학생관리프로그램.exe

# 학생 관리 프로그램

[3조 발표]

한형빈, 추형욱, 조민석

목차

PRESENTATION

# OVERVIEW

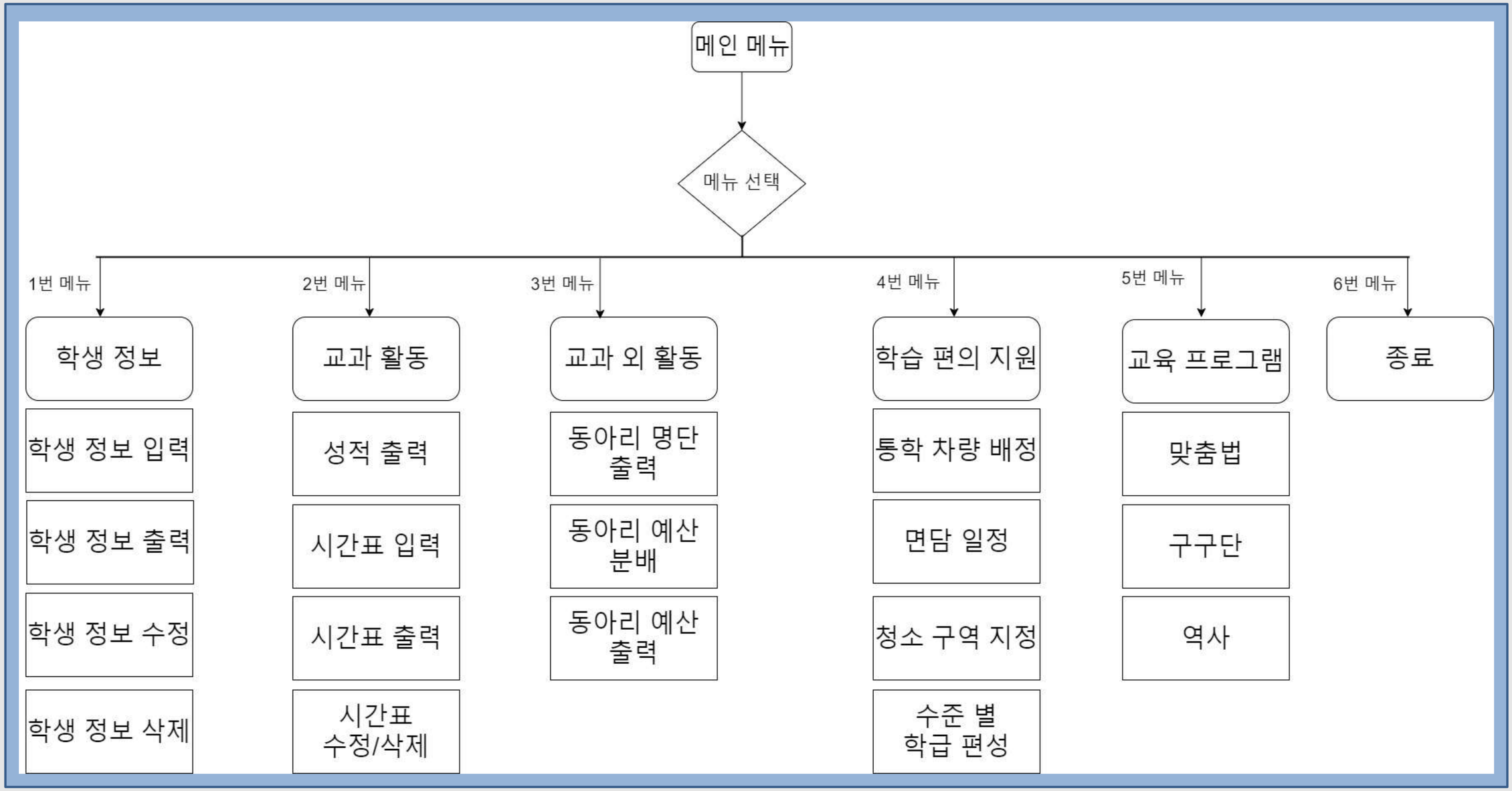
1. 프로젝트 개요
2. 역할 소개
3. 세부 코드 소개
4. 자체 평가 의견
5. Q & A

# 프로젝트 목표

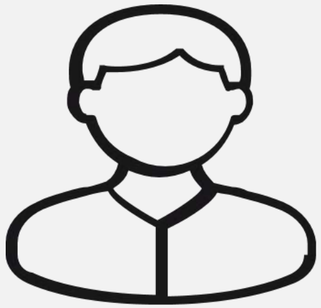
C언어만으로
파일 입출력을 이용하는
**학생 관리 프로그램**
제작

- ✓ 파일 입출력 사용
- ✓ 구조체 사용
- ✓ 조건문 사용
- ✓ 반복문 사용
- ✓ 랜덤 난수 사용
- ✓ 헤더 함수 따로 선언
- ✓ Locale 유니코드 처리

