C프로그래밍 및 실습

StudyPlan-30

진척 보고서 #3

제출일자: 2023.12.10
제출자명: 유소연
제출자학번: 234404

1. 프로젝트 목표

1) 배경 및 필요성

학창 시절 열심히 공부해 대학교에 왔지만, 대학 생활의 자유로움으로 인해 고등학교 때만큼 체계적인 공부 습관을 유지하지 못함. 학년이 올라감에 따라 학점도 중요해지는 시점에서 다시 공부 습관을 잡아야 할 필요가 있음. 이 문제를 해결하기 위해 자동으로 학습 플랜을 짜주는 프로그램이 필요함.

2) 프로젝트 목표

학습 계획을 세워 계획대로 공부에만 집중할 수 있도록 하는 것을 목표로 함.

3) 차별점

기존 프로그램들은 직접 계획을 세워야 함. 이는 계획 세우기에만 몰두하거나, 무리한 계획을 세울 문제가 있음. 우리는 계획을 세우고 달성하지 못한 계획과 부족했던 점을 입력해 놓으면 다시 추가적인 계획을 세우는 것에 기존 프로그램과 차별점이 있음.

2. 기능 계획

1) 사용자에게 작업 요청 받기

- 1. 30일용 계획 세우기, 2. 오늘의 D-day 확인하기, 3. 부족 개념 입력, 4. 부족한 개념 퀴즈, 5. 일정 미루기, 6. 종료

2) 30일용 계획 세우기 (① 입력 경우)

- 30일을 기준으로 각 과목의 회독, 문제 풀이&오답, 백지 테스트로 계획을 세우는 기능

3) 오늘의 D-day 확인하기 (② 입력 경우)

- 오늘의 D-day 를 입력하면 날짜에 맞는 계획을 출력해주는 기능

4) 부족 개념 입력하기 (③ 입력 경우)

- 부족했던 개념을 다시 상기할 수 있는 기능

5) 부족 개념 퀴즈 (④ 입력 경우)

- 부족했던 개념을 입력해 놓으면, 퀴즈 형식으로 출력된다.

6) 일정 미루기 (⑤ 입력 경우)

- 수행하지 못한 계획은 다음 날로 미루는 기능

- Day 30까지의 계획을 배열로 저장해 그 날 해야 하는 일의 수행 여부를 묻고,

수행하지 못했다면 배열에 저장된 Day 계획들을 한 칸씩 미룸.

