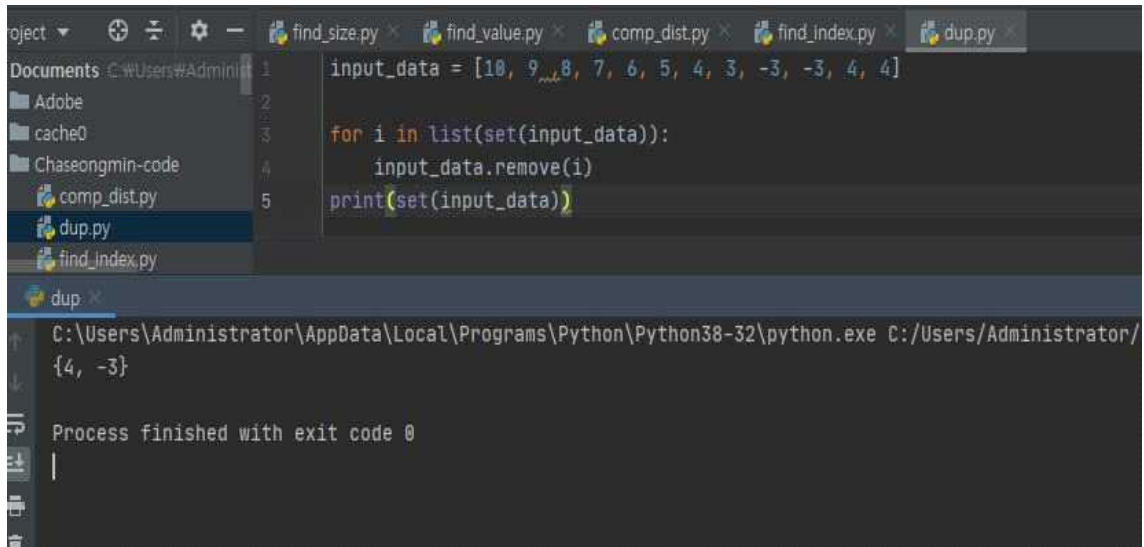
(10pt.) Design and implement a Python script that reads a set of integers and returns the integers that occur two or more times(file name: dup.py). The input is 10, 9, 8, 7, 6, 5, 4, 3, -3, -3, 4, 4.

코드

```
input_data = [10, 9, 8, 7, 6, 5, 4, 3, -3, -3, 4, 4]

for i in list(set(input_data)):
    input_data.remove(i)
print(set(input_data))
```

결과



```
1 input_data = [10, 9, 8, 7, 6, 5, 4, 3, -3, -3, 4, 4]
2
3 for i in list(set(input_data)):
4     input_data.remove(i)
5 print(set(input_data))
```

C:\Users\Administrator\AppData\Local\Programs\Python\Python38-32\python.exe C:/Users/Administrator/
{4, -3}

Process finished with exit code 0

Problem 5

(10pt.) Please list all the contents you have learned through this class. In addition, point out what the professor should improve to make this class better.

1. 오픈소스 SW활용을 통해 제가 배운 콘텐츠들 리스트

1) 파이썬이 무엇이고 왜 써야하는가에 대해 배웠습니다. (Intro, Why?)

2) 파이썬을 왜 써야하고 배워야 하는가에 대해 이해를 하였으니 본격적으로 파이썬에 대한 공부를 진행하며 파이썬에 대한 이해도를 높였습니다. (파이썬 구조의 기본, 환경 설정(IDLE 등), 데이터 타입과 문자열, 모듈과 객체, 리스트, 조건 문, Loop 문, math 등 import하여 사용할 수 있는 라이브러리들과 활용법, 파일 프로세싱, 슬라이싱, 빠질 수 없는 셋과 딕셔너리, 알고리즘과 탐색 알고리즘, 소팅 방법들과, 객체 지향 프로그래밍, 클래스..)

3) 단순히 파이썬 내용만 공부한 것이 아니라 중간 중간 c++ 코드로 코드 가이드 해주셨던 것도 기억에 남고, 셸 스크립트를 사용하기 위해 가상머신으로 리눅스를 올리고 그 위에서 셸 스크립트 연습도 하고, 실습도 했었고, 마찬가지로 GNU make와 Cmake도 해보는 시간을 가지며 어찌 보면 생소하면서도 알아두어야 할 것들에 대해서도 이번 강의를 통해 많이 얻어가게 되었습니다.

2. 오픈소스 SW활용 수업의 개선방향과 후기

개선 방향)

1. 실습 문제가 있어서 좋았지만, 실습 문제가 조금 더 있었으면 보다 이해가 더 잘되었을 것 같으면서도, 그렇게 되면 강의 시간이 늦어지기 때문에 이 부분은 학생이 공부할만한 학습 자료에 대한 추천이 있으면 더 낫지 않을까 싶습니다.

2. 과제를 하면서 제 실력이 많이 늘어난게 느껴졌었는데, 과제 제출 주기가 너무 길다보니 저도 모르게 꾸준히 하지 않고 있는 모습을 볼 수 있었고, 그렇기에 다음번에는 보다 양은 좀 적더라도 자주 과제가 있었으면 좋겠다는 생각을 가졌습니다.

후기)

코로나19로 인하여 대면강의를 듣지 못해 아쉬웠지만, 교수님께서 좋은 화질과 깔끔한 목소리로 녹화를 해주셔서 매우 감사했습니다.

1에서 제가 3가지 항목으로 나눈 내용을 정리하자면, 제가 이번학기에 오픈소스 SW활용 수업을 통하여 얻은 단순한 언어 공부에서 끝나는 수준이 아니라, 소프트웨어학도로서 알아야 할 가상머신을 통한 리눅스 활용법과 간단한 셸스크립트 작성법, 어찌보면 제일 생소했던 개념인 make Cmake 등 다양한 개념적이고 실습을 할 수 있는 부분과 수업 중간 중간 보여주는 수학 문제들 풀이 방식은 앞으로 많이 활용할 수 있을 것 같아 정말 교수님께 감사하다는 말씀 드리고 싶습니다.

이번학기 코로나로 인해 직접 찾아뵙지 못해서 아쉬웠는데 다음 학기에 뵙게 된다면 먼저 인사 드리겠습니다 이번학기 정말로 수고 많으셨습니다.