

C프로그래밍및실습

멘토링 자동 관리 프로그램

최종 보고서

제출일자: 2023.12.24

제출자명: 김성빈

제출자학번: 205224

1. 프로젝트 목표

1) 배경 및 필요성

현재 비대면으로 중학교 3학년 학생들에게 영어를 가르쳐 주는 EBS 멘토링 활동을 하고 있다. 비대면으로 진행되는 프로그램이다 보니 학생들과 메시지를 이용해 소통을 하는 경우가 많다. 보통 메시지를 보내는 상황과 멘트가 비슷한데, 매번 메시지를 직접 작성하다 보니 불편함을 느낀다. 그래서 멘티에게 보낼 메시지를 미리 저장해 출력해 주는 프로그램이 필요하다고 생각하게 되었다. 그 외에도 멘토링이 원활하게 진행되도록 하는 기능과 멘토와 멘티에게 도움이 될 수 있는 기능을 담은 프로그램을 만들어보고자 한다.

2) 프로젝트 목표

멘티에게 보낼 상황에 맞는 메시지를 멘티의 이름과 멘티의 수업 시간을 넣어서 바로 전송할 수 있도록 미리 저장된 메시지를 출력하는 프로그램을 만들고, 멘티의 수업 기록을 저장한 파일을 불러와서 멘티 별로 현재까지 진행된 수업의 횟수와 결석 횟수 출력, 멘토의 총 급여 출력을 목표로 한다. 그리고 멘티가 배운 단어를 복습할 수 있는 단어 시험 기능이나 문제 풀이 시간을 체크하는 스톱워치 기능 구현을 목표로 한다.

3) 차별점

기존 프로그램은 현재 진행하는 EBS 멘토링 프로그램을 기반으로 두고 구현한 프로그램이 아니다. 또 보통은 멘티의 입장에서 멘티에게만 도움이 되는 프로그램을 많이 구현하는데, 이 프로그램은 멘토의 편의도 고려하는 프로그램이라는 점에서 차별점이 있다.

2. 기능 계획

1) 멘티의 이름과 수업 시간 입력받기

- 프로그램 실행과 구현된 기능에 멘티들의 정보가 필요해 미리 멘티의 이름과 수업 시간을 입력받는 기능

(1) 구조체로 선언된 멘티의 이름을 입력받기

- 멘티의 이름을 순서대로 입력해 저장한다.

(2) 구조체로 선언된 멘티 별 수업 시간을 입력받기

- 멘티 별 수업 시간을 순서대로 입력해 저장한다.

2) 비밀번호 확인

- 멘티가 멘토가 사용하는 기능에 접근할 수 없도록 비밀번호 기능 구현

(1) 입력한 문자열이 비밀번호와 일치하는지 확인

- 사용자로부터 입력받은 문자열이 미리 저장된 비밀번호와 일치하는지 확인해 일치하면 다음 기능으로 넘어가고, 일치하지 않으면 비밀번호를 다시 입력받는다.

(2) 3회 이상 틀릴 시 프로그램 종료

- 입력한 문자열과 비밀번호가 3회 일치하지 않을 경우 프로그램을 종료하고, 3회가 되기 전에 맞추면 다음 기능으로 넘어간다.

3) 멘티에게 보낼 메시지 출력

- 멘토링 1시간 전과 멘티가 지각을 한 경우, 수업이 끝난 경우에 멘티에게 메시지를 보내는데, 매번 같은 문구를 작성하는 것이 불편하다고 느껴 고안한 기능

(1) 멘티의 이름과 수업 시간 정보를 미리 저장

(2) 조건문을 사용해 상황을 고르고 메시지 출력

- 멘티에게 수업 1시간 전에 보낼 수업 안내 메시지를 출력한다.

- 멘티가 지각한 경우에 보낼 메시지를 출력한다.

- 수업이 끝난 경우에 보낼 마무리 메시지를 출력한다.

4) 수업 진행 현황 출력

- 수업 기록 파일에서 총 수업 횟수와 남은 횟수, 멘티의 결석 횟수를 출력한다.
- 추가로 멘토가 지급받을 총 급여를 계산해 출력한다.