3. 진척 상황

1) 기능 구현

(1) 사용자에게 작업 요청 받기

- 1. 30일용 계획 세우기, 2. 오늘의 D-day 확인하기, 3. 부족 개념 입력, 4. 부족한 개념 퀴즈, 5. 일정 미루기, 6. 종료

번호를 입력 / 메뉴 실행 코드 출력

- 번호를 입력하면 해당 기능이 실행되도록 출력

- 함수 , 반복문

- 코드 스크린샷

```c
void printMenu() {
    printf("₩n[메뉴]₩n");
    printf("1. 30일용 계획 세우기₩n");
    printf("2. 오늘의 D-day 계획 확인하기₩n");
    printf("3. 부족 개념 입력₩n");
    printf("4. 부족한 개념 확인하기₩n");
    printf("5. 오늘 일정을 미루시겠습니까? (Y/N)₩n");
    printf("6. 종료₩n");
    printf("메뉴를 선택하세요: ");
}
```

```c
// 메뉴 선택 및 처리
int menuChoice;
int dDay = 0;  // D-day 변수 초기화
do {
    printMenu();
    scanf_s("%d", &menuChoice);

    switch (menuChoice) {
    case 1:
        // 30일용 계획 세우기
        totalDays = MAX_DAYS;
        printf("₩n[스터디 계획]₩n");

        // 로테이션 계획 및 부족한 개념 출력
        for (int day = 1; day <= totalDays; ++day) {
            int subjectIndex = (day - 1) / 3 % subjectCount;

            printf("Day %d: %s ", day, subjects[subjectIndex].name);

            switch ((day - 1) % 3) {
            case 0:
                printf("%d회독₩n", ((day - 1) / 3) + 1);
                break;
            case 1:
                printf("문제풀이&오답 체크₩n");
                break;
            case 2:
                printf("백지 테스트₩n");
                break;
            }

            if (day % 3 == 0) {
                printf("₩n");
            }
        }
        break;
    case 2:
        // 오늘의 D-day 계획 확인하기
        printf("오늘의 D-day를 입력하세요: ");
        scanf_s("%d", &dDay);
        printDdayPlan(subjects, subjectCount, dDay);
        break;

    case 3:
        // 부족 개념 입력
        inputConcepts(subjects, subjectCount);
        break;
    case 4:
        // 부족한 개념 확인하기
        printf("과목을 선택하세요 (1부터 %d까지): ", subjectCount);
        int subjectIndex;
        scanf_s("%d", &subjectIndex);
        if (subjectIndex >= 1 && subjectIndex <= subjectCount) {
            printConcepts(&subjects[subjectIndex - 1]);
        }
        else {
            printf("잘못된 선택입니다.₩n");
        }
        break;
```

```c
    case 5:
        // 오늘 일정을 미루시겠습니까?
        printf("오늘 일정을 미루시겠습니까? (Y/N): ");
        char choice;
        scanf_s(" %c", &choice, 1);

        if (choice == 'Y' || choice == 'y') {
            // Day 30의 전체 계획을 하루씩 뒤로 밀기
            for (int i = MAX_DAYS - 1; i > 0; --i) {
                subjects[(i - 1) / 3 % subjectCount].concepts[i % MAX_CONCEPTS][0] =
                    subjects[(i - 1) / 3 % subjectCount].concepts[(i - 1) % MAX_CONCEPTS][0];
            }
            printf("Day 30의 계획이 하루씩 뒤로 밀렸습니다.₩n");

        }
        break;
    case 6:
        // 종료
        printf("프로그램을 종료합니다.₩n");
        break;
    default:
        printf("잘못된 메뉴 선택입니다. 다시 선택하세요.₩n");
    }

} while (menuChoice != 6);
```

(2) 30일용 계획 세우기

- 시험 보는 과목의 개수와 이름, 날짜를 입력 / Day30 까지의 계획을 출력

- 입력값을 넣으면 [회독, 문제풀이&오답, 백지테스트] 1세트가 과목 하나당 로테이션이 돌아가도록 Day30까지의 계획을 출력함.

- 반복문, 조건문

- 코드 스크린샷

```
void printEntirePlan(struct Subject* subjects, int subjectCount) {
    int totalDays = MAX_DAYS;
    printf("₩n[전체 스터디 계획]₩n");

    // 전체 30일의 계획을 로테이션 형식으로 출력
    for (int day = 1; day <= totalDays; ++day) {
        int subjectIndex = (day - 1) / 3 % subjectCount;

        printf("Day %d: %s ", day, subjects[subjectIndex].name);

        switch ((day - 1) % 3) {
        case 0:
            printf("%d회독₩n", ((day - 1) / 3) + 1);
            break;
        case 1:
            printf("문제풀이&오답 체크₩n");
            break;
        case 2:
            printf("백지 테스트₩n");
            break;
        }

        if (day % 3 == 0) {
            printf("₩n");
        }
    }
}
```

(3) 오늘의 D-day 확인하기

- 오늘의 D-day를 입력 / D-day에 맞는 할 일 출력

- 그 날 해야 하는 할 일을 확인할 수 있도록 출력함.

- 배열, 구조체, 조건문

- 코드 스크린샷

```
void printDdayPlan(struct Subject* subjects, int subjectCount, int dDay) {
    if (dDay < 1 || dDay > MAX_DAYS) {
        printf("D-day 값이 아닙니다.₩n");
        return;
    }

    printf("₩n[Day %d의 계획]₩n", dDay);

    int subjectIndex = (dDay - 1) / 3 % subjectCount;

    printf("    %s ", subjects[subjectIndex].name);

    switch ((dDay - 1) % 3) {
    case 0:
        printf("%d회독₩n", ((dDay - 1) / 3) + 1);
        break;
    case 1:
        printf("문제풀이&오답 체크₩n");
        break;
    case 2:
        printf("백지 테스트₩n");
        break;
    }

    printf("₩n");
}
```

(4) 부족 개념 입력

- 부족한 개념의 단어와 정의를 입력 / 출력 없이 입력값 저장

- 프로그램 초반에 입력한 과목들을 돌아가며 부족한 개념을 묻고, 입력 받은 개념들을 저장함.

