

프로젝트형 실습 보고서

1. 서론

- 프로젝트 목적 및 배경
 - ① 기말 프로젝트 유형 파악과 보고서 작성법 숙지를 위한 실습 진행
 - ② 8주 차까지 학습한 함수 및 배열 등의 내용 복습 목적
- 목표 : 할 일을 추가하거나 삭제, 수정하고 목록을 출력할 수 있는 TODO 리스트 만들기

2. 요구사항

- 사용자 요구사항 : 사용자에게 목록 번호를 입력받아 할 일을 입력, 삭제 및 수정할 수 있고 입력받은 할 일의 목록을 출력할 수 있는 프로그램
- 기능 요구사항
 - ① 할 일 추가를 선택했을 경우, 사용자에게 할 일을 입력받고 저장
 - ② 할 일 삭제를 선택했을 경우, 사용자에게 할 일의 번호를 입력받고 해당 할 일 삭제
 - ③ 목록 보기를 선택했을 경우, 전체 할 일의 목록 출력
 - ④ 종료를 선택했을 경우, 프로그램 종료
 - ⑤ 할 일 수정을 선택했을 경우, 사용자에게 할 일의 번호와 수정할 할 일을 입력받고 해당 할 일 수정

3. 설계 및 구현

- ① 할 일 추가를 선택했을 경우, 사용자에게 할 일을 입력받고 저장

```
case 1: // 사용자가 1을 입력할 경우, 할 일 추가
    printf("할 일을 입력하세요 (공백 없이 입력하세요): ");

    // 입력받은 할 일을 배열 tasks에 저장
    scanf_s("%s", tasks[taskCount], (int)sizeof(tasks[taskCount]));
    printf("할 일 \"%s\"가 저장되었습니다\n\n", tasks[taskCount]);
    taskCount++; // 할 일의 개수 저장하는 변수의 값 증가
    break;
```

• 입력

- tasks = 할 일을 저장하는 2차원 배열
- taskCount = 할 일의 개수를 저장하는 변수

• 결과

- tasks에 할 일이 추가되고, 현재 할 일의 개수를 나타내는 taskCount 값 증가

• 설명

- 사용자가 1을 입력한 경우, 프로그램은 할 일 추가 기능을 수행한다.
- 사용자에게 저장할 할 일의 내용을 입력받고, tasks에 저장한다.
- 할 일의 개수 증가를 나타내기 위해 taskCount의 값을 1 증가시킨다.

② 할 일 삭제를 선택했을 경우, 사용자에게 할 일의 번호를 입력받고 해당 할 일 삭제

```
case 2: // 사용자가 2를 입력할 경우, 할 일 삭제
    printf("삭제할 할 일의 번호를 입력해주세요. (1부터 시작):");

    // 삭제할 할 일의 번호를 사용자에게 입력받아 delIndex 변수에 저장
    scanf_s("%d", &delIndex);
    if (delIndex > taskCount || delIndex <= 0) {
        printf("삭제 범위가 벗어났습니다.\n");
    }
    else {
        printf("%d. %s : 할 일을 삭제합니다.\n", delIndex, tasks[delIndex - 1]);

        // 문자열 복사 함수를 이용해 입력받은 번호의 할 일에 해당하는 인덱스의 내용 삭제
        strcpy_s(tasks[delIndex - 1], sizeof(tasks[delIndex - 1]), "");

        // 해당 인덱스의 뒤에 있는 인덱스의 할 일 가져오기
        for (int i = delIndex; i < taskCount + 1; i++) {
            strcpy_s(tasks[i - 1], sizeof(tasks[i]), tasks[i]);
        }
        taskCount -= 1; // 할 일의 개수 저장하는 변수의 값 감소
    }
    break;
```

- 입력

- tasks = 할 일을 저장하는 2차원 배열
- taskCount = 할 일의 개수를 저장하는 변수
- delIndex = 삭제할 할 일의 번호를 입력받고 저장하는 변수

- 결과

- 사용자가 입력한 번호의 할 일이 삭제되고, 현재 할 일의 개수를 나타내는 taskCount 값 감소

- 설명

- 사용자가 2를 입력한 경우, 프로그램은 할 일 삭제 기능을 수행한다.
- 사용자에게 삭제할 할 일의 번호를 입력받고, tasks 배열의 해당 인덱스(입력받은 번호 - 1)의 내용을 삭제한다.
- 삭제된 인덱스의 뒤에 있는 인덱스의 할 일을 앞으로 가져와 빈자리를 채운다.
- 할 일의 개수 감소를 나타내기 위해 taskCount의 값을 1 감소시킨다.