# 프로젝트 개요


메인 메뉴
메뉴 선택
1번 메뉴
학생 정보
학생 정보 입력
학생 정보 출력
학생 정보 수정
학생 정보 삭제
2번 메뉴
교과 활동
성적 출력
시간표 입력
시간표 출력
시간표 수정/삭제
3번 메뉴
교과 외 활동
동아리 명단 출력
동아리 예산 분배
동아리 예산 출력
4번 메뉴
학습 편의 지원
통학 차량 배정
면담 일정
청소 구역 지정
수준 별 학급 편성
5번 메뉴
교육 프로그램
맞춤법
구구단
역사
6번 메뉴
종료

# 역할 소개



## 한형빈(팀장)

- ✓코드 개요 작성
- ✓ 면담 일정 코드
- ✓ 청소 구역 지정 코드
- ✓ 역사 퀴즈 랜덤 출제 코드
- ✓ 요구사항 정의서 작성
- ✓ 회의록 작성

## 추형욱

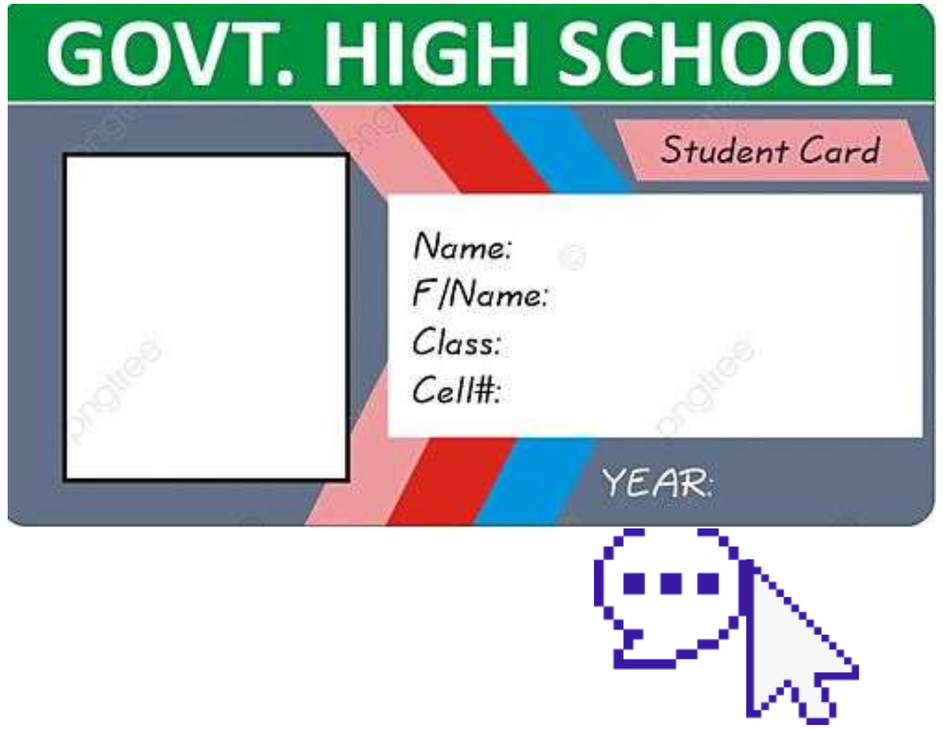
- ✓ 성적순 반 배정
- ✓ 통학 차량 배정
- ✓ 구구단 랜덤 출제 코드
- ✓ 발표자료 정리
- ✓ 시연 영상 제작

## 조민석

- ✓ 파일 입출력
- ✓ 메인 화면 코드
- ✓ 학생 정보 관련 기능 코드
- ✓ 동아리 관련 기능 코드
- ✓ 시간표 관련 기능 코드
- ✓ 코드 모듈화
- ✓ 맞춤법 퀴즈
- ✓ 오류 수정

# 파일 입출력 저장

```
FILE* file1 = fopen("학생저장목록파일.txt", "wt");
if (file1 != NULL) {
    fprintf(file1, "%d\n", count);
    for (int i = 0; i < count; i++) {
        fprintf(file1, "%d\t", student[i].snum);
        fwprintf(file1, L"%ls\t", student[i].sname);
        fprintf(file1, "%s\t", student[i].callnumber);
        fprintf(file1, "%d\t", student[i].birth);
        fprintf(file1, "%c\t", student[i].adress);
        fprintf(file1, "%d\t", student[i].Kor);
        fprintf(file1, "%d\t", student[i].Eng);
        fprintf(file1, "%d\t", student[i].Math);
        fprintf(file1, "%f\t", student[i].Avg);
        fprintf(file1, "%s\t", student[i].dong);
        fprintf(file1, "%s\t", student[i].vms);
        fprintf(file1, "%d\t", student[i].clubBudget);
        fprintf(file1, "%s\t", student[i].scoreclass);
        fprintf(file1, "\n");
    }
    printf("저장이 성공적으로 완료 됐습니다\n");
}
else {
    printf("파일 저장에 실패 했습니다.");
}
```