(1) 수업 기록 파일 불러오기

- 파일 입출력 기능을 사용해 멘티의 수업 기록이 저장된 파일을 불러온다.

(2) 수업 기록 횟수 계산 및 급여 계산

- 진행 시간을 기반으로 문자열 함수 strstr로 수업 시간을 계산해 멘티의 수업 횟수와 결석 횟수를 출력하고, 총 급여를 계산해 함께 출력한다.

5) 단어 퀴즈 프로그램

- 멘티가 학습한 단어를 복습할 수 있는 단어 퀴즈 프로그램 기능

(1) 학습할 단어 불러오기

- 미리 단어가 저장된 파일을 불러온다.

(2) 단어 맞추기 퀴즈를 통해 단어 학습 현황 확인

- 문자열 함수 strcmp로 멘티가 입력한 단어의 한국어 뜻이 답과 일치하는지 확인한다.
- 답과 일치하면 정답 멘트를 출력하고, 일치하지 않으면 오답 멘트와 함께 정답을 출력한다.

6) 스톱워치 기능

- 문제 풀이에 걸리는 시간을 측정할 수 있는 스톱워치 기능

(1) 사용자가 스톱워치 기능을 선택하면 시간 측정 시작

- 주어진 목록에서 스톱워치 기능을 선택하면 바로 시간 측정을 시작한다.

(2) 키보드를 누르면 스톱워치 기능 종료

- _kbhit 함수를 사용해 키보드를 누르면 시간 측정을 종료한다.

7) 프로그램 종료 기능

- 주어진 목록에서 프로그램 종료를 선택하면 프로그램을 종료한다.

(1) 할당된 메모리 해제

(2) 프로그램 종료

3. 기능구현

(1) 멘티의 이름과 수업 시간 입력받기

- 입력 : char 형태의 멘티 별 이름과 수업 시간
- 출력 : 입력받은 문자열
- name과 time을 멤버로 가지는 구조체를 선언하고 구조체 배열 동적 할당을 이용해 멘티들의 정보를 저장한다.
- 반복문, 조건문, 함수, 문자열 함수, 구조체, 포인터, 동적 메모리, 헤더 파일
- 코드 스크린샷(main.c)

```
int main() {
    int numStudents = 4; // 담당하는 멘티의 수는 4명
    int terminate = 0;

    // 멘티들의 이름은 김가가, 이나나, 박다다, 최라라
    // 수업 시간은 18:00, 19:30, 20:30, 21:00

    struct StudentInfo* students = (struct StudentInfo*)malloc(numStudents * sizeof(struct StudentInfo));
    if (students == NULL) {
        printf("메모리 할당 오류\n");
        return 1;
    }

    printf("멘토링 자동 관리 프로그램\n\n");
    printf("멘티의 이름을 입력해주세요!\n");
    inputStudentInfo(students, numStudents);

    printf("\n멘티별 수업시간을 입력해주세요!\n");
    printf("입력 형식은 00:00 입니다.\n");
    inputTimeInfo(students, numStudents);
}
```

- 코드 스크린샷(header.h)

```

struct StudentInfo {
    char* name;
    char* time;
};

void inputStudentInfo(struct StudentInfo* students, int numStudents);

void inputTimeInfo(struct StudentInfo* students, int numStudents);

```

- 코드 스크린샷(header.c)

```

void inputStudentInfo(struct StudentInfo* students, int numStudents) {
    for (int i = 0; i < numStudents; i++) {
        students[i].name = (char*)malloc(50 * sizeof(char));
        if (students[i].name == NULL) {
            printf("메모리 할당 오류\n");
            exit(1);
        }

        printf("%d번째 멘티의 이름 : ", i + 1);
        scanf_s("%s", students[i].name, 50);
    }
}

void inputTimeInfo(struct StudentInfo* students, int numStudents) {
    for (int i = 0; i < numStudents; i++) {
        students[i].time = (char*)malloc(50 * sizeof(char));
        if (students[i].time == NULL) {
            printf("메모리 할당 오류\n");
            exit(1);
        }

        printf("%s 멘티의 수업 시간 : ", students[i].name);
        scanf_s("%s", students[i].time, 50);
    }
}

