C프로그래밍 및 실습 TODO리스트_ 프로그램_생성_보고서 인공지능학부_231583_유희원

1. 서론

   1) 프로젝트 목적 및 배경: 7주차까지 배운 내용에 대한 실습을 위해 진행

   2) 목표: TODO 리스트 만들기

2. 요구사항

   1) 사용자 요구사항: 할 일을 입력 및 삭제, 할 일 목록을 출력할 수 있는 프로그램

   2) 기능 요구사항: 할 일 추가, 삭제, 목록 출력, 프로그램 종료, 수정

3. 설계 및 구현

   1) 기능 별 구현 사항

      (1) 할 일 추가

```
case 1:
    // 할 일 입력받기
    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;
```

      A. 코드블록 스크린샷(위)

      B. 입력

         - taskCount = 현재 할 일의 수
         - tasks = 할 일 목록을 저장하는 2차원 배열

      C. 반환값

         - 함수가 아니므로 없음

      D. 결과

         - 할 일이 추가된 tasks

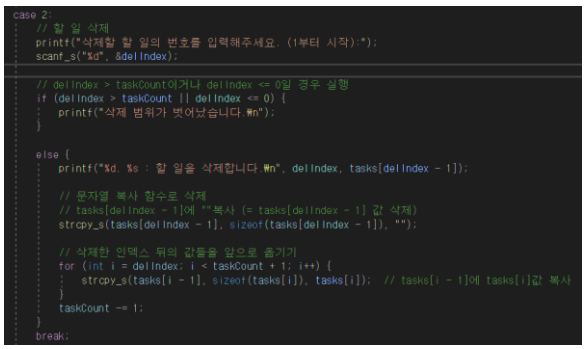
      E. 설명

         a. 사용자에게 할 일을 입력 받기

         b. 할 일을 tasks의 taskCount 값의 행에 저장하기

c. 저장 후 taskCount + 1

(2) 할 일 삭제



```
case 2:
    // 할 일 삭제
    printf("삭제할 할 일의 번호를 입력해주세요. (1부터 시작):");
    scanf_s("%d", &delIndex);

    // delIndex > taskCount이거나 delIndex <= 0일 경우 실행
    if (delIndex > taskCount || delIndex <= 0) {
        printf("삭제 범위가 벗어났습니다.₩n");
    }

    else {
        printf("%d. %s : 할 일을 삭제합니다.₩n", delIndex, tasks[delIndex - 1]);

        // 문자열 복사 함수로 삭제
        // tasks[delIndex - 1]에 ""복사 (= tasks[delIndex - 1] 값 삭제)
        strcpy_s(tasks[delIndex - 1], sizeof(tasks[delIndex - 1]), "");

        // 삭제한 인덱스 뒤의 값들을 앞으로 옮기기
        for (int i = delIndex; i < taskCount + 1; i++) {
            strcpy_s(tasks[i - 1], sizeof(tasks[i]), tasks[i]);  // tasks[i - 1]에 tasks[i]값 복사
        }
        taskCount -= 1;
    }
    break;
```

A. 코드블록 스크린샷(위)

B. 입력

- delIndex = 입력받은 삭제할 인덱스 번호
- taskCount = 현재 할 일의 수
- tasks = 할 일 목록을 저장하는 2차원 배열

C. 반환값

- 함수가 아니므로 없음

D. 결과

- 입력받은 인덱스 - 1의 위치에 해당하는 할 일이 삭제되고 뒤에 있는 할 일들이 앞으로 당겨진 tasks

E. 설명

a. 사용자에게 삭제할 할 일 번호를 입력 받기

b. 입력받은 번호를 delIndex에 저장

c. delIndex 값이 taskCount보다 크거나, 0보다 같거나 작을 때 삭제 범위을 벗어났음을 알리는 문구 출력

d. 위의 경우가 아닌 경우 입력받은 인덱스 – 1 위치에 있는 tasks 값 삭제

e. 삭제 후 뒤의 값들을 앞으로 당기기

(3) 할 일 목록 출력

```
case 3:
    // 할 일 목록 출력
    printf("할 일 목록₩n");

    // i < taskCount까지 task 각 행을 출력
    for (int i = 0; i < taskCount; i++) {
        printf("%d. %s ₩n", i + 1, tasks[i]);
    }
    printf("₩n");
    break;
```