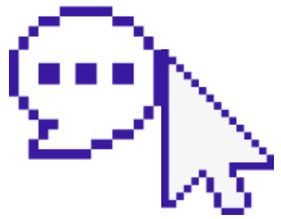
# 파일 입출력 불러오기, 학생 테이블 구조체 선언

```
FILE* file1 = fopen("학생저장목록파일.txt", "rt");
if (file1 != NULL) {
    fscanf(file1, "%d", count);
    // 기본 정보 및 시간표 정보 읽기
    for (int i = 0; i < *count; i++) {
        fscanf(file1, "%d", &student[i].snum);
        fwscanf(file1, L"%ls", student[i].sname);
        fscanf(file1, "%s", student[i].callnumber);
        fscanf(file1, "%d", &student[i].birth);//19890430 20020430
        fscanf(file1, "\t%c", &student[i].adress);
        fscanf(file1, "%d", &student[i].Kor);
        fscanf(file1, "%d", &student[i].Eng);
        fscanf(file1, "%d", &student[i].Math);
        fscanf(file1, "%f", &student[i].Avg);
        fscanf(file1, "%s", student[i].dong);
        fscanf(file1, "%s", student[i].vms);
        fscanf(file1, "%d", &student[i].clubBudget);
        fscanf(file1, "%s", &student[i].scoreclass);
        fscanf(file1, "\n");
    }
    fscanf(file1, "\n");
```

```
typedef struct _Student {
    int snum; // 학번
    wchar_t sname[20]; //이름
    char adress; // 주소
    char callnumber[15]; // 전화번호
    char cleanadress[3]; // 청소구역
    int birth; // 생년월일
    int Kor; // 국어
    int Eng; // 영어
    int Math; // 수학
    float Avg; // 평균
    char vms[20]; // 봉사활동
    char dong[20]; // 동아리
    int clubBudget; // 동아리 예산
} Student;
```

# 이름 입력 시 예외처리(한글만 입력)

```
while (1) {
    wint_t c;
    wscanf(L"%ls", student[*count].sname);

    while ((c = getwchar()) != L'\n' && c != WEOF);  // 입력 버퍼 비우기

    if (is_korean_string(student[*count].sname)) {
        break;
    }
    else {
        printf("오류: 올바른 한글 이름을 입력하세요.\n");
        printf("이름 : ");
    }
}
```



# 이름 입력 시 예외처리(한글만 입력)

```
int is_korean_string(const wchar_t* str) {
    int koreanCount = 0;

    while (*str) {
        if (iswalpha(*str) && ((*str < L'가' || *str > L'힣'))) {
            // 알파벳이나 다른 문자가 있으면서, 한글이 아닌 경우
            return 0;
        }

        if (*str >= L'가' && *str <= L'힣'
            koreanCount++;
        }

        str++;
    }

    return (koreanCount >= 2);  // 한글이
}
```

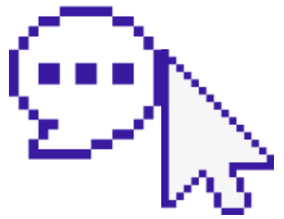
학생관리프로그램

```
===== 학생 정보 입력하기 [1 / 100] =====
학번 (1반 1번 1001, 2반 1번 2001, 3반 1번 3001) : 1001
이름 : a
오류: 올바른 한글 이름을 입력하세요.
이름 : 123
오류: 올바른 한글 이름을 입력하세요.
이름 :
```

# 학생 정보 이름으로 찾기

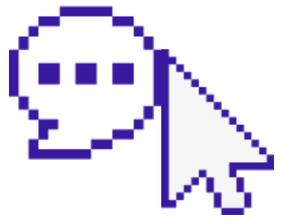
```
printf("찾을 이름을 입력 하세요 : ");
wchar_t name[20] = { NULL };
wscanf(L"%ls", name);
printName(*count, student, name);
```

```
학생관리프로그램
===== 학생 정보 출력하기 =====
1. 전체 출력하기          2. 선택 출력하기 3. 취소
원하는 메뉴를 입력하세요 : 입력하세요 (1 ~ 3): 2
찾을 이름을 입력 하세요 : 홍윤하
===== 학생 정보 선택 출력하기(홍윤하) =====
학번 : 1001 이름 : 홍윤하 생년월일 : 20100909 주소 : b 전화번호 : 010-1234-5678
```