```

(2) 비밀번호 확인

- 입력 : char 형태의 비밀번호, 3회까지만 입력 가능
- 출력 : 입력한 문자열이 저장된 비밀번호와 같다면 정답 멘트 출력, 같지 않다면 오답 멘트와 함께 남은 도전 횟수 출력
- 문자열을 입력받고 비밀번호의 일치 여부를 판단해 주는 문자열 출력
- 3회 이상 틀리면 프로그램이 종료되고, 3회 이내에 비밀번호를 맞추면 다음 기능에 접근 가능
- 반복문, 조건문, 함수, 문자열 함수, 배열, 헤더 파일
- 코드 스크린샷(main.c)

```

case 2:
    // 수업 진행 현황
    if (getPassword() == 1) {
        return 0;
    }
    else {
        printf("수업 진행 현황 프로그램입니다!\n");
        getNow(students, numStudents);
    }
    break;

```

- 코드 스크린샷(header.h)

```
int getPassword();
```

- 코드 스크린샷(header.c)

```

int getPassword() {
    int tryCount = 0;

    while (1) {
        printf("비밀번호를 입력하세요: ");
        scanf_s("%s", password, (int)sizeof(password));

        if (strcmp(password, "mentoring") == 0) {
            printf("맞았습니다!\n");
            printf("\n");
            return 0;
        }
        else {
            tryCount++;
            printf("틀렸습니다. 다시 입력해주세요. 기회는 %d번 남았습니다.\n", 3 - tryCount);

            if (tryCount >= 3) {
                printf("비밀번호 3회 이상 실패. 프로그램을 종료합니다.\n");
                return 1;
            }
        }
    }
}

```

(3) 멘티에게 보낼 메시지 출력

- 입력 : 비밀번호 입력을 위한 문자열과 목록 선택을 위한 int 형태의 숫자
- 출력 : 선택한 멘티와 상황에 맞는 수업 메시지 출력
- 미리 저장해 둔 멘티의 이름과 멘티 별 수업 시간을 선택하기 위해 숫자를 입력받고, 메시지를 보낼 상황을 선택하기 위해 마찬가지로 숫자를 입력받아 저장된 메시지를 출력
- 조건문, 함수, 구조체, 포인터, 헤더 파일
- 코드 스크린샷(main.c)

```

while (1) {
    int choice = -1;
    scanf_s("%d", &choice);

    switch (choice) {
    case 1:
        // 수업 메시지 출력
        if (getPassword() == 1) {
            return 0;
        }
        else {
            int numName = -1;

            printf("수업 메시지 출력 프로그램입니다!\n");
            printf("1: %s 2: %s 3: %s 4: %s\n",
                students[0].name, students[1].name, students[2].name, students[3].name);
            printf("멘티의 이름을 선택해주세요 : ");
            scanf_s("%d", &numName);

            getMessage(students, numName);
        }
        break;
    }
}

```

- 코드 스크린샷(header.h)

```
void getMessage(struct StudentInfo* students, int numName);
```

- 코드 스크린샷(header.c)

```

void getMessage(struct StudentInfo* students, int numName) {
    int situation;

    printf("상황을 선택해주세요.\n");
    printf("1. 수업 시작 전\n");
    printf("2. 멘티의 지각\n");
    printf("3. 수업 끝\n");
    scanf_s("%d", &situation);

    if (situation == 1) {
        printf("%s 멘티님, 잠시 후 %s에 멘토링이 진행될 예정입니다~! 교재와 함께 시간 맞춰 접속해주세요 :) \n",
            students[numName - 1].name, students[numName - 1].time);
    }
    else if (situation == 2) {
        printf("%s 멘티님, 수업 시작 후 10분이 경과되었습니다. 수업 참여에 어려움이 있을까요?\n",
            students[numName - 1].name);
    }
    else if (situation == 3) {
        printf("%s 멘티님, 오늘 수업도 수고하셨습니다! 다음 수업 시간에 다시 봐요 :) \n",
            students[numName - 1].name);
    }
}

```

(4) 수업 진행 현황 출력

- 입력 : 비밀번호 입력을 위한 문자열과 목록 선택을 위한 int 형태의 숫자
- 출력 : 멘티 별 수업 횟수, 남은 수업 횟수, 결석 횟수와 멘티의 총 급여를 출력
- 미리 저장된 텍스트 파일을 읽고, 텍스트 파일에서 문자열의 개수를 세어 출력한다.

