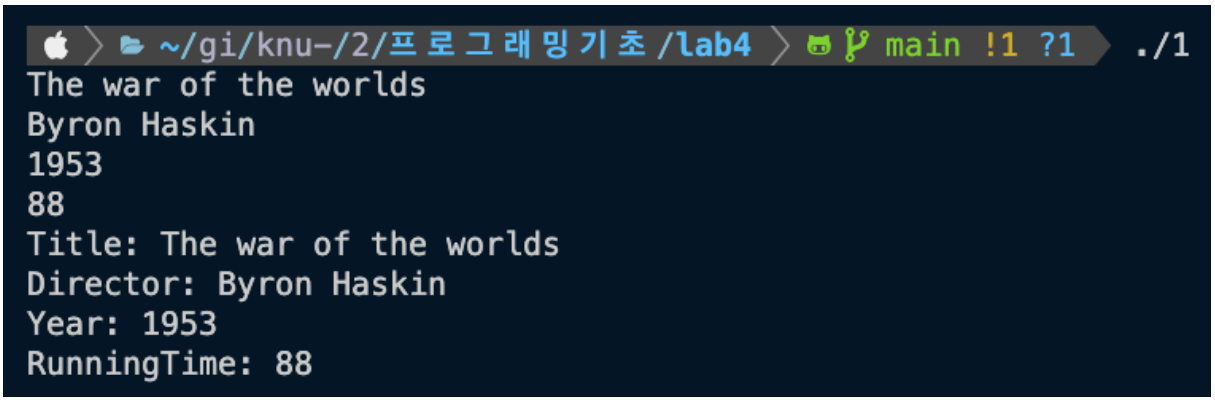
lab4 과제

상태	In progress

1번

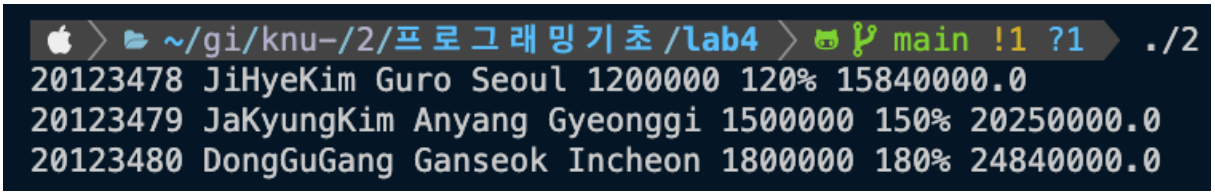
```
#include <stdio.h>

typedef struct {
    char title[50];
    char director[50];
    int year;
    int runningTime;
} MovieData;

void inputMovie(MovieData *m) {
    fgets(m->title, 50, stdin);
    fgets(m->director, 50, stdin);
    scanf("%d", &m->year);
    scanf("%d", &m->runningTime);
}

void printMovie(MovieData *m) { printf("Title: %sDirector: %sYear: %d \nRun
ningTime: %d\n", m->title, m->director, m->year, m->runningTime); }

int main() {
    MovieData movie;

    inputMovie(&movie);
    printMovie(&movie);
}
```



2번

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

#define MAX 10

typedef struct {
    char id[10];
    char name[30];
    char address[50];
    int monthWage;
    double incentive;
} Employee;

void removeNewLine(char *str) {
    int len = strlen(str);
    if (len > 0 && str[len - 1] == '\n') {
        str[len - 1] = '\0';
    }
}

void printEmployee(Employee e[], int finalIdx) {
    for (int i = 0; i < finalIdx; i++) {
        printf("%s %s %s %d %d%% %.1lf", e[i].id, e[i].name, e[i].address,
e[i].monthWage, (int)((e[i].incentive) * 100),
               e[i].monthWage * (12 + e[i].incentive));
        printf("\n");
    }
}

int main() {
    FILE *fp;
    fp = fopen("f2.txt", "r");
    Employee e[MAX];
    int finalIdx = 0;

