

목차

1.서론

- 프로젝트 목적 및 배경 : 8주차 이전까지 배운 내용 실습
- 목표 : todo 리스트 만들기

2.요구사항

- 사용자 요구사항
- 기능 요구사항

3.설계 및 구현

- 기능 별 구현 사항 (요구사항 별 코드)

4.테스트

- 기능 별 테스트 결과 (요구사항 별 스크린 샷)
- 최종 테스트 스크린 샷 (프로그램 전체 동작 스크린 샷)

5.결과 및 결론

- 프로젝트 결과

1.서론

-프로젝트 목적 및 배경

8주차 이전까지 배운 내용 실습

-목표

Todo 리스트 만들기

2.요구 사항

-사용자 요구사항

: 사용자가 할 일을 입력, 삭제, 출력, 수정 수 있는 프로그램

-기능 요구사항

1)사용자에게 작업 요청 받기

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 일을 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 인덱스의 할 일 변경

-제약 조건

:할 일 목록은 2차원 배열($10 * 100$) = 100 개의 문자를 저장할 수 있는 문자열 10개 저장

3. 설계 및 구현

1)사용자에게 작업 요청 받기

```
// 사용자에게 메뉴를 출력하고, 메뉴를 입력받기
printf("-----\n");
printf("메뉴를 입력해 주세요.\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);
```

-입력

없음

-반환값

없음

-결과

사용자에게 메뉴를 출력하고 메뉴를 입력 받는다

-설명

1 ~ 5 번까지 출력하여 사용자에게 입력 받는다.

2) 요청 받은 작업에 따라 아래 기능 수행

```

// 입력한 내용에 따라 switch 조건문을 이용해 해당 경우 진행
switch (choice) {
case 1:
    addTask(tasks[taskCount]);
    taskCount++;
    break;
case 2:
    printf("삭제할 할 일의 번호를 입력해 주세요. (1부터 시작): ");
    scanf_s("%d", &delIndex);
    if (delIndex > taskCount || delIndex <= 0) {
        printf("삭제 범위가 벗어났습니다.\n");
    }
    else {
        delTask(delIndex, taskCount);
        taskCount -= 1;
    }
    break;
case 3:
    printf("할 일 목록\n");
    printTask(taskCount);
    break;
case 4:
    terminate = 1;
    break;
case 5:
    printf("수정할 할 일의 번호를 입력해 주세요. (1부터 시작): ");
    scanf_s("%d", &modifyIndex);
    ch = getchar();
    printf("새로운 할 일을 입력해 주세요");
    scanf_s("%s", tasks[modifyIndex - 1], (int)sizeof(tasks[modifyIndex - 1]));
    printf("새로운 할 일이 추가되었습니다: %d. %s\n", modifyIndex, tasks[modifyIndex - 1]);
    break;
default:
    printf("잘못된 선택입니다. 다시 선택하세요.\n");
}

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

```

-입력

1 ~ 5 중에서 값을 사용자가 입력한다.

-반환값

해당하는 경우의 문장을 출력

-결과

1 ~ 5번에 해당하는 값을 입력하면 그 값에 맞는 경우로 진행

-설명

1-할 일 추가를 입력했을 경우, 사용자에게 할 일을 입력 받고 저장

2-할 일 삭제를 입력했을 경우, 인덱스를 입력 받고 해당 할 일을 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 인덱스의 할 일 변경

4.테스트

-기능 별 테스트 결과

1)사용자에게 작업 요청 받기

```
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
-----
```

2) 요청 받은 작업에 따라 아래 기능 수행

```
-----
1
할 일을 입력하세요 (공백 없이 입력하세요): 청소
할 일 청소가 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
삭제할 할 일의 번호를 입력해주세요. (1부터 시작):1
1. 청소 : 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
1.
2. 청소
-----
메뉴를 입력해주세요.
1. 할 일 추가
2. 할 일 삭제
3. 목록 보기
4. 종료
5. 할 일 수정
현재 할 일 수 = 2
-----

```

```

4
종료를 선택하셨습니다. 프로그램을 종료합니다.

```

```

5
수정할 할 일의 번호를 입력해주세요. (1부터 시작): 1
새로운 할 일을 입력해주세요:잠자기
새로운 할 일이 추가되었습니다: 1. 잠자기
-----
메뉴를 입력해주세요.
1. 할 일 추가
2. 할 일 삭제
3. 목록 보기
4. 종료
5. 할 일 수정
현재 할 일 수 = 2
-----

```

-최종 테스트 스크린 샷

```
-----
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.
2. 청소
-----
```

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

```
4
종료를 선택하셨습니다. 프로그램을 종료합니다.
```

5.결과 및 결론

결과 : todo 관리 프로그램을 만들어서 실행 했다.