- 파일에서 수업 시간과 멘티의 이름의 개수를 세고 구한 값을 사용해 멘티의 수업 횟수와 결석 횟수를 출력한다.
- 각 수업 시간 별 금액을 곱해 총 급여를 함께 출력한다.
- 조건문, 반복문, 배열, 구조체, 파일 입출력, 함수, 문자열 함수, 헤더 파일, 동적 메모리
- 코드 스크린샷(main.c)

```

case 2:
    // 수업 진행 현황
    if (getPassword() == 1) {
        return 0;
    }
    else {
        printf("수업 진행 현황 프로그램입니다!##n");
        getNow(students, numStudents);
    }
    break;

```

- 코드 스크린샷(header.h)

```

void getNow(struct StudentInfo* students, int numStudents);

```

- 코드 스크린샷(header.c)

```

void getNow(struct StudentInfo* students, int numStudents) {
    FILE* fp;
    fopen_s(&fp, "./mentoring.txt", "r");

    if (fp == NULL) {
        printf("파일을 열 수 없습니다.\n");
        return;
    }

    char line[MAX];
    int zeroCount = 0;
    int fortyFiveCount = 0;

    int* sessionCounts = (int*)calloc(numStudents, sizeof(int));
    int* absentCounts = (int*)calloc(numStudents, sizeof(int));

    while (fgets(line, MAX, fp) != NULL) {
        // 급여 계산을 위해 수업이 진행된 횟수 계산
        if (strstr(line, "0분") != NULL) {
            zeroCount++;
        }
        if (strstr(line, "45분") != NULL) {
            fortyFiveCount++;
        }

        // 멘티의 수업 진행 횟수와 결석 횟수 계산
        for (int i = 0; i < numStudents; i++) {
            if (strstr(line, students[i].name) != NULL) {
                // 수업이 진행된 총 횟수
                sessionCounts[i]++;
            }
            if (strstr(line, students[i].name) != NULL && strstr(line, "0분") != NULL) {
                // 멘티가 결석한 횟수
                absentCounts[i]++;
            }
        }
    }

    fclose(fp);

    // 총 급여 계산
    int money = 6000 * zeroCount + 12000 * fortyFiveCount;

    printf("\n");
    for (int i = 0; i < numStudents; i++) {
        printf("%s 멘티의 수업 진행 회차 : 총 %d회 / 남은 회차 : %d회\n", students[i].name,
            sessionCounts[i], 20 - sessionCounts[i]);
        printf("%s 멘티의 결석 횟수 : %d\n", students[i].name, absentCounts[i]);
    }

    printf("총 급여 : %d원\n", money);

    free(sessionCounts);
    free(absentCounts);
}

```

(5) 단어 퀴즈 프로그램

- 입력 : 목록 선택을 위한 int 형태의 숫자, 단어의 한국어 뜻인 문자열

- 출력

① 미리 저장된 텍스트 파일을 읽고 순서대로 문제를 출력

② 입력한 문자가 답과 같으면 정답 멘트를, 답과 같지 않으면 오답 멘트와 함께 정답을 출력

- 학습할 단어가 저장된 텍스트 파일의 영단어와 한국어 뜻을 이용해 단어의 뜻을 입력하고 정답을 맞추는 단어 퀴즈 프로그램

- 파일 입출력, 조건문, 반복문, 함수, 문자열 함수, 배열, 헤더 파일

- 코드 스크린샷(main.c)

```
case 3:
    // 단어 퀴즈
    printf("단어 퀴즈 프로그램입니다!\n");
    testVocabulary();
    break;
```