- 함수, 반복문, 조건문, 포인터, 구조체

- 코드 스크린샷

```
void inputConcepts(struct Subject* subjects, int subjectCount) {
    for (int i = 0; i < subjectCount; ++i) {
        printf("%s 과목의 부족했던 개념을 입력해주세요 (입력을 마치려면 q를 입력하세요):₩n", subjects[i].name);

        for (int j = 0; j < MAX_CONCEPTS; ++j) {
            printf("단어: ");
            if (scanf_s("%s", subjects[i].concepts[j], 50) != 1 || subjects[i].concepts[j][0] == 'q') {
                // 'q'를 입력하면 입력 종료
                break;
            }

            printf("정의: ");
            scanf_s("%s", subjects[i].definitions[j], 100);
        }
    }
}
```

(5) , (6) 부족 개념 퀴즈, 일정 미루기

- 코드 오류로 재구현 예정

2) 테스트 결과

(1) 사용자에게 작업 요청 받기

- 함수와 do-while 문을 통해 메뉴 함수를 지정하고 해당 메뉴를 출력하는 기능 작성

- 메뉴 출력과 입력 번호 기능이 잘 수행되는지 테스트

- 테스트 결과 스크린샷(기능 스크린샷은 (2)~에서 설명)

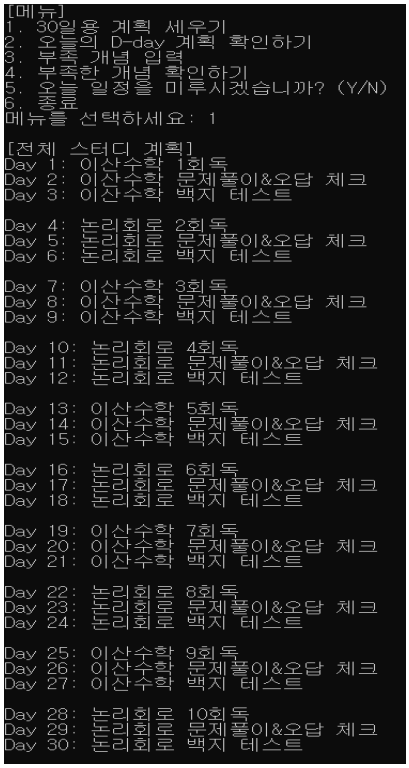
```
[메뉴]
1. 30일용 계획 세우기
2. 오늘의 D-day 계획 확인하기
3. 부족 개념 입력
4. 부족한 개념 확인하기
5. 오늘 일정을 미루시겠습니까? (Y/N)
6. 종료
메뉴를 선택하세요: 1
```

(2) 30일용 계획 세우기

– for 문을 통해 Day30 까지의 계획 출력 기능 작성

- 30일의 계획이 잘 출력이 되는지 테스트

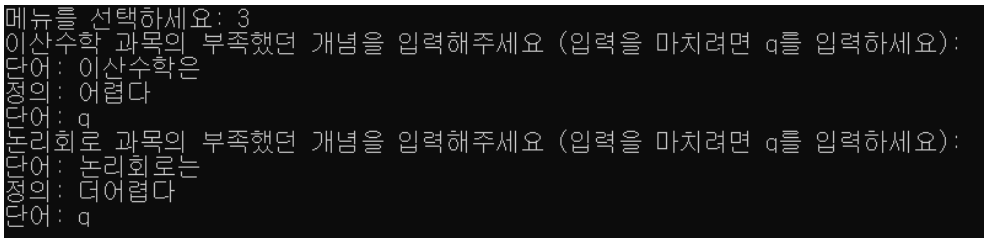
- 테스트 결과 스크린샷

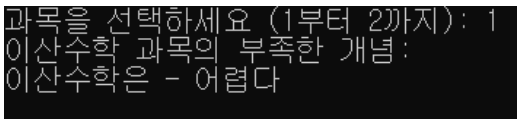


(3) 부족한 개념 입력

- 프로그램 초반에 입력한 과목들을 돌아가며 부족한 개념을 묻고, 입력 받은 개념들을 저장하는 기능 작성

- 테스트 결과 스크린샷





4. 계획 대비 변경 사항

1) 사용자에게 작업 요청 매뉴얼 제작

- 전 : 메뉴얼 없이 사용자가 직접 함수를 통해 출력 코드 작성

- 후 : 기능을 골라 출력할 수 있는 메뉴얼 제작

5. 프로젝트 일정

업무		11/3	11/20	12/10	.....
제안서 작성		완료			
기능1	-		완료		
기능2	-		완료		
기능3	-		완료		
기능4	-			---------->	
기능5				---------->	