# 학생 정보 이름으로 찾기

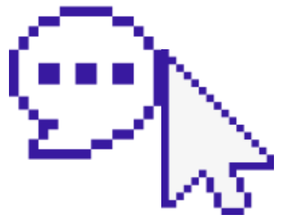
```
void printName(int count, Student* student, wchar_t name[20])
{
    printf("===== 학생 정보 선택 출력하기(%ls) =====\n", name);
    for (int i = 0; i < count; i++)
    {
        if (strcmp(name, student[i].sname) == 0)
        {
            printf("학번 : %d 이름 : %ls 생년월일 : %d.%d.%d 주소 : %c 전화번호
: %.3s-%.4s-%.4s \t \n\n",
                student[i].snum, student[i].sname, (student[i].birth /
10000), (student[i].birth % 10000) / 100, student[i].birth % 100,
student[i].adress, student[i].callnumber,
                student[i].callnumber + 3, student[i].callnumber + 7);
            return;
        }
    }
    printf("찾는 학생이 존재하지 않습니다.\n");
}
```



# 주소 입력 예외 처리

```c
while (1) {
    char buffer[100];  // 충분한 크기의 문자열을 저장할 버퍼
    printf("수정할 주소 (ex: a, b, c, ...) : ");
    if (scanf("%99s", buffer) == 1) {
        int c;
        // 입력 버퍼를 비우기 위해 개행 문자까지 모두 읽어서 처리
        while ((c = getchar()) != '\n' && c != EOF);

        // 입력된 값이 알파벳이고, 문자열 길이가 1이면 종료
        if (isalpha(buffer[0]) && buffer[1] == '\0') {
            student->adress = buffer[0];
            break;
        }
        else {
            printf("오류: 알파벳 하나만 입력하세요.\n");
        }
    }
    else {
        int c;
        while ((c = getchar()) != '\n' && c != EOF); // 입력 버퍼 비우기
        printf("오류: 알파벳 하나만 입력하세요.\n");
    }
}
```

```
수정할 주소 (ex: a, b, c, ...) : c
수정할 전화번호 : (11자리 전체입력)01099999999
수정 후 학생 정보:
학번: 1001
이름: 홍윤하
생년월일: 20100909
전화번호: 01099999999
주소: c
국어: 90
영어: 80
수학: 70
평균: 37.50
봉사활동: 양로원
동아리: 사진부
```



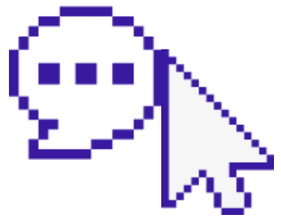
# 시간표 구조체 정의

```c
#define MAX_DAYS 5 //시간표 요일
#define MAX_PERIODS 6 //시간표 시간
#define MAX_CLASSES 3 //시간표 반

typedef struct {
    char courses[MAX_CLASSES][MAX_DAYS][MAX_PERIODS][50]; // 5일 x 6교시의 강의명을 저장
} Timetable;
```

# 시간표 입력 및 저장

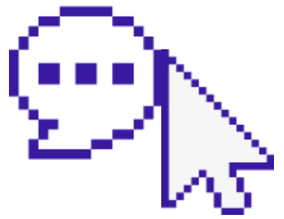
```
void addCourseMenu(Timetable* timetable) {
    int cls, day, startPeriod, endPeriod;
    char courseName[50];

    cls = getInput("반 입력", 1, 3);
    day = getInput("요일 입력", 1, 5);
    startPeriod = getInput("시작 교시 입력", 1, 6);
    endPeriod = getInput("종료 교시 입력", startPeriod, 6);

    printf("과목명 입력: ");
    scanf("%s", courseName);

    bool added = addCourse(timetable, cls - 1, day - 1, startPeriod - 1,
endPeriod - 1, courseName);
    if (added) {
        printf("강의가 추가되었습니다.\n");
    }
}
```

```
반 입력 (1 ~ 3): 1
요일 입력 (1 ~ 5): 1
시작 교시 입력 (1 ~ 6): 1
종료 교시 입력 (1 ~ 6): 4
과목명 입력 : 국어
강의가 추가되었습니다.
```



# 시간표 입력 및 저장

```
int getInput(const char* prompt, int min, int max) {
    char input[100];
    int value;
    char extra;
    while (1) {
        printf("%s (%d ~ %d): ", prompt, min, max);

        if (fgets(input, sizeof(input), stdin) == NULL) {
            printf("입력 오류가 발생했습니다.\n");
            exit(EXIT_FAILURE);
        }

        // 문자열에 엔터가 포함되어 있으면 제거
        size_t len = strlen(input);
        if (len > 0 && input[len - 1] == '\n') {
            input[len - 1] = '\0';
        }

        // 입력 받은 문자열을 정수로 변환
        if (sscanf(input, "%d %c", &value, &extra) == 1) {
            if (value >= min && value <= max) {
                break;
            }
            else {
                printf("잘못된 범위의 입력입니다. 다시 입력하세요.\n");
            }
        }
        else {
            printf("입력 오류가 발생했습니다. 다시 입력하세요.\n");
        }
    }

    return value;
}
```



