

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 입력, 삭제, 출력할 수 있는 프로그램
2. 기능 요구사항
 1. 할 일 추가(최대 10개까지 + 10개를 넘을 경우 프로그램 종료)
 2. 할 일 삭제 + 삭제하고 남은 칸 제거
 3. 목록 보기
 4. 종료
 5. 할 일 수정

3. 설계 및 구현

1. 기능 별 구현 사항:

1. 할 일 추가

```
case 1:
    if (taskCount >= 10) { // 추가 조건인 할 일:
        terminate = 2;
        break;
    }
    addTask(taskCount, tasks);
    taskCount++; // 할 일 의 갯수 1개 증가
    break; // switch-case 문 종료

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

2. 할 일 삭제

```
case 2:
    printf("삭제할 일의 번호를 입력해주세요. (1부터 시작) :");
    scanf_s("%d", &delIndex);
    if (delIndex > taskCount || delIndex <= 0) { // 입력받은 값 > 할 일 개수
        printf("삭제 범위가 벗어났습니다.\n");
    }
    else {
        delTask(delIndex, taskCount, tasks);
        taskCount -= 1;
    }
    break; // switch-case 문 종료
```

```
void delTask(int delIndex, int taskCount, char tasks[][CHAR_NUM]) {
    printf("%d. %s : 할 일을 삭제합니다.\n", delIndex, tasks[delIndex - 1]);
    strcpy_s(tasks[delIndex - 1], sizeof(tasks[delIndex - 1]), "");
    //입력받은 번호의 값을 빈 값으로 변경
    for (int i = delIndex; i < taskCount + 1; i++) {
        strcpy_s(tasks[i - 1], sizeof(tasks[i]), tasks[i]);
    } // 목록중 한개를 제거하기 위해 그 뒤의 값을 한칸씩 앞의 인덱스에 복사하여 빈 칸 만들기
}
```

3. 목록 출력

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

4. 종료

```
if (terminate == 1) {
    printf("종료를 선택하셨습니다. 프로그램을 종료합니다.\n");
}
else if (terminate == 2) {
    printf("할 일이 10개로 최대치이므로 프로그램을 종료합니다.");
}
if (terminate != 0) {
    break; //while문 종료
}
```

5. 할 일 수정

```
case 5:
    for (int i = 0; i < taskCount; i++) { //어떤 번호를 수정할지 입력받기 전 한번
        printf("%d. %s\n", i + 1, tasks[i]);
    }
    printf("수정할 목록의 번호를 입력해주세요(숫자만 입력) :");
    scanf_s("%d", &changeIndex);
    ch = getchar(); //버퍼 제거
    printf("목록의 수정될 이름을 입력하세요:");
    scanf_s("%s", tasks[changeIndex - 1], (int)sizeof(tasks[changeIndex - 1]));
    //입력받은 값을 이전에 선택한 번호의 인덱스에 저장
    break; // switch-case 문 종료
```

4. 테스트

1.기능

별

테스트

결과:

TODO 리스트 시작!

메뉴를 입력해주세요.

1. 할 일 추가
2. 할 일 삭제
3. 목록보기
4. 종료
5. 할 일 수정

현재 할 일 수 = 0

1

할 일을 입력하세요 (공백 없이 입력하세요) : abc

할 일abc가 저장되었습니다

메뉴를 입력해주세요.

1. 할 일 추가
2. 할 일 삭제
3. 목록보기
4. 종료
5. 할 일 수정

현재 할 일 수 = 1

3

할 일 목록

1. abc

```

3
할 일 목록
1. abc
2. def

-----
메뉴를 입력해주세요.
1. 할 일 추가
2. 할 일 삭제
3. 목록보기
4. 종료
5. 할 일 수정
현재 할 일 수 = 2
-----
2
삭제할 일의 번호를 입력해주세요. (1부터 시작) :1
1. abc : 할 일을 삭제합니다.

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

```

```

5
1. def
수정할 목록의 번호를 입력해주세요(숫자만 입력) :1
목록의 수정될 이름을 입력하세요:abcd

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

```

5. 결과 및 결론

1. 프로젝트 결과: todo 관리프로그램을 만들었다.
2. 느낀 점: 중간 단계를 넘기고 바로 심화단계로 간 느낌을 받았습니다. 그나마 어려운 부분이 작성된 코드를 미리 받아 할 만 했던 것 같습니다.