③ 목록 보기를 선택했을 경우, 전체 할 일의 목록 출력

```

case 3: // 사용자가 3을 입력할 경우, 목록 보기
    printf("할 일 목록\n");

    // 저장된 할 일의 개수인 taskCount 만큼 반복
    for (int i = 0; i < taskCount; i++) {
        printf("%d. %s\n", i + 1, tasks[i]); // 저장된 할 일의 번호와 내용 출력
    }
    printf("\n");
    break;

```

• 입력

- tasks = 할 일을 저장하는 2차원 배열
- taskCount = 할 일의 개수를 저장하는 변수

• 결과

- 사용자가 추가하거나 삭제, 수정한 할 일들의 현재 결과를 출력한다.

• 설명

- 사용자가 3을 입력한 경우, 프로그램은 할 일 목록 출력 기능을 수행한다.
- 저장된 taskCount 변수의 값만큼 반복하여 tasks 배열의 값들을 인덱스 + 1과 함께 출력한다.

④ 종료를 선택했을 경우, 프로그램 종료

```

case 4: // 사용자가 4를 입력할 경우, 프로그램 종료
    terminate = 1; // 프로그램 종료를 위한 변수 값 변경
    break;

```

+) 프로그램 종료 기능을 수행해주는 코드 블록 스크린샷 추가

```

if (terminate == 1) { // terminate 값이 1이 되면 프로그램 종료
    printf("종료를 선택하셨습니다. 프로그램을 종료합니다.\n");
    break;
}

```

• 입력

- terminate = 프로그램 종료를 위한 변수

• 결과

- TODO 리스트 프로그램 종료

• 설명

- 사용자가 4를 입력한 경우, 프로그램은 종료 기능을 수행한다.
- terminate 값을 1로 변경해 프로그램을 종료시켜준다.

⑤ 할 일 수정을 선택했을 경우, 사용자에게 할 일의 번호와 수정할 할 일을 입력받고 해당 할 일 수정

```
case 5: // 사용자가 5를 입력할 경우, 할 일 수정
    printf("수정할 할 일의 번호를 입력해주세요. (1부터 시작):");
    // 수정할 할 일의 번호를 사용자에게 입력받아 changeIndex 변수에 저장
    scanf_s("%d", &changeIndex);

    // 사용자가 저장된 할 일의 개수보다 큰 번호나 0 이하의 번호를 입력할 경우
    if (changeIndex > taskCount || changeIndex <= 0) {
        printf("삭제 범위가 벗어났습니다.\n");
    }
    else {
        printf("%d. %s : 할 일을 수정합니다.\n", changeIndex, tasks[changeIndex - 1]);
        // 입력받은 번호의 할 일에 해당하는 인덱스의 내용 삭제
        strcpy_s(tasks[changeIndex - 1], sizeof(tasks[changeIndex - 1]), "");
        printf("새로운 할 일을 입력하세요 (공백 없이 입력하세요): ");
        // 새롭게 입력받은 할 일을 tasks[changeIndex - 1]에 저장
        scanf_s("%s", tasks[changeIndex - 1], (int)sizeof(tasks[changeIndex - 1]));
        printf("할 일 \"%d. %s\"로 수정되었습니다.\n\n", changeIndex, tasks[changeIndex - 1]);
    }
    break;
```

• 입력

- tasks = 할 일을 저장하는 2차원 배열
- taskCount = 할 일의 개수를 저장하는 변수
- changeIndex = 수정할 할 일의 번호를 입력받고 저장하는 변수

• 결과

- 사용자가 입력한 번호의 할 일이 새로 입력한 내용으로 수정된다.

