

## 1. 서론

1-1. 프로젝트 목적 및 배경: 7주차까지 배운 내용에 대한 프로젝트형 실습

1-2. 목표: 사용자가 할 일을 입력, 삭제, 수정, 출력의 기능을 하는 TODO 리스트 만들기

## 2. 요구사항

2-1. 사용자 요구사항: 사용자가 할 일을 입력, 삭제, 수정, 출력할 수 있는 프로그램

2-2. 기능 요구사항:

1. 할 일 추가(1번)를 선택할 경우 사용자에게 할 일을 입력 받고 저장
2. 할 일 삭제(2번)를 선택할 경우 사용자가 선택한 번호의 할 일을 삭제
3. 목록 보기(3번)를 선택할 경우 전체 할 일 목록을 보여주기
4. 종료(4번)를 선택할 경우 프로그램 종료
5. 할 일 수정(5번)을 선택할 경우 사용자가 선택한 번호의 할 일을 새로 입력 받고 수정
6. 할 일이 10개로 다 찰 경우 할 일이 다 찼다고 출력하고 프로그램 종료

## 3. 설계 및 구현

1. 기능 별 구현 사항

1-1. 코드블록

```
case 1: // 할 일 입력 블록
    printf("할 일을 입력하세요 (공백 없이 입력하세요): ");
    scanf_s("%s", tasks[taskCount], (int)sizeof(tasks[taskCount])); // 입력한 할 일을 tasks 2차원 배열의 문자열에 입력
    printf("할 일 \"%s\"가 저장되었습니다\n\n", tasks[taskCount]);
    taskCount++; // 다음 문자열로 넘어가기 위해 +1, 1번 재입력시 다음 문자열에 입력됨
    break;
```

2. 입력

taskCount= 할 일 번호-1

tasks[]= 할 일을 저장하는 2차원 배열

### 3. 결과

할 일이 번호가 붙어서 추가된다.

### 4. 설명

5. 사용자에게 할 일을 입력 받는다.

6. 입력 받은 할 일을 배열의 맞는 문자열에 저장한다.

7. taskCount+1 로 다음 실행 시 배열의 다음 문자열에 저장되도록 한다.

## 2-1. 코드블록

```
case 2: // 할 일 삭제 블록
    printf("삭제할 할 일의 번호를 입력해 주세요 . (1부터 시작):");
    scanf_s("%d", &delIndex); // 입력한 번호를 변수에 저장
    if (delIndex > taskCount || delIndex <= 0) {
        printf("삭제 범위가 벗어났습니다.\n"); // 번호가 범위에 맞지 않을 때 알림
    }
    else {
        printf("%d. %s : 할 일을 삭제합니다.\n", delIndex, tasks[delIndex - 1]);

        strcpy_s(tasks[delIndex - 1], sizeof(tasks[delIndex - 1]), "");
        // 2차원 배열에선 0부터 저장 되므로 입력받은 번호의 -1에 해당하는 문자열을 복사한 후 삭제

        for (int i = delIndex; i < taskCount + 1; i++) {
            strcpy_s(tasks[i - 1], sizeof(tasks[i]), tasks[i]);
        }
        taskCount -= 1; // 삭제되어 빈 번호를 매우기 위해 뒤쪽 번호를 전부 1씩 낮춤
    }
    break;
```

## 2. 입력

delIndex= 삭제하고 싶은 번호를 저장하는 변수

taskCount= 할 일의 개수

tasks[]= 할 일을 저장하는 2차원 배열

### 3. 결과

선택한 번호의 할 일이 삭제되고 그 뒤의 할 일 들이 앞으로 당겨진다.

선택한 번호가 할 일의 개수보다 클 경우 범위를 벗어나 프로그램 처음으로 돌아간다.

4. 설명

5. 사용자에게 삭제할 할 일의 번호를 입력 받는다.

6. 입력 받은 번호의 할 일을 삭제한다.

7. 삭제한 후 그 뒤에 있던 할 일들을 앞으로 당겨온다.

### 3-1. 코드블록

```
case 3: // 할 일 목록 출력 블록
    printf("할 일 목록\n");
    for (int i = 0; i < taskCount; i++) {
        printf("%d. %s\n", i + 1, tasks[i]); // 할 일의 목록을 꼭 출력
    }
    printf("\n");
    break;
```

2. 입력

taskCount= 할 일의 개수

tasks[]= 할 일을 저장하는 2차원 배열

3. 결과

할 일의 목록을 보여준다.

4. 설명

5. 할 일의 목록을 하나씩 출력한다.

### 4-1. 코드블록

```
case 4: // 종료 블록
    terminate = 1;
    break;
```

2. 입력

Terminate= 종료를 위한 변수, 1이 될 시 프로그램 종료

### 3. 결과

프로그램을 종료한다.