A. 코드블록 스크린샷(위)

B. 입력

- taskCount = 현재 할 일의 수
- tasks = 할 일 목록을 저장하는 2차원 배열

C. 반환값

- 함수가 아니므로 없음

D. 결과

- 할 일 목록 출력

E. 설명

a. i < taskCount 일 때까지 for 문 반복

b. for문: i + 1. Tasks의 i행 값 출력

c. 출력 후 i + 1

(4) 프로그램 종료

```
case 4:
    // 4. 프로그램 종료
    terminate = 1;
    break;
```

```
// terminate가 1이 되면 프로그램 종료
if (terminate == 1) {
    printf("종료를 선택하셨습니다. 프로그램을 종료합니다.₩n");
    break;
}
```

A. 코드블록 스크린샷(위)

B. 입력

- terminate = 프로그램 종료를 위한 변수

C. 반환값

- 함수가 아니므로 없음

D. 결과

- 프로그램 종료

E. 설명

a. terminate에 1 저장

b. 프로그램 종료

(5) 할 일 수정

```
case 5:
    // 5. 할 일 수정
    printf("수정할 할 일의 번호를 입력해주세요: ");  // 수정할 할 일 번호 입력 받기
    scanf_s("%d", &changeIndex);

    ch = getchar();  // enter 값 삭제

    printf("수정된 할 일을 입력해주세요: ");  // 수정된 할 일 내용 입력 받기
    scanf_s("%s", tasks[changeIndex-1], (int)sizeof(tasks[changeIndex-1]));  // 입력받은 번호보다 하나 작은 인덱스에 할 일 저장
    printf("""%d""번 할 일이 ""%s""로 수정되었습니다.₩n", changeIndex, tasks[changeIndex - 1]);
    break;
```

A. 코드블록 스크린샷(위)

B. 입력

- changeIndex = 입력받은 수정할 번호
- tasks = 할 일 목록을 저장하는 2차원 배열

C. 반환값

- 함수가 아니므로 없음

D. 결과

- 특정번호의 할 일이 수정된 tasks

E. 설명

a. 사용자에게 수정할 번호를 입력 받기

b. 입력받은 번호를 changeIndex에 저장

c. 버퍼제거

d. 사용자에게 수정된 할 일을 입력 받기

e. 입력받은 할 일을 tasks의 changeIndex-1의 행에 저장

4. 테스트

1) 기능 별 테스트 결과: (요구사항 별 스크린샷)

(1) 할 일 추가

```
1
할 일을 입력하세요 (공백 없이 입력하세요): 밥먹기
할 일 밥먹기가 저장되었습니다
```

(2) 할 일 삭제

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

(3) 할 일 목록 출력

```
3
할 일 목록
1. 밥먹기
2. 잠자기
3. 공부하기
```

(4) 프로그램 종료

```
4
종료를 선택하셨습니다. 프로그램을 종료합니다.
```

(5) 할 일 수정

```
5
수정할 할 일의 번호를 입력해주세요: 1
수정된 할 일을 입력해주세요: 운동하기
1번 할 일이 운동하기로 수정되었습니다.
```

2) 최종 테스트 스크린샷

```
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
----------------
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 = 1
----------------
1
할 일을 입력하세요 (공백 없이 입력하세요): 잠자기
할 일 잠자기가 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
----------------
3
할 일 목록
1. 밥먹기
2. 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
----------------
2
삭제할 할 일의 번호를 입력해주세요. (1부터 시작):1
1. 밥먹기 : 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. 잠자기
----------------
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 할 일의 번호를 입력해주세요: 1
수정된 할 일을 입력해주세요: 공부하기
1번 할 일이 공부하기로 수정되었습니다.
----------------
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.
```

5. 결과 및 결론

1) 프로젝트 결과: TODO 리스트 관리 프로그램을 만듦.

2) 느낀점: 배운 내용보다 훨씬 심화된 느낌이라 코드를 이해하고 작성하는 데에 어려움이 있었다. 타자를 치는 속도가 느려 시간이 부족해서 타자 연습을 할 필요성을 느꼈다