# 시간표 내 강의 중복 제한

```
bool addCourse(Timetable* timetable, int cls, int day, int startPeriod, int
endPeriod, const char* courseName) {
    // 범위 내에 이미 강의가 있는지 확인
    for (int period = startPeriod; period <= endPeriod; period++) {
        if (timetable->courses[cls][day][period][0] != '\0') {
            printf("이미 해당 교시에 강
            return false;  // 강의 추
        }
    }

    // 범위 내에 강의명 저장
    for (int period = startPeriod; pe
        snprintf(timetable->courses[cls][day][period], sizeof(timetable-
>courses[cls][day][period]), "%s", courseName);
    }

    return true;  // 강의 추가 성공
}
```

학생관리프로그램

```
반 입력 (1 ~ 3): 1
요일 입력 (1 ~ 5): 1
시작 교시 입력 (1 ~ 6): 2
종료 교시 입력 (2 ~ 6): 3
과목명 입력 : 국어
이미 해당 교시에 강의가 있습니다. 다른 교시를 선택하세요.
```

# 시간표 출력

```
void printTimetable(const Timetable* timetable, int cls) {
    printf("    월    화    수    목    금\n");
    for (int period = 0; period < MAX_PERIODS; period++) {
        printf("%d ", period + 1);
        for (int day = 0; day < MAX_DAYS; day++) {
            // 문자열이 비어있는 경우를 처리하여 출력
            if (timetable->courses[cls][day][period][0] != '\0') {
                printf("| %-4s ", timetable->courses[cls][day][period]);
            }
            else {
                printf("|      ");  // 비어있는 경우
            }
        }
        printf("|\n");
    }
}
```

```
파일에서 시간표를 불러왔습니다.
반 입력 (1-3): 1
    월    화    수    목    금
1 | 수학 |      |      |      |
2 | 수학 |      | 과학 |      | 국어 |
3 |      | 국어 | 과학 | 영어 | 국어 |
4 |      | 국어 |      | 영어 |      |
5 |      | 국어 |      | 영영어 |      |
6 |      |      |      | 영어 |      |
계속하려면 아무 키나 누르십시오 . . .
```

# 동아리 예산 분배

```
void distributeClubBudget(int count, Student* student, int totalBudget, char
clubName[20]) {
    int clubExists = 0;
    for (int i = 0; i < count; i++) {
        if (strcmp(student[i].dong, clubName) == 0) {
            clubExists = 1;
            break;
        }
    }

    if (clubExists) {
        // 동아리 예산 분배 함수 호출
            // 각 학생에게 동일한 예산을 할당
        int baseBudget = totalBudget / count;

        // 입력받은 동아리와 이름이 같은 학생들에게 동일한 예산을 할당
        for (int i = 0; i < count; ++i) {
            if (strcmp(student[i].dong, clubName) == 0) {
                student[i].clubBudget = baseBudget;
            }
        }
        printf("%s 동아리에 예산이 분배되었습니다.\n", clubName);
    }
    else {
        printf("%s 동아리는 존재하지 않습니다. 예산을 분배하지 않습니다.\n", clubName);
    }
}
```

# 동아리 예산 분배

```
// 사용자로부터 예산 총액 입력
printf("동아리 예산의 총액을 입력하세요: ");
int totalBudget = inputInt(0, 1000000);

// 동아리 예산 분배 함수 호출
distributeClubBudget(count, student, totalBudget, "독서부");
distributeClubBudget(count, student, totalBudget, "영화부");
distributeClubBudget(count, student, totalBudget, "사진부");
printf("\n");
printClubBudget(count, student);
```

```
동아리 예산의 총액을 입력하세요: 입력하세요 (0 ~ 1000000): 800000
독서부 동아리에 예산이 분배되었습니다.
영화부 동아리에 예산이 분배되었습니다.
사진부 동아리에 예산이 분배되었습니다.

동아리          동아리 예산
----------------------
독서부          292600 원
영화부          319200 원
사진부          186200 원
----------------------
남은 예산 : 2000계속하려면 아무 키나 누르십시오 . . .
```

# 통학 차량 배정

```
void car(int count, Student* student)
{
    printf("===통학시 차량조정===\n");
    if (count > 0) {
        rewind(stdin);
        int addressList[26] = { NULL };

        for (int i = 0; i < count; i++)
        {
            rewind(stdin);
            char address;
            address = changeBigChar(student[i].adress);

            addressList[address - 'A']++;

        }

        for (int i = 0; i < 26; i++)
        {
            if (addressList[i] != 0)
            {
                printf("%c지역 %d명입니다\n", i + 'A', addressList[i]);
            }
        }
```

