



# TODO\_MANAGER.CPP

## 프로젝트 요약

2023-10-23	프로젝트 이름	작성자
전남대학교	Todo_manager.cpp	한동호

## 상황 요약

10 월 23 일 교수님께서 7 주차까지 배운 내용에 대한 실습을 위해  
할일 목록 관리 프로그램 관련 실습을 진행, 이에 따른 보고서 작성.  
목표 - : TODO 리스트 만들기

## 요구사항(기능 계획)

**사용자 요구사항:** 사용자가 할 일을 입력, 삭제, 출력할 수 있는 프로그램

### 기능 요구 사항

#### ① 사용자에게 작업 요청 받기

- 1. 할 일 추가, 2. 할 일 삭제, 3. 목록 보기, 4. 종료, 5. 할일 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 10 개로 다 찬 경우는 할 일이 다 찼다고 출력하고 프로그램 종료

**제약 조건 :** 할 일 목록은 2 차원 배열 (10 x 100) = 100 개의 문자를 저장할 수 있는 문자열 10 개  
저장

## 요구사항(함수 계획)

할일 추가기능 => `void addTask(char task[]`

할일 삭제기능 => `void delTask(int delIndex, int taskCount)`

할일 목록 출력 기능 => `void printTask(int taskCount)`

## 설계 및 구현( 함수)

### <함수계획>

할일 추가기능 => `void addTask(char task[]`

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

할일 삭제기능 => `void delTask(int delIndex, int taskCount)`

```
void delTask(int delIndex, int taskCount) {  
  
    printf("%d. %s : 할 일을 삭제합니다.\n", delIndex, 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]);  
    }  
}
```

할일 목록 출력 기능 => `void printTask(int taskCount)`

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

## <기능요구사항>

### ① 사용자에게 작업 요청 받기

- 1. 할 일 추가, 2. 할 일 삭제, 3. 목록 보기, 4. 종료, 5. 할 일 수정

`int choice = -1; // 사용자 입력 메뉴를 저장하기 위한 변수`

```
// 사용자에게 메뉴를 출력하고, 메뉴를 입력받기
printf("-----\n");
printf("메뉴를 입력해주세요.\n");
printf("1. 할 일 추가\n2. 할 일 삭제\n3. 목록 보기\n4. 종료\n5. 할 일 수정\n");
printf("현재 할 일 수 = %d\n", taskCount);
printf("-----\n");
scanf_s("%d", &choice);
```

### 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

```
// 입력에 따른 기능 수행
switch (choice) {
case 1:
    addTask(tasks[taskCount]);
    taskCount++;
    break;
case 2:
    printf("삭제할 할 일의 번호를 입력해주세요. (1부터 시작):");
    scanf_s("%d", &delIndex);
    if (delIndex > taskCount || delIndex <= 0) {
        printf("삭제 범위가 벗어났습니다.\n");
    }
    else {
        delTask(delIndex, taskCount);
        taskCount -= 1;
    }
    break;
case 3:
    printf("할 일 목록\n");
    printTask(taskCount);
    break;
case 4:
    terminate = 1;
```

```

        break;
case 5:
    printf("수정할 할 일의 번호를 입력해주세요. (1부터 시작): ");
    scanf_s("%d", &modifyIndex);
    ch = getchar();
    printf("새로운 할 일을 입력해주세요");
    scanf_s("%s", tasks[modifyIndex - 1], (int)sizeof(tasks[modifyIndex - 1]));
    printf("새로운 할 일이 추가되었습니다: %d. %s\n", modifyIndex, tasks[modifyIndex - 1]);
    break;
default:
    printf("잘못된 선택입니다. 다시 선택하세요.\n");
}

if (terminate == 1) {
    printf("종료를 선택하셨습니다. 프로그램을 종료합니다.\n");
    break;
}

```

### ③ 할 일이 10 개로 다 찬 경우는 할 일이 다 찼다고 출력하고 프로그램 종료

```

if (taskCount == 10) {
    printf("할 일이 %d개로 다 찼습니다.", taskCount);
    break;
}
}

```

## 테스트 (기능별)

### 기능 요구 사항

#### ① 사용자에게 작업 요청 받기

- 1. 할 일 추가, 2. 할 일 삭제, 3. 목록 보기, 4. 종료, 5. 할 일 수정

```

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

```

## ② 요청 받은 작업에 따라 아래 기능 수행

① 할 일 추가를 입력했을 경우, 사용자에게 할 일을 입력 받고 저장

```
-----  
1  
할 일을 입력하세요 (공백 없이 입력하세요): 동뎡기  
할 일 동뎡기가 저장되었습니다
```

② 할 일 삭제를 입력했을 경우, 인덱스를 입력 받고 해당 할 일 삭제

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

③ 목록 보기를 입력했을 경우, 전체 할 일 목록을 보여주기

```
3  
할 일 목록  
1. 동싸기
```

④ 종료를 입력했을 경우, 프로그램 종료

```
-----  
4  
종료를 선택하셨습니다. 프로그램을 종료합니다.
```

⑤ 할 일 수정을 입력했을 경우, 인덱스와 할 일 (문자열)을 입력 받고, 해당 인덱스의 할 일 변경

```
5  
수정할 할 일의 번호를 입력해주세요. (1부터 시작): 1  
새로운 할 일을 입력해주세요 이뎡기  
새로운 할 일이 추가되었습니다: 1. 이뎡기
```

③ 할 일이 10 개로 다 찬 경우는 할 일이 다 찼다고 출력하고 프로그램 종료

```
-----  
1q  
할 일을 입력하세요 (공백 없이 입력하세요): 할 일 q가 저장되었습니다  
  
할 일이 10개로 다 찼습니다.
```

```
-----
메뉴를 입력해주세요.
1. 할 일 추가
2. 할 일 삭제
3. 목록 보기
4. 종료
5. 할 일 수정
현재 할 일 수 = 9
-----
1q
할 일을 입력하세요 (공백 없이 입력하세요): 할 일 q가 저장되었습니다
할 일이 10개로 다 찹습니다.
```

## 결과 및 결론

### 프로젝트 결과

⇒ 투두 관리프로그램을 만들었다.

### 느낀 점

⇒ 실제로 (2.할일 삭제) 옵션이 있었으면 좋겠다.

