

할 일 관리 프로그램

231135 이나경

1. 서론

- 프로젝트 목적 및 배경: 7주차까지 배운 내용 실습, 2차원 배열 및 문자열에 대한 이해와 코드의 함수화에 대한 실습.
- 목표: TODO리스트 만들기.

2. 요구사항

1. 사용자 요구사항: 사용자가 할 일을 입력, 삭제, 출력할 수 있는 프로그램, 프로그램 종료, 할 일 수정.
2. 기능 요구사항(사용자 요구사항에 나열된 것들과 같습니다).

3. 설계 및 구현

1. 사용자가 할 일을 입력.

```
case 1: // 할 일 추가
    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++; // 할 일의 수 관리
    break;
```

- 입력: tasks = 할 일 목록을 저장한 2차원 배열, taskCount = 현재 작업 수
- 결과: 할 일이 추가된 tasks.
- 설명: 사용자에게 새로 할 일을 입력받아 tasks의 taskCount번째에 저장합니다.

2. 사용자가 할 일을 삭제.

```
case 2: // 할 일 삭제
    printf("삭제할 할 일의 번호를 입력해주세요. (1부터 시작):");
    scanf("%d", &delIndex);
    // 없는 번호를 삭제하려 했을 때 실행 x
    if (delIndex > taskCount || delIndex <= 0) {
        printf("삭제 범위가 벗어났습니다.\n");
    }
    else {
        printf("%d. %s : 할 일을 삭제합니다.\n", delIndex, tasks[delIndex - 1]);

        // tasks[delIndex - 1]자리에 ""을 넣어 삭제(복사 형식)
        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: delIndex = 할 일 삭제를 위해 입력받는 수, tasks = 할 일 목록을 저장한 2차원 배열, taskCount = 현재 작업 수

- 결과: 할 일이 삭제된 task.

- 설명: 사용자에게 삭제할 일 delIndex를 입력받습니다. delIndex를 통해 입력받은 수가 tasks의 범위 밖인 경우 실행하지 않습니다.

문자열 복사 함수로 delIndex-1번째 할 일을 제거하고, 마찬가지로 문자열 복사 함수를 통해 뒤에 있던 할 일들을 앞으로 한칸씩 당겨옵니다.

3. 출력할 수 있는 프로그램.

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

- 입력: taskCount = 현재 작업 수

- 결과: 할 일 목록을 출력합니다.

- 설명: 반복문을 통해 taskCount만큼 tasks의 요소를 하나씩 돌며 출력합니다.

4. 프로그램 종료

```
case 4: // 전체 종료
    terminate = 1; // terminate를 이용하여 while문 탈출
    break;
```

- 입력: terminate = 프로그램 종료를 결정하는 수
- 결과: 프로그램 종료

5. 할 일 수정

```
case 5: // 할 일 수정
    printf("수정할 일의 번호를 알려주세요.");
    scanf_s("%d", &changeIndex);
    ch = getchar(); // 인덱스에서 입력받은 공백 제거
    printf("새로 할 일을 알려주세요.");
    scanf_s("%s", tasks[changeIndex - 1], (int)sizeof(tasks[changeIndex - 1])); // tasks에 새 할 일 저장
    printf("할 일이 변경되었습니다: %d %s\n", changeIndex, tasks[changeIndex - 1]);
    break;
```

- 입력: tasks = 할 일 목록을 저장한 2차원 배열, changeIndex = 변경할 Index 번호
- 결과: 할 일이 변경된 tasks
- 설명: 사용자에게 수정할 일의 번호를 입력받습니다. 입력받으면서 생기는 enter를 제거합니다. 사용자에게 새로 할 일을 입력받아 tasks에 저장합니다. 사용자에게 변경된 부분을 보여줍니다.

4. 테스트

1. 기능 별 테스트 결과

1. 사용자가 할 일을 입력.

```
1
할 일을 입력하세요 (공백 없이 입력하세요): ddd
할 일 ddd가 저장되었습니다
```

2. 사용자가 할 일을 삭제.

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

3. 출력할 수 있는 프로그램.

```
3
할 일 목록
1. eee
```

4. 프로그램 종료

```
4
종료를 선택하셨습니다. 프로그램을 종료합니다.
```

5. 할 일 수정

```
5
수정할 일의 번호를 알려주세요.1
새로 할 일을 알려주세요.rrr
할 일이 변경되었습니다: 1 rrr
```

```

10000 리스트 시작!

메뉴를 입력해주세요.
1. q quit 추가
2. a add 삭제
3. r remove 보기
4. l list
5. h help 수정
현재 q 파일 수 = 0

1
q a add를 입력하세요 (공백 없이 입력하세요): ddd
q a ddd가 저장되었습니다

메뉴를 입력해주세요.
1. q quit 추가
2. a add 삭제
3. r remove 보기
4. l list
5. h help 수정
현재 q 파일 수 = 1

1
q a add를 입력하세요 (공백 없이 입력하세요): www
q a www가 저장되었습니다

메뉴를 입력해주세요.
1. q quit 추가
2. a add 삭제
3. r remove 보기
4. l list
5. h help 수정
현재 q 파일 수 = 2

2
삭제할 q a l의 번호를 입력해주세요. (1부터 시작)
1. ddd : q a l을 삭제합니다.

메뉴를 입력해주세요.
1. q quit 추가
2. a add 삭제
3. r remove 보기
4. l list
5. h help 수정
현재 q 파일 수 = 1

3
q a l 목록
1. www

메뉴를 입력해주세요.
1. q quit 추가
2. a add 삭제
3. r remove 보기
4. l list
5. h help 수정
현재 q 파일 수 = 1

5
수정할 q a l의 번호를 알려주세요.1
새로 q a l을 알려주세요.ttt
q a l이 변경되었습니다. 1 ttt

메뉴를 입력해주세요.
1. q quit 추가
2. a add 삭제
3. r remove 보기
4. l list
5. h help 수정
현재 q 파일 수 = 1

4
종료를 선택하셨습니다. 프로그램을 종료합니다.

```