    for (int i = 0; i < MAX; i++) {
        if (fgets(e[i].id, sizeof(e[i].id), fp) == NULL) {
            finalIdx = i;
            break;
        }
        fgets(e[i].name, sizeof(e[i].name), fp);
        fgets(e[i].address, sizeof(e[i].address), fp);
        fscanf(fp, "%d", &e[i].monthWage);
```

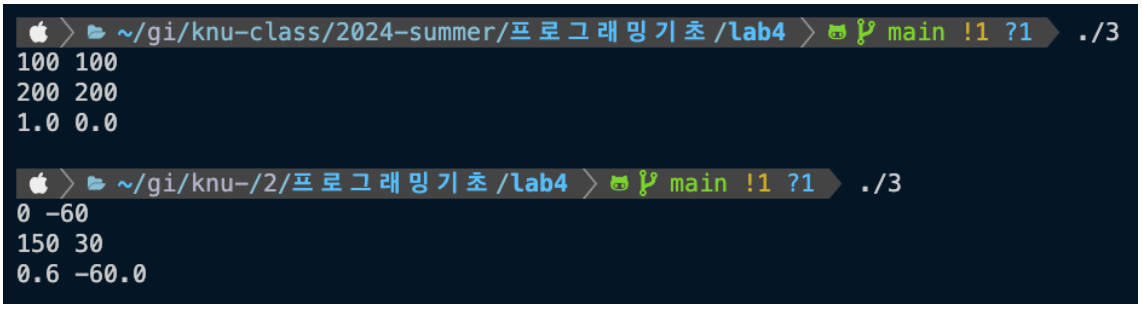
```
        fscanf(fp, "%lf", &e[i].incentive);
        fscanf(fp, "%*c"); // 마지막에 남아있는 문자 제거

        removeNewLine(e[i].id);
        removeNewLine(e[i].name);
        removeNewLine(e[i].address);
    }

    fclose(fp);

    printEmployee(e, finalIdx);
}
```



3번

```
#include <stdio.h>

typedef struct {
    double x;
    double y;
} point;

typedef struct {
    double slope;
    double yintersect;
} line;

void setLineSlope(line *l, point *p1, point *p2) { l->slope = (p2->y - p1->
y) / (p2->x - p1->x); }

void setLineIntersect(line *l, point *p1, point *p2) { l->yintersect = p1->
y - l->slope * p1->x; }

int main() {
    point p1, p2;
    line l1;
```

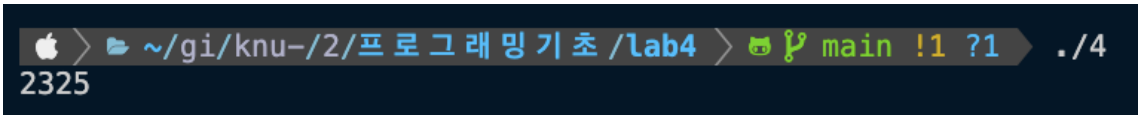
```
    scanf("%lf %lf", &p1.x, &p1.y);
    scanf("%lf %lf", &p2.x, &p2.y);

    setLineSlope(&l1, &p1, &p2);
    setLineIntersect(&l1, &p1, &p2);

    printf("%.1lf %.1lf\n", l1.slope, l1.yintersect);
}
```



4번

```
#include <stdio.h>

typedef struct {
    char fname[100];
    int calories;
} Food;

int totalCalories(Food ary[], int size) {
    int tot = 0;
    for (int i = 0; i < size; i++) {
        tot += ary[i].calories;
    }
    return tot;
}

int main() {
    FILE *fp;
    fp = fopen("f4.txt", "r");
    int idx = 0;
    Food f[10];

    while (fscanf(fp, "%s %d", &f[idx].fname, &f[idx].calories) == 2) {
        idx++;
```

```
        if (idx >= 10)
            break;
    }

    fclose(fp);

    printf("%d\n", totalCalories(f, idx));
}
```



5번

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

typedef struct {
    int id;
    char name[10];
    double score1;
    double score2;
    double quiz;
} Student;

void printStudent(Student s) { printf("%d %s %.1lf %.1lf %.1lf %.1lf\n", s.
id, s.name, s.score1, s.score2, s.quiz, (s.score1 + s.score2 + s.quiz)); }

int find(Student *s, char name[], int idx) {
    for (int i = 0; i < idx; i++) {
        if (strcmp(s[i].name, name) == 0) {
            return i;
        }
    }
    return -1;
}

int main() {
    FILE *fp;
    fp = fopen("f5.txt", "r");
    int idx;
```

```
    char targetName[10];

    fscanf(fp, "%d", &idx);
    Student *s = (Student *)malloc(sizeof(Student) * idx);

    for (int i = 0; i < idx; i++) {
        s[i].id = i + 1;
        fscanf(fp, "%s %lf %lf %lf", s[i].name, &s[i].score1, &s[i].score2,
&s[i].quiz);
    }

    for (int i = 0; i < idx; i++) {
        printStudent(s[i]);
    }

    printf("\n\n");

    scanf("%s", targetName);
    int findIdx = find(s, targetName, idx);
    findIdx != -1 ? printStudent(s[findIdx]) : printf("Not Found!!!\n");
}
```