

## 1.서론

1. 프로젝트 목적 및 배경: 7주차까지 배운 내용을 특정한 목적을 가진 프로그램을 만듦으로써 배운 내용이 어떻게 사용되는지 이해하는 것이다.

2. 목표: TODO리스트 만들기

## 2.요구사항

1. 사용자 요구사항: 사용자가 할 일을 입력, 삭제, 출력할 수 있는 프로그램이다.

2. 기능 요구 사항

2-1 사용자에게 작업 요청 받기: 1. 할 일 추가, 2. 할 일 삭제, 3. 목록 보기, 4. 종료, 5. 할 일 수정

2-2 요청 받은 작업에 따라 아래 기능 수행: 할 일 추가를 입력했을 경우, 사용자에게 할 일을 입력 받고 저장, 할 일 삭제를 입력했을 경우, 인덱스를 입력 받고 해당 할 일 삭제, 목록 보기를 입력했을 경우, 전체 할 일 목록을 보여주기, 종료를 입력했을 경우, 프로그램 종료, 할 일 수정을 입력했을 경우, 인덱스와 할 일 (문자열)을 입력 받고, 해당 인덱스의 할 일 변경

## 3.설계 및 구현

```
#include <stdio.h>
#define MAX_TASKS 10
#define CHAR_NUM 100
#include <string.h>

int main() {
    char tasks[MAX_TASKS][CHAR_NUM] = { "" }; //할일 목록 생성
    int taskCount = 0; //할 일 개수 변수 생성

    printf("TODO 리스트 시작! \n");

    while (1) {
        int choice = -1;

        //메뉴를 출력하고 메뉴에 있는 수 입력 받기
        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);

        int terminate = 0;
        int delIndex = -1;
        int changeIndex = -1;
        char ch;
```

2. 입력: choice = 사용자가 선택한 메뉴

3. 반환값: 함수가 아니므로 없음

4. 결과: `char tasks[MAX_TASKS][CHAR_NUM]` 목록과 `int taskCount` 변수 생성  
메뉴와 현재 할 일 수 출력, choice 값을 입력 받음

5. 설명: 프로그램 시작을 위한 변수를 생성하고 메뉴를 출력하고 choice를 입력 받는다.

```
switch (choice) {  
    //할 일 입력받고 저장하기  
case 1:  
    printf("할 일을 입력하세요 (공백 없이 입력하세요): ");  
  
    //입력받은 할 일을 할 일 목록에 추가  
    scanf_s("%s", tasks[taskCount], (int)sizeof(tasks[taskCount]));  
    printf("할 일 \"%s\"가 저장되었습니다.\n\n", tasks[taskCount]);  
  
    // 할 일 개수 +1  
    taskCount++;  
    break;
```

2. 입력: `task[taskCount]`= 사용자가 입력한 할 일

3. 반환값: 함수가 아니므로 없음

4. 결과: 할 일이 저장된 `tasks`에 저장

5. 설명: choice = 1일 경우 사용자에게 할 일을 입력 받아 : `task[taskcount]`에 저장한다

```
case 2:  
    printf("삭제할 할 일의 번호를 입력해주세요. (1부터 시작):");  
    scanf_s("%d", &delIndex);  
  
    // 입력한 번호에 할 일이 존재하지 않을때 "범위를 벗어났습니다" 출력 후 break로 이동  
    if (delIndex > taskCount || delIndex <= 0) {  
        printf("삭제 범위가 벗어났습니다.\n\n");  
    }  
  
    //입력한 번호에 할 일이 존재할때 할 일 삭제하기  
    else {  
        printf("%d. %s : 할 일을 삭제합니다.\n\n", delIndex, tasks[delIndex - 1]);  
  
        //입력한 할 일 삭제  
        strcpy_s(tasks[delIndex - 1], sizeof(tasks[delIndex - 1]), "");  
  
        //할 일을 삭제하면서 생긴 할 일 목록에 빈 공간을 할 일을 한칸씩 당겨서 메우기  
        for (int i = delIndex; i < taskCount + 1; i++) {  
            strcpy_s(tasks[i - 1], sizeof(tasks[i]), tasks[i]);  
        }  
        taskCount -= 1;  
    }  
    break;
```

2. 입력: delindex: 삭제할 인덱스
3. 반환값: 함수가 아니므로 없음
4. 결과: 입력한 번호의 할 일이 tasks에서 삭제
5. 설명: choice = 2일 경우 사용자에게 삭제할 할 일의 인덱스를 입력 받는다.
6. 입력 받은 인덱스가 taskCount보다 클 경우 잘 못 입력된 것이므로 "삭제 범위를 벗어났습니다"를 출력 후 종료한다.
7. 입력 받은 인덱스가 taskCount보다 작을 경우 입력 받은 인덱스 -1 에 있는 할 일을 배열에서 제거한다
8. 제거한 후, 뒤에 있던 할 일들을 앞으로 당겨온다.

```
//할 일 목록 출력
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;

//프로그램 종료를 위해 terminate = 1로 만들고 swich문 나가기
case 4:
    terminate = 1;
    break;
```