### 4. 설명

5. terminate 변수에 1을 저장해 종료한다.

## 5-1. 코드블록

```
case 5: // 할 일 수정 블록
    printf("수정 할 일의 번호를 입력해주세요. (1부터 시작):");
    scanf_s("%d", &changeIndex); // 입력한 번호를 변수에 저장
    if (changeIndex > taskCount || changeIndex <= 0) {
        printf("수정 범위가 벗어났습니다.\n"); // 번호가 범위에 맞지 않을 때 알림
    }
    else {
        printf("할 일을 입력하세요 (공백 없이 입력하세요):");
        scanf_s("%s", tasks[changeIndex - 1], (int)sizeof(tasks[changeIndex - 1]));
        // 2차원 배열에선 0부터 저장 되므로 입력받은 번호의 -1에 해당하는 문자열을 선택후 입력받은 할 일로 수정
        printf("%d, %s : 할 일을 수정합니다.\n", changeIndex, tasks[changeIndex - 1]);
    }
    break;
```

### 2. 입력

changeIndex= 수정하고 싶은 번호를 저장하는 변수

taskCount= 할 일의 개수

tasks[]= 할 일을 저장하는 2차원 배열

### 3. 결과

선택한 번호의 할 일을 수정한다. 선택한 번호가 할 일의 개수보다 클 경우 범위를 벗어나 프로그램 처음으로 돌아간다.

### 4. 설명

5. 사용자에게 수정할 할 일의 번호를 입력 받는다.

6. 입력 받은 번호의 할 일을 수정한다.

## 6-1. 코드블록

```
if (taskCount == 10) {  
    printf("할 일이 다 찼습니다. 프로그램을 종료합니다.");  
    terminate = 1; // 할 일이 10개가 될 시 종료  
    break;  
}
```

### 2. 입력

taskCount= 할 일의 개수

Terminate= 종료를 위한 변수, 1이 될 시 프로그램 종료

### 3. 결과

할 일의 개수가 10개가 될 경우 프로그램을 종료한다.

### 4. 설명

5. 사용자가 할 일을 추가하다 할 일이 10개가 될 시 terminate변수에 1을 저장해 프로그램을 종료한다.

## 4. 테스트

### 1. 할 일 추가하기

```
1  
할 일을 입력하세요 (공백 없이 입력하세요): 시험공부  
할 일 시험공부가 저장되었습니다
```

### 2. 할 일 삭제하기

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

### 3. 할 일 목록보기

```
3
할 일 목록
1. a
2. b
3. c
4. d
5. e
```

#### 4. 종료하기

```
4
종료를 선택하셨습니다. 프로그램을 종료합니다.
```

#### 5. 할 일 수정하기

```
5
수정할 일의 번호를 입력해주세요. (1부터 시작):1
할 일을 입력하세요 (공백 없이 입력하세요):A
1. A : 할 일을 수정합니다.
```

#### 6. 할 일이 10개일 시 종료하기

```
1
할 일을 입력하세요 (공백 없이 입력하세요): w
할 일 w가 저장되었습니다
할 일이 다 찼습니다. 프로그램을 종료합니다.
```

#### 4-2. 최종 테스트 스크린샷

-----  
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): 시험 공부  
할 일 시험공부가 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

-----  
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 = 2

-----  
2

삭제할 할 일의 번호를 입력해주세요. (1부터 시작): 1

1. 시험공부 : 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 = 1

-----  
3

할 일 목록

1. 과제

-----  
메뉴를 입력해주세요.

1. 할 일 추가
2. 할 일 삭제
3. 목록 보기
4. 종료

삭제할 할 일의 번호를 입력해주세요. (1부터 시작):1

1. 시험공부 : 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 = 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

-----

5

수정할 일의 번호를 입력해주세요. (1부터 시작):1

할 일을 입력하세요 (공백 없이 입력하세요):시험공부

1. 시험공부 : 할 일을 수정합니다.

-----

메뉴를 입력해주세요.

1. 할 일 추가
  2. 할 일 삭제
  3. 목록 보기
  4. 종료
  5. 할 일 수정
- 현재 할 일 수 = 1

-----

4

종료를 선택하셨습니다. 프로그램을 종료합니다.

C:\Users\AI\Desktop\Project10\64\Debug\Project10.exe(프로세스 11468개)이(가) 종료되었습니다(코드: 0개).  
이 창을 닫으려면 아무 키나 누르세요...



## 5. 결과 및 결론

1. 프로젝트 결과: todo 관리 프로그램을 만들었다.
2. 느낀 점: 한번에 하나씩만 시켰으면 좋겠다. 함수 만들고, 추가하고, 수정하고 여러 개 하니까 오래 걸리고 너무 힘들다. 수능 수학 30번 겨우겨우 풀었는데 갑자기 30번 하나 더 풀라는 느낌이다.