• 설명

- 사용자가 5를 입력한 경우, 프로그램은 할 일 수정 기능을 수행한다.
- 사용자에게 수정할 할 일의 번호를 입력받고, tasks 배열의 해당 인덱스(입력받은 번호 - 1)의 내용을 삭제한다.
- 삭제된 인덱스의 자리에 새로운 내용을 저장하기 위해 사용자에게 새로 저장할 할 일을 입력받는다.
- 수정된 할 일의 번호와 내용을 출력한다.

4. 테스트

- 기능별 테스트 결과

① 할 일 추가를 선택했을 경우, 사용자에게 할 일을 입력받고 저장

```
1
할 일을 입력하세요 (공백 없이 입력하세요): 밥먹기
할 일 밥먹기가 저장되었습니다
```

```
1
할 일을 입력하세요 (공백 없이 입력하세요): 공부하기
할 일 공부하기가 저장되었습니다
```

② 할 일 삭제를 선택했을 경우, 사용자에게 할 일의 번호를 입력받고 해당 할 일 삭제

```
2
삭제할 할 일의 번호를 입력해주세요. (1부터 시작):1
1. 밥먹기 : 할 일을 삭제합니다.
```

③ 목록 보기를 선택했을 경우, 전체 할 일의 목록 출력

```
3
할 일 목록
1. 공부하기
```

④ 종료를 선택했을 경우, 프로그램 종료

```
4
종료를 선택하셨습니다. 프로그램을 종료합니다.
```

⑤ 할 일 수정을 선택했을 경우, 사용자에게 할 일의 번호와 수정할 할 일을 입력받고 해당 할 일 수정

```
5
수정할 할 일의 번호를 입력해주세요. (1부터 시작):1
1. 공부하기 : 할 일을 수정합니다.
새로운 할 일을 입력하세요 (공백 없이 입력하세요): 일하기
할 일 1. 일하기로 수정되었습니다
```

- 최종 테스트 결과

```

1
할 일을 입력하세요 (공백 없이 입력하세요): 공부하기
할 일 공부하기가 저장되었습니다

-----
메뉴를 입력해주세요.
1. 할 일 추가
2. 할 일 삭제
3. 목록 보기
4. 종료
5. 할 일 수정
현재 할 일 수 = 1

-----
3
할 일 목록
1. 공부하기

-----
메뉴를 입력해주세요.
1. 할 일 추가
2. 할 일 삭제
3. 목록 보기
4. 종료
5. 할 일 수정
현재 할 일 수 = 1

-----
5
수정할 할 일의 번호를 입력해주세요. (1부터 시작):1
1. 공부하기 : 할 일을 수정합니다.
새로운 할 일을 입력하세요 (공백 없이 입력하세요): 일하기
할 일 1. 일하기로 수정되었습니다

-----
메뉴를 입력해주세요.
1. 할 일 추가
2. 할 일 삭제
3. 목록 보기
4. 종료
5. 할 일 수정
현재 할 일 수 = 1

-----
2
삭제할 할 일의 번호를 입력해주세요. (1부터 시작):1
1. 일하기 : 할 일을 삭제합니다.

-----
메뉴를 입력해주세요.
1. 할 일 추가
2. 할 일 삭제
3. 목록 보기
4. 종료
5. 할 일 수정
현재 할 일 수 = 0

-----
4
종료를 선택하셨습니다. 프로그램을 종료합니다.

```

5. 결과 및 결론

- 프로젝트 결과

- 할 일을 추가, 삭제 및 수정할 수 있는 TODO 리스트 프로그램을 만들었다.
- 프로그램에는 저장된 할 일의 목록을 출력하는 기능과 프로그램을 종료하는 기능도 포함되어 있다.

- 결론 및 느낀 점 : C 프로그램의 기본 개념을 학습한 후에 바로 응용으로 프로젝트를 진행해서 진도가 너무 빠른 것 같다는 생각이 든다.