

1. 서론

- 1) 프로젝트 목적 및 배경 : 7주차까지 배운 내용에 대한 실습을 위해 진행
- 2) 목표 : TODO 리스트 만들기

2. 요구사항

- 1) 사용자 요구사항 : 사용자가 할 일을 추가, 삭제, 변경, 출력 등을 할 수 있는 프로그램
- 2) 기능 요구사항
 - ① 사용자에게 작업 요청 받기
 - 1. 할 일 추가 2. 할 일 삭제 3. 목록 보기 4. 종료 5. 할 일 수정
 - ② 요청 받은 작업에 따라 아래 기능 수행
 - 1) 할 일 추가를 입력했을 경우, 사용자에게 할 일을 입력 받고 저장
 - 2) 할 일 삭제를 입력했을 경우, 인덱스를 입력 받고 해당 할 일 삭제
 - 3) 목록 보기를 입력했을 경우, 전체 할 일 목록을 보여주기
 - 4) 종료를 입력했을 경우, 프로그램 종료
 - 5) 할 일 수정을 입력했을 경우, 인덱스와 할 일 (문자열)을 입력 받고, 해당 인덱스의 할 일 변경
 - ③ 할 일이 10개로 다 찬 경우는 할 일이 다 찼다고 출력하고 프로그램 종료

3. 설계 및 구현

- 1) 기능 별 구현 사항 :

- ① 할 일 생성하는 함수

```
// 할일 생성하는 함수
void addTask(char tasks[][CHAR_NUM], int taskCount) {
    printf("할 일을 입력하세요 (공백 없이 입력하세요): ");
    scanf_s("%s", tasks[taskCount], CHAR_NUM);
    printf("할 일 \"%s\"가 저장되었습니다\n", tasks[taskCount]);
}
```

- tasks = 할 일 목록 저장 2차원 배열
- taskCount = 현재 작업 수
- 반환값 = 없음
- 결과 = 사용자에게 입력받은 할 일을 2차원 배열에 저장후 출력
- 설명 = 사용자에게 할 일을 입력받고, 그값을 저장하고 사용자에게 저장되었다는 문구 출력

② 할 일 삭제하는 함수

```
// 할일 삭제하는 함수
void delTask(int delIndex, int taskCount) {
    printf("삭제할 할 일의 번호를 입력해주세요. (1부터 시작):");
    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]);

        // 배열간 대입 (=배열에 문자 배열인 문자열의 대입) 이 불가능하기 때문에
        // 문자열 복사 함수로 삭제
        strcpy_s(tasks[delIndex - 1], sizeof(tasks[delIndex - 1]), "");

        // 특정 인덱스의 할 일 삭제 후 뒤에 있는 할 일 앞으로 옮기기
        for (int i = delIndex; i < taskCount + 1; i++) {
            strcpy_s(tasks[i - 1], sizeof(tasks[i]), tasks[i]);
        }
    }
}
```

- delIndex = 할 일이 저장되어있는 순번
- taskCount = 현재 작업 수
- 반환값 = 없음
- 결과 = 사용자에게 입력받은 삭제할 할 일 의 번호와 내용을 삭제함
- 설명 = 1. 사용자에게 삭제할 할 일 을 입력받음(이때 입력받은 삭제할 할 일의 번호가 할 일 목록의 최댓값을 넘어가면 에러가 발생하게 함)
2. 삭제할 할 일의 내용을 삭제하고 빈 공간을 앞으로 당김

③ 할 일 목록을 불러오는 함수

```
// 할일 목록 불러오는 함수
void printTask(int taskCount) {
    printf("할 일 목록\n");
    for (int i = 0; i < taskCount; i++) {
        printf("%d. %s\n", i + 1, tasks[i]);
    }
    printf("\n");
}
```

- taskCount = 현재 작업 수
- 반환값 = 없음
- 결과 = 현재 작업목록을 출력해줌
- 설명 = 반복문을 활용해 현재 작업수만큼 반복하며 할 일을 출력해줌