- 코드 스크린샷(header.h)

```
void testVocabulary();
```

- 코드 스크린샷(header.c)

```

void testVocabulary() {
    FILE* fp;
    fopen_s(&fp, "./vocabulary.txt", "r");

    if (fp == NULL) {
        printf("파일을 열 수 없습니다.\n");
        return;
    }

    char eng_word[256] = ""; // 영어 단어
    char kor_word[256] = ""; // 한국어 뜻
    char answer[256] = ""; // 멘티의 답

    int lineCount = 0;

    while (fscanf_s(fp, "%s %s", eng_word, (unsigned int)sizeof(eng_word), kor_word,
        (unsigned int)sizeof(kor_word)) != EOF)
    {
        lineCount++;
    }

    rewind(fp);

    for (int i = 0; i < lineCount; i++) {
        if (fscanf_s(fp, "%s %s", eng_word, (unsigned int)sizeof(eng_word), kor_word,
            (unsigned int)sizeof(kor_word)) != EOF)
        {
            printf("\n'%s'의 한국어 뜻은? ", eng_word);
            scanf_s("%s", answer, (unsigned int)sizeof(answer));

            if (strcmp(answer, kor_word) == 0) {
                printf("%s, 정답입니다!\n", kor_word);
            }
            else {
                printf("틀렸습니다. 정답은 %s입니다.\n", kor_word);
            }
        }
    }
    fclose(fp);
}

```

(6) 스톱워치 기능

- 입력 : 목록 선택을 위한 int 형태의 숫자
- 출력 : 시간 측정이 종료된 후 경과 시간 출력
- 사용자가 스톱워치 기능을 선택하면 시간을 측정하고 키보드를 누르면 시간 측정이 종료되어 경과 시간 출력
- 반복문, 조건문, 함수, 헤더 파일
- 코드 스크린샷(main.c)

```

case 4:
    // 스톱워치 프로그램
    printf("스톱워치 프로그램입니다!\n\n");
    stopWatch();
    break;

```

- 코드 스크린샷(header.h)

```
void stopWatch();
```

- 코드 스크린샷(header.c)

```
void stopWatch() {
    clock_t start = 0, now = 0;
    clock_t duration = 0, sec = 0, min = 0, hour = 0, milsec = 0;

    printf("바로 시작, 키보드를 누르면 종료\n");

    start = clock();

    while (1) {

        now = clock();
        duration = now - start;

        sec = duration / CLOCKS_PER_SEC;
        milsec = duration % CLOCKS_PER_SEC;
        hour = sec / 3600;
        min = (sec % 3600) / 60;
        sec = sec % 60;

        printf("%d:%02d:%02d.%03d\n", hour, min, sec, milsec);

        if (_kbhit()) {
            break;
        }
    }
    printf("\n");
}
```

(7) 프로그램 종료 기능

- 입력 : 목록 선택을 위한 int 형태의 숫자
- 출력 : 프로그램 종료를 알리는 문구 출력
- 사용자가 목록에서 프로그램 종료를 선택하면 할당된 동적 메모리를 해제하고 프로그램을 종료
- 조건문, 반복문, 함수, 동적 메모리, 구조체, 헤더 파일
- 코드 스크린샷(main.c)

```

        case 5:
            // 종료
            terminate = 1;
            break;

        default: // 사용자가 1~5가 아닌 다른 값을 입력할 경우
            printf("잘못된 선택입니다. 다시 선택하세요.\n");
    }

    if (terminate == 1) {
        printf("종료를 선택하셨습니다. 프로그램을 종료합니다.");
        freeStudentInfo(students, numStudents);
        break;
    }
}

```

- 코드 스크린샷(header.h)

```
void freeStudentInfo(struct StudentInfo* students, int numStudents);
```

- 코드 스크린샷(header.c)

```

void freeStudentInfo(struct StudentInfo* students, int numStudents) {
    for (int i = 0; i < numStudents; i++) {
        free(students[i].name);
        free(students[i].time);
    }
    free(students);
}

```

4. 테스트 결과

(1) 멘티의 이름과 수업 시간 입력받기

- 프로그램 실행과 구현된 기능에 멘티들의 정보가 필요해 미리 멘티의 이름과 수업 시간을 입력받는 기능

- 테스트 결과 스크린샷

멘토링 자동 관리 프로그램

멘티의 이름을 입력해주세요!