# 통학 차량 배정 출력

```
        for (int i = 0; i < 26; i++)
        {

            if (addressList[i] >= 4)
            {
                if (addressList[i] % 4 == 0)
                    printf("%c지역 통학차량 %d대 배치필요\n", i + 'A',
(addressList[i] / 4));
                else if (addressList[i] % 4 != 0)
                    printf("%c지역 통학차량 %d대 배치필요\n", i + 'A',
(addressList[i] / 4) + 1);
            }
            else if (addressList[i] <= 3 && addressList[i] > 0)
                printf("3명이하인 %c지역은 차량지원불가 통학비지원확인바람\n", i
+ 'A');

        }
    }
    else {
        printf("등록된 학생이 없습니다.\n");
    }
    system("pause");
}
```

```
학생관리프로그램
====================
=학생 관리 프로그램=
====================
1. 학생 정보
2. 교과 활동
3. 교과 외 활동
4. 학습 편의 지원
5. 교육 프로그램
6. 종료
입력하세요 (1 ~ 6): 4


1. 통학차량 배정
2. 면담 일정
3. 청소 구역 지정
4. 수준별 학습 반 편성
5. 메인메뉴로 돌아가기
입력하세요 (1 ~ 5): 1
===통학시 차량조정===
A지역 7명입니다
B지역 12명입니다
C지역 11명입니다
A지역 통학차량 2대 배치필요
B지역 통학차량 3대 배치필요
C지역 통학차량 3대 배치필요
계속하려면 아무 키나 누르십시오 . . .
```

학생관리프로그램.exe

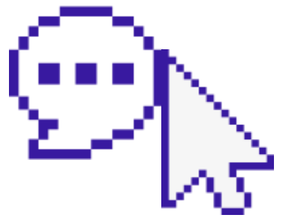
# 면담 일정 시연 영상

# 면담 일정 사용 함수

```
while (!inputDate(&startYear, &startMonth, &startDay)); // 연월일 입력에 오류가 없으면
while (!inputMeetingHour(&meetingHour)); // 면담 시작 시간에 오류가 없으면

while (!inputExcludeWeekday(&excludeWeekday)); // 불가능한 요일 입력에 오류가 없으면

while (!inputExcludeDate(&excludeYear, &excludeMonth, &excludeDay)); // 불가능한 날짜 입력에 오류가 없으면

while (!inputClass(&selectedClass)); // 출력할 반 입력에 오류가 없으면

generateClassMeetingSchedule(selectedClass, classes, startYear, startMonth, startDay, meetingHour, excludeWeekday, excludeYear, excludeMonth, excludeDay,&student,count); // 면담 일정 생성
```

# 면담 시작 날짜 시간

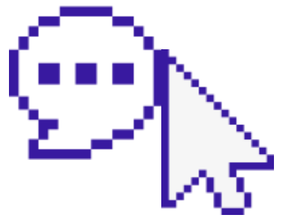
```c
int inputDate(int* year, int* month, int* day) {
    printf("면담 시작 날짜 입력 (예: 2024. 01. 01.): ");
    int inputResult = scanf("%04d. %02d. %02d.", year, month, day);
    // 날짜 입력이 정수가 아니거나, 날짜가 잘못 입력된 경우에 오류 메시지 출력
    if (inputResult != 3 || *year <= 0 || *month <= 0 || *day <= 0 || *month >
12 || *day > daysInMonth(*year, *month)) {
        printf("\"날짜 정보 입력 오류\"\n");
        // 실패한 입력값 무시
        while (getchar() != '\n');
        printf("\n");
        return 0; // 날짜 입력을 실패하면 다시 입력받는 문구로 반복
    }
    else {
        return 1;  // 성공
    }
}
```



# 면담 시작 날짜 시간

```c
int inputMeetingHour(int* meetingHour) {
    printf("면담 시작 시간 입력 (예: 9 ~ 16): ");
    int inputResult = scanf("%d", meetingHour);

    // 입력값이 정수가 아니거나, 시간 범위를 벗어난 경우를 처리
    if (inputResult != 1 || *meetingHour < 9 || *meetingHour > 16) {
        printf("\"시간 정보 입력 오류\"\n");
        // 실패한 입력값 무시
        while (getchar() != '\n');
        printf("\n");
        return 0;

    }
    else {
        return 1;
    }
}
```

```
면담 시작 날짜 입력 (예: 2024. 01. 01.): 2024.01.01
면담 시작 시간 입력 (예: 8): 4
```

