

## 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 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 인덱스  
의 할 일 변경 • 주의: 입력 받는 인덱스에 -1 한 것이 실제 배열의 인덱스가 됨
  - ③ 할 일이 10개로 다 찬 경우는 할 일이 다 찼다고 출력하고 프로그램 종료

## 3. 설계 및 구현

1. 기능 별 구현사항 :

```

while (1) {
    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);
}

```

#### 1. 입력 :

1. printf : 할 일 출력
2. scanf : 할 일을 입력받음.
3. choice : 사용자 입력 메뉴를 저장하기 위한 변수
4. taskCount : 현재 작업수

#### 2. 결과 :

1. scanf에 할일을 입력받음
2. choice에 번호를 입력

```

int terminate = 0; // 종료를 위한 flag
int delIndex = -1; // 할 일 삭제를 위한 index 저장 변수
int changeIndex = -1; // 할 일 수정을 위한 index 저장 변수
char ch; // 할 일 수정시 버퍼를 받기 위한 문자 변수

```

#### 1. 입력 :

주석으로 서술.

```

// 입력에 따른 기능 수행
switch (choice) {
case 1:
    printf("할 일을 입력하세요 (공백 없이 입력하세요): ");
    scanf_s("%s", tasks[taskCount], (int)sizeof(tasks[taskCount]));
    printf("할 일 \"%s\"가 저장되었습니다.\n\n", tasks[taskCount]);
    taskCount++;
    break;
case 2:
    // 할 일 삭제하는 코드 블록
    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]);
        }
        taskCount -= 1;
    }
    break;
case 3:
    printf("할 일 목록\n");
    for (int i = 0; i < taskCount; i++) {
        printf("%d. %s\n", i + 1, tasks[i]);
    }
    printf("\n");
    break;
case 4:
    terminate = 1;
    break;
case 5:
    printf("수정할 할 일의 번호를 입력해주세요. (1부터 시작): ");
    scanf_s("%d", &changeIndex);
    if (changeIndex > taskCount || changeIndex <= 0) {
        printf("수정할 할 일이 없습니다.\n");
    }
    else {
        printf("현재 할 일 \"%s\"를 어떻게 수정하시겠습니까? ", tasks[changeIndex - 1]);
        scanf_s("%99s", tasks[changeIndex - 1], (int)sizeof(tasks[changeIndex - 1])); // %99s로 공백처리
        printf("할 일이 수정되었습니다.\n");
    }
}
break;

```

## 1. 입력 :

1. switch 조건문을 이용해 사용자에게 입력받은 choice에 따라 할 일을 입력받는다.

1. case 1 : print문과 scanf문으로 할 일을 입력받아 추가한다.

2. case 2 : print문과 scanf문으로 case1에서 받은 할 일을 지정받은 다음 if문으

로 조건을 따진뒤에 삭제한다 또 for문을 이용하여 삭제 한뒤 삭제

한 번호를 제외한 것을 앞으로 옮긴다 후에 taskCount -1

3. case 3 : for문을 이용해 tasks 배열을 출력함으로써 할 일 목록을 출력
4. case 4 : terminate 변수를 1로 설정해서 프로그램 종료 이 코드는 밑에 서술
5. case 5 : 수정할번호를 changeIndex 변수에 입력받고 if 로 이용해서 있는 번호인지 구별 tasks배열에 수정할 문구를 입력받고 수정

## 2. 결과 :

1. 입력한 번호에 따라 switch문 출력

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

// TODO: 할 일이 다 찼는지 체크하는 코드 작성
if (taskCount == 10) {
    printf("할 일이 다했습니다 프로그램을 종료합니다...\n");
    break;
}
```

## 1. 입력 :

1. if

1. if문을 이용해서 terminate(위에서 서술) 변수가 1이면 종료
2. if문을 이용해서 taskCount(위에서 서술) 변수가 10이면 종료

## 2. 결과 :

프로그램종료

```

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);
}

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]);
    }
}

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

```

#### 1. 입력 :

1. addTask : 할 일을 추가하는 함수
2. task[] : 할 일을 모아두는 배열
3. delTask : 할 일을 삭제하는 함수
4. delIndex : 할 일을 삭제 할 인덱스
5. taskCount : 할 일 개수 (위에 서술)
6. printfTask : 할 일 목록 보여주는 함수

#### 2. 결과 :

위에 서술

#### 3. 반환값

전달값만 존재 반환값 없음

#### 4. 설명

1. addTask : 사용자에게 추가할 할 일을 입력받고 task[] 배열에 추가후 출력
2. delTask : 삭제 할 일을 입력받은 뒤 삭제 후 뒤에 있는 할 일 앞으로 옮기는 것
3. printTask : 할 일의 목록을 for문으로 task 배열 전체 출력

#### 4. 테스트

##### 1. 기능 별 테스트 결과 :

###### 1. 할 일 추가하기

```
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
-----
1
할 일을 입력하세요 (공백 없이 입력하세요): 이재현
할 일 이재현가 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
-----
```

###### 2. 할 일 삭제하기

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

###### 3. 목록 보기

```
3
할 일 목록
1. 이재현
```

###### 4. 종료

```
4
종료를 선택하셨습니다. 프로그램을 종료합니다.

C:\Users\sozed\source\repos\week8\64\Debug\week8.exe(프로세스 15896개)이(가) 종료되었습니다(코드: 0개).
이 창을 닫으려면 아무 키나 누르세요...
```

## 5. 할 일 수정

```
5
수정할 할 일의 번호를 입력해주세요. (1부터 시작): 1
현재 할 일 "dlwogus"를 어떻게 수정하시겠습니까? 이재현
할 일이 수정되었습니다.
```

## 5. 결과 및 결론

1. 프로젝트 결과 : 투두 관리프로그램을 만들었다.
2. 느낀 점 : 원래의 코드를 함수로 다시 바꾸는 과정이 쉽지않아서 연습의 필요성을 느껴 열심히 연습중이다.