1번째 멘티의 이름 : 김가가
 2번째 멘티의 이름 : 이나나
 3번째 멘티의 이름 : 박다다
 4번째 멘티의 이름 : 최라라

멘티별 수업시간을 입력해주세요!

입력 형식은 00:00 입니다.
 김가가 멘티의 수업 시간 : 18:00
 이나나 멘티의 수업 시간 : 19:30
 박다다 멘티의 수업 시간 : 20:30
 최라라 멘티의 수업 시간 : 21:00

1: 김가가 2: 이나나 3: 박다다 4: 최라라
 1: 18:00 2: 19:30 3: 20:30 4: 21:00

(2) 비밀번호 확인

- 입력한 문자열과 미리 저장된 비밀번호를 비교해 3회 이상 틀리면 프로그램이 종료되고, 3회 이내에 비밀번호를 맞추면 다음 기능에 접근 가능
- 테스트 결과 스크린샷

① 비밀번호를 바로 맞춘 경우

```
-----
메뉴를 선택해주세요.
1. 수업 메시지 출력
2. 수업 진행 현황
3. 단어 퀴즈
4. 스톱워치
5. 프로그램 종료
-----
2
비밀번호를 입력하세요: mentoring
맞았습니다!

수업 진행 현황 프로그램입니다!
```

② 비밀번호를 3회 이내에 맞춘 경우

```
-----
메뉴를 선택해주세요.
1. 수업 메시지 출력
2. 수업 진행 현황
3. 단어 퀴즈
4. 스톱워치
5. 프로그램 종료
-----
2
비밀번호를 입력하세요: aa
틀렸습니다. 다시 입력해주세요. 기회는 2번 남았습니다.
비밀번호를 입력하세요: bb
틀렸습니다. 다시 입력해주세요. 기회는 1번 남았습니다.
비밀번호를 입력하세요: mentoring
맞았습니다!

수업 진행 현황 프로그램입니다!
```

③ 비밀번호를 3회 이상 틀릴 경우

```

-----
메뉴를 선택해주세요.
1. 수업 메시지 출력
2. 수업 진행 현황
3. 단어 퀴즈
4. 스톱워치
5. 프로그램 종료
-----
2
비밀번호를 입력하세요: aa
틀렸습니다. 다시 입력해주세요. 기회는 2번 남았습니다.
비밀번호를 입력하세요: bb
틀렸습니다. 다시 입력해주세요. 기회는 1번 남았습니다.
비밀번호를 입력하세요: cc
틀렸습니다. 다시 입력해주세요. 기회는 0번 남았습니다.
비밀번호 3회 이상 실패. 프로그램을 종료합니다.

C:\Users\USER\Desktop\Test\x64\Debug\Test.exe(프로세스 3212개)이(가) 종료되었습니다(코드: 0개).
이 창을 닫으려면 아무 키나 누르세요...

```

(3) 멘티에게 보낼 메시지 출력

- 멘티의 이름과 메시지를 보낼 상황을 선택하고 각 상황에 맞는 메시지 출력
- 테스트 결과 스크린샷

① 수업 시작 전

```

-----
메뉴를 선택해주세요.
1. 수업 메시지 출력
2. 수업 진행 현황
3. 단어 퀴즈
4. 스톱워치
5. 프로그램 종료
-----
1
비밀번호를 입력하세요: mentoring
맞았습니다!

수업 메시지 출력 프로그램입니다!
1: 김가가 2: 이나나 3: 박다다 4: 최라라
멘티의 이름을 선택해주세요 : 2
상황을 선택해주세요.
1. 수업 시작 전
2. 멘티의 지각
3. 수업 끝
1
이나나 멘티님, 잠시 후 19:30에 멘토링이 진행될 예정입니다~! 교재와 함께 시간 맞춰 접속해주세요 :)