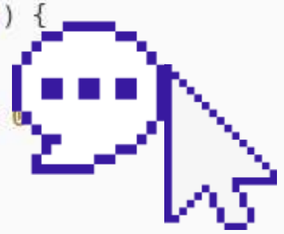


# 면담 불가능 요일 예외 처리

```c
int inputExcludeWeekday(int* excludeWeekday, int* excludeWeekday1)
    printf("면담이 불가능한 요일 입력 (월요일: 월, ..., 금요일: 금, 없으면
");
    char weekdayStr[20]; // 최대 요일 문자열 길이는 3 (ex: 월)
    int inputResult = scanf("%7s", weekdayStr);
    int con = 0;

    // 입력값이 문자열이 아니거나, 요일 문자열이 아닌 경우를 처리
    if (inputResult != 1 || getchar() != '\n' ||
weekdayStringToNumber(weekdayStr) == -1) {
        printf("\"요일 정보 입력 오류\"\n");
        // 실패한 입력값 무시
        while (getchar() != '\n');
    }
    else if (weekdayStringToNumber(weekdayStr) == 7) {
        *excludeWeekday = weekdayStringToNumber(weekdayStr);
        return 1;
    }
    else {
        *excludeWeekday = weekdayStringToNumber(weekdayStr); // 요일
숫자로 변환하여 저장
        return 0;
    }
}
```

```c
int weekdayStringToNumber(const char* weekdayStr) {
    int dayNumber = -1; // 기본적으로 -1로 초기화하여 잘못된 입력을 처리

    // 요일 문자열을 비교하여 해당하는 요일의 숫자를 반환
    if (strcmp(weekdayStr, "일") == 0) {
        dayNumber = 0;
    }
    else if (strcmp(weekdayStr, "월") == 0) {
        dayNumber = 1;
    }
    else if (strcmp(weekdayStr, "화") == 0) {
        dayNumber = 2;
    }
    else if (strcmp(weekdayStr, "수") == 0) {
        dayNumber = 3;
    }
    }
    else if (strcmp(weekdayStr, "금") == 0) {
        dayNumber = 5;
    }
    else if (strcmp(weekdayStr, "토") == 0) {
        dayNumber = 6;
    }
    else if (strcmp(weekdayStr, "없음") == 0
        dayNumber = 7;
    }

    return dayNumber; // 변환된 요일 숫자 반환
}
```

면담이 불가능한 요일 입력 (월요일: 월, ..., 금요일: 금, 없으면 없음): 월



# 면담 불가능 날짜

```
int inputExcludeDate(int* excludeYear, int* excludeMonth, int* excludeDay) {
    printf("면담이 불가능한 날짜 입력 (예: 2024. 01. 01.): ");
    int inputResult = scanf("%d. %d. %d.", excludeYear, excludeMonth, excludeDay);

    // 입력값이 정수가 아니거나, 날짜가 잘못 입력된 경우에 오류 메시지 출력
    if (inputResult != 3 || *excludeYear <= 0 || *excludeMonth <= 0 || *excludeDay
<= 0 || *excludeMonth > 12 || *excludeDay > daysInMonth(*excludeYear,
*excludeMonth)) {
        printf("\"날짜 정보 입력 오류\"\n");
        // 실패한 입
        while (get
        printf("\n
        return 0; // 날짜 입력을 실패하면 다시 입력받는 함수로 반복
    }
    else {
        return 1;  // 성공
    }
}
```

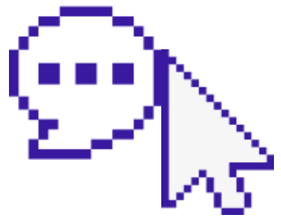
```
면담이 불가능한 날짜 입력 (예: 2024. 01. 01.): 2024.01
"날짜 정보 입력 오류"

면담이 불가능한 날짜 입력 (예: 2024. 01. 01.): 2024.01.02
```

# 면담 일정 출력

```
void generateClassMeetingSchedule(int selectedClass, int classes[3][10], int startYear, int startMonth,
int startDay, int meetingHour, int excludeWeekday, int excludeYear, int excludeMonth, int excludeDay,
Student* student) {
    MeetingSchedule schedule[10];

    // 학생 10명 배치
    int studentOrder[10];
    for (int i = 0; i < 10; ++i) {
        studentOrder[i] = i;
    }
    shuffleArray(studentOrder, 10); // 학생 배열 랜덤

    int currentYear = startYear;
    int currentMonth = startMonth;
    int currentDay = startDay;
    int currentHour = meetingHour;
```

# 면담 일정 출력

```
    for (int i = 0; i < 10; ++i) {
        int studentIndex = studentOrder[i];

// 면담 불가능한 요일 또는 날짜인 경우 다음 날로 조정
while (day_of_week(currentYear, currentMonth, currentDay) == excludeWeekday ||
day_of_week(currentYear, currentMonth, currentDay) == excludeWeekday1 ||
(currentYear == excludeYear
&& currentMonth == excludeMonth && currentDay == excludeDay) ||
day_of_week(currentYear, currentMonth, currentDay) == 0 || // 토요일
day_of_week(currentYear, currentMonth, currentDay) == 6) { // 일요일

currentDay += 1;
// 해당 월의 일수를 초과하면 한 달 추가하고 1일로 지정
if (currentDay > daysInMonth(currentYear, currentMonth)) {
    currentDay = 1;
    currentMonth += 1;
// 월 수가 12월을 초과하면 한 연도를 추가하고 1월로 지정
    if (currentMonth > 12) {
        currentMonth = 1;
        currentYear += 1;
```



