

C 보고서 작성

인공지능학부 235492 김유로

1. 서론

1. 프로젝트 목적 및 배경: 7주차까지 배운 내용에 대한 응용을 위해 실습을 진행
2. 목표: TODO 리스트 만들기

2. 요구사항

1. 사용자 요구사항: 사용자가 할 일을 입력, 삭제, 출력, 수정할 수 있는 프로그램
2. 기능 요구사항 (앞에 나열된 것들)

3. 설계 및 구현

1. 기능 별 구현 사항: 아래 3-1에서 서술

4. 테스트

1. 기능 별 테스트 결과: 아래 4-1에서 서술
2. 최종 테스트 스크린샷: 아래 4-2에서 서술

5. 결과 및 결론

1. 프로젝트 결과: TODO 리스트를 제작했다.
2. 느낀 점: 문자열의 배열과 그를 활용한 함수 활용의 공부가 매우 부족하다는 점을 알게 되었고, 평소 간단해 보이던 프로그램들도 복잡한 코딩을 통해 만들어진다는 점을 느꼈다.

3-1. 기능 별 구현 사항

```
// 할 일 추가 블록
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;
```

입력: tasks = 할 일 목록 저장 2차원 배열

taskCount = 현재 할 일 수

반환값: 함수X

결과: 할 일이 추가된 tasks

- 설명:
1. 사용자에게 추가할 할 일을 입력받는다.
 2. 입력 받은 할 일을 2차원 배열 tasks에 넣는다.
 3. 저장되었다는 문구와 함께 작업 수를 1 증가시킨다.

```
// 할 일 삭제 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;
```

입력: taskCount = 현재 할 일 수

delIndex = 할 일 삭제를 위한 인덱스 저장 변수

반환값: 함수 X

결과: 할 일이 삭제된 tasks

- 설명:
1. 사용자에게 삭제할 일의 인덱스를 입력받는다.
 2. 입력 받은 인덱스 -1에 있는 할 일을 배열에서 제거한다.
 3. 제거한 후, 뒤에 있던 할 일들을 앞으로 당겨온다.

```

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

```

입력, 반환값 동일

결과: 할 일 목록을 모두 출력함.

설명: 1. for문에 현재 할 일 수만큼의 수가 들어감

2. for문으로 할 일 수만큼 print문을 반복시켜 배열에 있는 모든 할 일을 출력함

```

// 할 일 수정 블록
printf("수정할 일의 번호를 입력해주세요. :");
scanf_s("%d", &changeIndex);
ch = getchar();
printf("새로운 할 일을 입력해주세요. :");
scanf_s("%s", &tasks[changeIndex - 1], (int)sizeof(tasks[changeIndex - 1])); // 입력받은
printf("%d번의 할 일을 %s로 수정합니다.\n", changeIndex, tasks[changeIndex - 1]);
break;

```

입력: changeIndex = 할 일 수정을 위한 인덱스 저장 변수

ch = 할 일 수정 시 버퍼를 받기 위한 문자 변수

반환값: 함수X

결과: 할 일이 수정된 tasks

설명: 1. 수정할 인덱스를 입력 받는다.

2. 새로운 할 일을 입력 받는다.

3. 입력 받은 인덱스 - 1에 있는 할 일을 배열에서 수정한다.

```

else if (taskCount ≥ 10) {
    printf("할 일이 10개로 다 찼기에 프로그램을 종료합니다.\n");
    break;
}

```

입력, 반환값 동일

결과: 프로그램 종료

설명: 할 일의 개수가 10개 이상이면 프로그램을 종료함.

4-1

할 일 추가하기

```
1
할 일을 입력하세요 (공백 없이 입력하세요): 손씻기
할 일 손씻기가 저장되었습니다
```

할 일 수정하기

```
5
수정할 일의 번호를 입력해주세요. :1
새로운 할 일을 입력해주세요. :세수하기
1번의 할 일을 세수하기로 수정합니다.
-----
```

할 일 삭제하기

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

목록 보기

```
3
할 일 목록
1. 세수하기
2. 밥먹기
```

4-2

```
-----
1
할 일을 입력하세요 (공백 없이 입력하세요): 손씻기
할 일 손씻기가 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): 밥먹기
할 일 밥먹기가 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
-----

5
수정할 일의 번호를 입력해주세요. : 1
새로운 할 일을 입력해주세요. : 세수하기
1번의 할 일을 세수하기로 수정합니다.
-----

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

3
할 일 목록
1. 세수하기
2. 밥먹기
-----

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): 2
2. 밥먹기 : 할 일을 삭제합니다.
-----

메뉴를 입력해주세요.
1. 할 일 추가
2. 할 일 삭제
3. 목록 보기
```