```


② 멘티의 지각

```
-----
메뉴를 선택해주세요.
1. 수업 메시지 출력
2. 수업 진행 현황
3. 단어 퀴즈
4. 스톱워치
5. 프로그램 종료
-----
1
비밀번호를 입력하세요: mentoring
맞았습니다!

수업 메시지 출력 프로그램입니다!
1: 김가가 2: 이나나 3: 박다다 4: 최라라
멘티의 이름을 선택해주세요 : 4
상황을 선택해주세요.
1. 수업 시작 전
2. 멘티의 지각
3. 수업 끝
2
최라라 멘티님, 수업 시작 후 10분이 경과되었습니다. 수업 참여에 어려움이 있을까요?
```

③ 수업 끝

```
-----
메뉴를 선택해주세요.
1. 수업 메시지 출력
2. 수업 진행 현황
3. 단어 퀴즈
4. 스톱워치
5. 프로그램 종료
-----
1
비밀번호를 입력하세요: mentoring
맞았습니다!

수업 메시지 출력 프로그램입니다!
1: 김가가 2: 이나나 3: 박다다 4: 최라라
멘티의 이름을 선택해주세요 : 3
상황을 선택해주세요.
1. 수업 시작 전
2. 멘티의 지각
3. 수업 끝
3
박다다 멘티님, 오늘 수업도 수고하셨습니다! 다음 수업 시간에 다시 봐요 :)
```

(4) 수업 진행 현황 출력

- 파일에서 수업 시간과 멘티 별 이름의 개수를 세고 구한 값을 사용해 멘티의 수업 횟수와 결석 횟수를 출력하고, 각 수업 별 금액을 곱해 총 급여를 함께 출력한다.

- 테스트 결과 스크린샷

```

-----
메뉴를 선택해주세요.
1. 수업 메시지 출력
2. 수업 진행 현황
3. 단어 퀴즈
4. 스톱워치
5. 프로그램 종료
-----
2
비밀번호를 입력하세요: mentoring
맞았습니다!

수업 진행 현황 프로그램입니다!

김가가 멘티의 수업 진행 회차 : 총 4회 / 남은 회차 : 16회
김가가 멘티의 결석 횟수 : 1
이나라 멘티의 수업 진행 회차 : 총 3회 / 남은 회차 : 17회
이나라 멘티의 결석 횟수 : 1
박다다 멘티의 수업 진행 회차 : 총 2회 / 남은 회차 : 18회
박다다 멘티의 결석 횟수 : 1
최라라 멘티의 수업 진행 회차 : 총 4회 / 남은 회차 : 16회
최라라 멘티의 결석 횟수 : 2
총 급여 : 126000원

```

(5) 단어 퀴즈 프로그램

- 학습할 단어가 저장된 텍스트 파일의 영단어와 한국어 뜻을 이용해 영단어의 뜻을 입력하는 단어 퀴즈 프로그램
- 테스트 결과 스크린샷

```

-----
메뉴를 선택해주세요.
1. 수업 메시지 출력
2. 수업 진행 현황
3. 단어 퀴즈
4. 스톱워치
5. 프로그램 종료
-----
3
단어 퀴즈 프로그램입니다!

'apple'의 한국어 뜻은? 사과
사과, 정답입니다!

'bear'의 한국어 뜻은? 문
틀렸습니다. 정답은 곰입니다.

'cake'의 한국어 뜻은? 케이크
케이크, 정답입니다!

'door'의 한국어 뜻은? 문
문, 정답입니다!

```

(6) 스톱워치 기능

- 사용자가 스톱워치 기능을 선택하면 시간을 측정하고 키보드를 누르면 시간 측정이 종료되어 경과 시간 출력

- 테스트 결과 스크린샷

```
-----
메뉴를 선택해주세요.
1. 수업 메시지 출력
2. 수업 진행 현황
3. 단어 퀴즈
4. 스톱워치
5. 프로그램 종료
-----
4
스톱워치 프로그램입니다!

바로 시작, 키보드를 누르면 종료
0:00:05.835
```