④ 할 일 수정하는 함수

```
// 할 일 수정하는 함수
void modifyTask(int delIndex, char tasks[][CHAR_NUM], int taskCount) {
    printf("수정할 할 일의 번호를 입력해주세요. (1부터 시작):");
    scanf_s("%d", &delIndex);
    if (delIndex > taskCount || delIndex <= 0) {
        printf("번호의 범위가 벗어났습니다.\n");
    }
    else { // 삭제하는 함수 변형해서 수정하는 함수로 바꿈
        printf("새로운 일을 입력하세요 (공백 없이 입력하세요): ");
        scanf_s("%s", tasks[taskCount], CHAR_NUM);
        strcpy_s(tasks[delIndex - 1], sizeof(tasks[delIndex - 1]), tasks[taskCount]);
        printf("새로운 일 \"%s\"가 저장되었습니다.\n", tasks[taskCount]);
    }
}
```

- delIndex = 할 일이 저장되어있는 순번
- tasks = 할 일 목록 저장 2차원 배열
- taskCount = 현재 작업 수
- 반환값 = 없음
- 결과 = 수정할 할 일의 번호를 입력받고 그 번호에 맞는 할 일을 새로 입력한 할 일로 수정해줌
- 설명 = 1. 수정 할 할 일의 번호를 입력받음(이때 입력받은 삭제할 할 일의 번호가 할 일 목록의 최대값을 넘어가면 에러가 발생하게 함)
2. 새로운 할 일을 입력받고 2차원 배열에 저장하고 그 값을 기존 할 일에 덮어쓰기 함

⑤ 할 일이 10개가 되면 프로그램을 종료하는 코드

```
// taskCount 가 10개가 되면 종료하고 아니면 밑의 main함수를 진행
if (taskCount == 10) {
    printf("-----\n");
    printf("할일이 다 찼습니다. 프로그램을 종료합니다.\n");
    printf("-----\n");
    break;
}
else {
```

- taskCount = 현재 작업 수
- 반환값 = 없음
- 결과 = 할 일이 10개가 되면 할 일이 다 찼다는 출력과 함께 프로그램 종료
- 설명 = 조건문으로 현재 작업 수가 10개가 되면 프로그램을 종료하고 그렇지 않을 경우 else를 통해 밑의 main함수 진행

4. 테스트

1) 기능 별 테스트 결과 :

① 할 일 생성하는 함수

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

② 할 일 삭제하는 함수

1) 제대로 입력했을때

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

2) 범위가 벗어났을 때 → 오류 발생

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

③ 할 일 목록을 불러오는 함수

```
3
할 일 목록
1. 공부하기
2. 밥먹기
3. 과제하기
```

④ 할 일 수정 하기

```
5
수정할 할 일의 번호를 입력해주세요. (1부터 시작):1
새로운 일을 입력하세요 (공백 없이 입력하세요): C언어프로그램실습하기
새로운 일 C언어프로그램실습하기가 저장되었습니다
-----
메뉴를 입력해주세요.
1. 할 일 추가
2. 할 일 삭제
3. 목록 보기
4. 종료
5. 할 일 수정
현재 할 일 수 = 3
-----
3
할 일 목록
1. C언어프로그램실습하기
2. 밥먹기
3. 과제하기
```

⑤ 할 일이 10개가 되면 프로그램을 종료하는 코드

```
0. 할 일 추가
현재 할 일 수 = 9
-----
1
할 일을 입력하세요 (공백 없이 입력하세요): 1
할 일 1가 저장되었습니다
-----
할일이 다 찼습니다. 프로그램을 종료합니다.
-----
```

2) 최종 테스트 스크린샷

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

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. 밥먹기 : 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

- 1) 프로젝트 결과 : 다른 추가 사항은 문제없이 작동하는데
할 일 삭제하는 함수에서 에러가 발생했는데 수정에 실패했다.
- 2) 느낀점 : C언어에 대해 실제 실습을 하면서 어떻게 작동하는지 알 수 있었다.