2. 입력: 없음
3. 반환값: 함수가 아니므로 없음
4. 결과: choice = 3일 경우 할 일 목록 출력  
       choice = 4일 경우 프로그램 종료
5. 설명: choice = 3일 경우 for문을 이용해 tasks에 있는 할 일 목록을 숫자와 함께 한 줄에 하나씩 출력한다.,  
       choice = 4일 경우 terminate을 1로 만들어 swich문을 빠져나간 이후 while문도 빠져나갈 수 있게 한다

```

case 5:
    printf("수정할 할 일의 번호를 입력해주세요. (1부터 시작):");
    scanf_s("%d", &delIndex);

    //입력한 번호에 할 일이 없을때 "수정 범위가 벗어났습니다"출력후 break로 이동
    if (delIndex > taskCount || delIndex <= 0) {
        printf("수정 범위가 벗어났습니다.\n");
    }
    else {

        //할 일 삭제
        strcpy_s(tasks[delIndex - 1], sizeof(tasks[delIndex - 1]), "");

        //해당 위치에 할 일 입력받고 저장하기
        printf("수정할 할 일의 내용을 입력하세요 (공백 없이 입력하세요): ");
        scanf_s("%s", tasks[delIndex - 1], (int)sizeof(tasks[delIndex - 1]));
        printf("할 일 \"%s\"가 수정되었습니다.\n\n", tasks[delIndex - 1]);
        break;
    }
}

```

2. 입력: delindex: 수정할 인덱스
3. 반환값: 함수가 아니므로 없음
4. 결과: 입력한 번호의 할 일이 tasks에서 사라지고 입력한 문자열로 대체(수정)됨
5. 설명: choice = 4일 경우 사용자에게 수정할 할 일의 인덱스를 입력 받는다.
6. 입력 받은 인덱스가 taskCount보다 클 경우 잘 못 입력된 것이므로 "삭제 범위를 벗어났습니다"를 출력 후 종료한다.
7. 입력 받은 인덱스가 taskCount보다 작을 경우 입력 받은 인덱스 -1 에 있는 할 일을 배열에서 제거한다
8. 제거한 후 입력 받은 인덱스 -1에 할 일을 입력 받아 저장한다.

```

    // 1, 2, 3, 4, 5이외의 번호 입력시 "잘못된 선택입니다. 다시 선택하세요"출력
default:
    printf("잘못된 선택입니다. 다시 선택하세요.\n");
}

//case 4에서 terminate = 1로 했을경우 프로그램 종료
if (terminate == 1) {
    printf("종료를 선택하셨습니다. 프로그램을 종료합니다.\n");
    break;
}

//할 일이 10개 이상일때 프로그램 종료
if (taskCount >= 10) {
    printf("할 일이 다 찹습니다");
    break;
}
}
}

```

2. 입력: 없음

3. 반환값: 함수가 아니므로 없음

4. 결과: choice에 1, 2, 3, 4, 5가 아닌 다른 값을 입력했을 경우 "잘못된 선택입니다. 다시 선택하세요." 출력, terminate = 1일 경우 "종료를 선택하셨습니다. 프로그램을 종료합니다" 출력 후 while문 빠져나가기, taskCount = 10일 경우 "할 일이 다 찹니다"출력 후 while문 빠져나가기.

5. 설명: terminate = 1이면 메뉴에서 4:종료를 골랐을 때 이고, taskCount = 10이면 tasks에 저장된 할 일이 10개이기 때문에 두 경우에는 while문을 빠져나가는 것으로 프로그램을 종료한다.

## 4.테스트

1. 기능 별 테스트 결과: ① 할 일 추가하기 ② 할 일 목록 보여주기

```
1
할 일을 입력하세요 (공백 없이 입력하세요): 치킨먹기
할 일 가 저장되었습니다
```

② 할 일 목록 보여주기

```
3
할 일 목록
1. 치킨먹기
```

2. 최종 테스트 스크린샷:

```
1000 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): 과제
할 일 과제가 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. 과제

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. 과제 : 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

3
할 일 목록
```

## 5.결과 및 결론

1. 프로젝트 결과: TODO 관리프로그램을 만들었다.
2. 느낀 점: 중간고사 공부 열심히 해야겠다.