(7) 프로그램 종료 기능

- 주어진 목록에서 프로그램 종료를 선택하면 프로그램을 종료하는 기능
- 테스트 결과 스크린샷

```
-----
메뉴를 선택해주세요.
1. 수업 메시지 출력
2. 수업 진행 현황
3. 단어 퀴즈
4. 스톱워치
5. 프로그램 종료
-----
5
종료를 선택하셨습니다. 프로그램을 종료합니다.
C:\Users\USER\Desktop\Test\x64\Debug\Test.exe(프로세스 33480개)이(가) 종료되었습니다(코드: 0개).
이 창을 닫으려면 아무 키나 누르세요...
```

5. 계획 대비 변경 사항

1) 비밀번호 확인 기능 추가

- 이전 : 비밀번호를 확인하는 과정 없이 바로 프로그램의 기능 실행
- 이후 : 멘토가 사용하는 기능은 비밀번호를 확인하는 과정 추가
- 사유 :

멘토링 자동 관리 프로그램은 멘토와 멘티가 사용하는 기능이 함께 있는 프로그램이다. 따라서 멘토를 위해 구현된 기능에는 멘티가 접근할 수 없도록 비밀번호 기능 구현이 필요하다고 판단하였다.

2) 프로젝트 일정 변경

- 이전 : 수업 진행 현황 출력 기능과 단어 퀴즈 기능을 기존에 계획한 프로젝트 일정인 11/24까지 완료
- 이후 : 수업 진행 현황 출력 기능과 단어 퀴즈 기능을 다음 중간 구현물 과제 제출일인 12/10까지 완료
- 사유 : 파일 입출력이 사용되는 기능이므로 해당 범위까지 수업을 듣고 난 후 기능을 구현할 수 있을 것이라고 판단하였다.

3) 스톱워치 기능 추가

- 이전 : 스톱워치 기능 없이 기존에 구상한 기능만 구현
- 이후 : 새로운 기능인 스톱워치 기능 추가
- 사유 : 멘티가 학습을 하며 문제를 풀 때 시간이 얼마나 걸리는지를 측정하기 위한 스톱워치 기능이 필요하다고 생각해 기능을 추가로 고안하였다.

4) 단어장 프로그램 기능 이름 변경

- 이전 : 단어장 프로그램
- 이후 : 단어 퀴즈 프로그램
- 사유 : 단어 맞추기 퀴즈로 멘티의 단어 학습 현황을 확인하므로 단어장 프로그램보다는 단어 퀴즈 프로그램이 더 기능에 잘 어울린다고 생각해 이름을 변경하였다.

5) 기능 계획 및 구현, 테스트에서 기능 작성 순서 변경

- 이전 : 구현한 순서대로 기능 작성
- 이후 : 프로그램 실행 시 실행되는 순서대로 기능 작성
- 사유 : 최종 보고서이므로 최종 프로그램이 실행될 때 실행되는 기능 순대로 작성해야 평가하기 편리할 것이라고 판단하여 기능 계획 및 기능 구현, 테스트 순서를 변경하였다.

6. 느낀점

수업을 통해 c언어와 github를 처음 접하게 되었는데, 수업 초반에는 마냥 어렵게 느껴졌지만 사용하다 보니 점점 익숙해져 개인적으로 얻어 가는 게 많은 수업이라서 만족스러웠다.

개념을 배우고 무작정 외워서 손 코딩으로 시험을 보는 것보다 프로젝트 형식으로 내가 구현하고 싶은 기능을 생각하고, 구현해 보는 과정이 손 코딩형 시험보다 프로그래밍 언어 습득에 확실히 더 도움이 된다고 느꼈다.

프로젝트를 진행하며 코드를 작성할 때 다양한 이유로 오류들이 발생하는데, 대부분은 함수의 형태를 변경하고 헤더 파일의 함수에는 적용하지 않는 등의 오류였기 때문에, 코드를 살피는 주의력을 높일 필요성을 느꼈다.

코드를 작성할 때 대부분 익숙한 기능만 사용하게 되는데, 필수로 새롭게 배운 기능들을 적용해 보는 태도를 앞으로 팀 프로젝트나 개인 프로젝트를 진행할 때도 가지면 좋을 것 같다.