# 면담 일정 출력

```
    // 면담일정 (결과) 출력
    printf("%d반 면담 일정\n", selectedClass);
    for (int i = 0; i
        for (int j =
            if (sched
                print
schedule[i].year, sch                                    i].year,
schedule[i].month, sc                                    ear,
schedule[i].month, sc                                    ear,
schedule[i].month, sc                                    ear,
schedule[i].month, sc                                    ear,
schedule[i].month, sc                                    ear,
schedule[i].month, sc
schedule[i].snum, stu
            }
        }

    }
}
```

```
1반 면담 일정
2024. 01. 03. 수 04시 1009 정다율 학생
2024. 01. 03. 수 05시 1003 성다솜 학생
2024. 01. 04. 목 04시 1008 이승찬 학생
2024. 01. 04. 목 05시 1005 정민건 학생
2024. 01. 05. 금 04시 1002 정태양 학생
2024. 01. 05. 금 05시 1006 최하윤 학생
2024. 01. 09. 화 04시 1001 홍윤하 학생
2024. 01. 09. 화 05시 1010 신소영 학생
2024. 01. 10. 수 04시 1007 고지혜 학생
2024. 01. 10. 수 05시 1004 윤다희 학생
```

# 성적 순 반 배정

```
void class1(int count, Student* student)
{
    printf("==방과후 자율학
    printf("==성적순으로 자

    sortList(student, coun
    //평균값을 내림차순으로
    for (int i = 0; i < co
    {
        if (i < 10)
        {
            strcpy(student
            if (i == 0)
                printf("==
            printf("%d %ls
        }
```

```
==방과후 자율학습배정 프로그램==
==성적순으로 자율학습 반을배정합니다==
==========A반==========
2008 박세은 3004 박지호 1010 신소영 3002 이민서 3003 이시우
1009 정다율 3009 정지수 1005 정민건 1006 최하윤 1008 이승찬
==========B반==========
1002 정태양 1004 윤다희 3008 김하린 2001 황서연 3010 이소율
1007 고지혜 2009 송아영 3007 서은우 2004 조예진 2005 김채아
==========C반==========
3001 김도윤 3006 박유준 2010 이재이 2007 조민규 1003 성다솜
2003 송예나 3005 김하은 2002 한소율 1001 홍윤하 2006 곽유빈
```

| | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 1001 | 홍윤하 | 01012345678 | 20100909 | b | 90 | 80 | 70 | 37.500000 | 사진부 | 양로원 | -858993460 | C반 |
| 1002 | 정태양 | 01012345678 | 20100909 | b | 90 | 80 | 70 | 78.000000 | 영화부 | 문화센터 | -858993460 | B반 |
| 1003 | 성다솜 | 01012345678 | 20100909 | b | 90 | 80 | 70 | 44.000000 | 영화부 | 고아원 | -858993460 | C반 |
| 1004 | 윤다희 | 01012345678 | 20100909 | c | 90 | 80 | 70 | 77.000000 | 독서부 | 고아원 | -858993460 | B반 |
| 1005 | 정민건 | 01012345678 | 20100909 | c | 90 | 80 | 70 | 81.000000 | 영화부 | 문화센터 | -858993460 | A반 |
| 1006 | 최하윤 | 01012345678 | 20100909 | a | 90 | 80 | 70 | 80.500000 | 영화부 | 문화센터 | -858993460 | A반 |
| 1007 | 고지혜 | 01012345678 | 20100909 | a | 90 | 80 | 70 | 66.699997 | 독서부 | 문화센터 | -858993460 | B반 |
| 1008 | 이승찬 | 01012345678 | 20100909 | a | 90 | 80 | 70 | 78.099998 | 독서부 | 고아원 | -858993460 | A반 |

# 자체 평가 의견

**한형빈 (팀장)**

이론적으로 배웠던 C언어 지식들을 활용해
실제로 활용될 수 있는 기능이 무엇일지
고민하는 시간이 되었다.

**추형욱**

숙련도가 부족해 많은 코드를 넣진 못했지만
팀원과의 소통을 배울 수 있는 좋은 기회였다.

**조민석**

파일 입출력이라는 함수의 사용법과 난이도를 알게 되었고
구조체를 정의하고 활용하는데 어려움을 느꼈고
페이지를 여러 개 나누어 작업 할 때 포인터의 중요점을
알게 되었다.

REVIEW

# THANK YOU!