

Todo_manager

224632 조혜민

1. 서론

1. 프로젝트 목적 및 배경: 사용자의 할 일을 관리해주는 프로그램을 만들며 지금까지 배운 것 통해서 프로젝트를 진행한다.

2. 목표: TODO 리스트 만들어 사용자에게 작업요청 받고 요청받은 작업에 따라 수행하는 프로그램을 짜는 것이다.

2. 요구사항

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

2. 기능 요구사항

① 사용자에게 작업 요청 받기(1. 할 일 추가, 2. 할 일 삭제, 3. 목록 보기, 4. 종료, 5. 할 일 수정)

② 요청 받은 작업에 따라 아래 기능 수행

(1->사용자에게 할 일을 입력 받고 저장

2->인덱스를 입력받고 해당 할 일 삭제

3->전체 할 일 목록 보여주기

4->종료를 입력했을 경우 프로그램 종료

5->인덱스와 할 일을 입력받고, 해당 인덱스 할 일을 변경)

③ 할 일이 10개로 다 찬 경우는 할 일이 다 찼다고 출력하고 프로그램 종료

3. 설계 및 구현

1. 기능 별 구현 사항: (요구사항 별 코드)

사용자에게 작업 요청 받기

```
printf("-----\n");  
printf("메뉴를 입력해주세요.\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("-----Wn");
scanf_s("%d", &choice);
```

printf로 인해 메뉴를 입력하고 scanf_s를 통해 choice로 받는다.

② 요청 받은 작업에 따라 아래 기능 수행

```
switch (choice) {
    //할일 입력할때
    case 1:
        printf("할 일을 입력하세요 (공백 없이 입력하세요): ");
        scanf_s("%s", tasks[taskCount], (int)sizeof(tasks[taskCount]));
        printf("할 일 \"%s\"가 저장되었습니다WnWn", tasks[taskCount]);
        taskCount++;
        break;
```

->case1에서는 scanf_s를 통해 taskCount번호에 있는거에 넣고 sizeof도 설정해준다. 그 이후 ++을 한다.

```
    //할일 삭제할때
    case 2:
        // 할 일 삭제하는 코드 블록
        printf("삭제 할 할 일의 번호를 입력해주세요. (1부터 시작):");
        scanf_s("%d", &delIndex);
        if (delIndex > taskCount || delIndex <= 0) {
            printf("삭제 범위가 벗어났습니다.Wn");
        }
        else {
            printf("%. %s : 할 일을 삭제합니다.Wn", 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]);
            }
            //삭제했으니 -1해 줌
            taskCount -= 1;
        }
        break;
    //할일 목록을 선택했을때
```

->taskCount는 현재 작업수고 tasks는 할일 목록 저장 2차원 배열이다. 결과는 할일이 삭제된 tasks이다. 삭제했으니 taskCount를 마이너스해서 저장해둔다. 사용자에게 삭제할 할일의 인덱스를 입력받고 입력받은 인덱스에서 제거한다 (-1해줘야함)제거한 후 한칸씩 당겨줌

```
    case 3:
        printf("할 일 목록Wn");
        for (int i = 0; i < taskCount; i++) {
```

```

        //할일 목록 출력
        printf("%d. %s\n", i + 1, tasks[i]);
    }
    printf("\n");
    break;
    //종료할때

```

->할일 목록을 taskCount 0부터 출력하는데 나오는건 i+1로 하여 1부터 쪽 나온다.

```

case 4:
    terminate = 1;
    break;
    //할일 수정

```

->종료하는 값은 1로 설정해둔다.

```

case 5:
    printf("할 일 수정할 번호를 입력하세요: \n");
    scanf_s("%d", &delIndex);
    if (delIndex > taskCount || delIndex <= 0) {
        printf("수정 범위가 벗어났습니다.\n");
    }
    else {
        printf("%d : 할 일을 수정합니다.\n", delIndex);

        strcpy_s(tasks[delIndex - 1], sizeof(tasks[delIndex - 1]),
tasks[delIndex - 1]);

        printf("할 일을 입력하세요 (공백 없이 입력하세요): ");
        scanf_s("%s", tasks[delIndex - 1], (int)sizeof(tasks[delIndex
- 1]));

        printf("할 일 \"%s\"가 저장되었습니다\n\n", tasks[delIndex -
1]);

    }
    break;
    //다른걸 입력했을때

```

->할일 수정하여 삭제하고 그 자리에 저장하게 되는 함수를 입력하였다. delIndex에 수정할 번호를 입력하고 -1을 하여 자리를 찾고 대체를 한다. 수정하여 넣음

```

default:
    printf("잘못된 선택입니다. 다시 선택하세요.\n");
}

```

->다른 걸 입력하면 잘못된 선택이라 한다

```

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

```

->4번에서 종료라면 종료를 시키다.

```

if (taskCount >= 10) {
    printf("할일 10개를 넘었습니다. 프로그램을 종료하였습니다.\n");
    break;
}

```

->taskcount가 10과 같아진다면 멈춘다.

```
}  
}
```

4. 테스트

- (테스트 : 입력에 따른 원하는 결과나 나오는지 확인하는 과정)

1. 기능 별 테스트 결과: (요구사항 별 스크린샷)

- ① 사용자에게 작업 요청 받기(1. 할 일 추가, 2. 할 일 삭제, 3. 목록 보기, 4. 종료, 5. 할 일 수정)

```
g(ccc@yamin:~/cncnrc/10-1-1) $  
-----  
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->사용자에게 할일을 입력 받고 저장

```
현재 할 일 수 = 0  
-----  
1  
할 일을 입력하세요 (공백 없이 입력하세요): C언어공부  
할 일 C언어공부가 저장되었습니다  
-----  
메뉴를 입력해주세요.  
1. 할 일 추가
```

2->인덱스를 입력받고 해당 할 일 삭제

```
현재 할 일 수 = 1  
-----  
2  
삭제할 할 일의 번호를 입력해주세요. (1부터 시작):1  
1. C언어공부 : 할 일을 삭제합니다.  
-----
```

3->전체 할일 목록 보여주기


```
Microsoft Visual Studio 디버그 콘솔

메뉴를 입력해주세요.
1. 할 일 추가
2. 할 일 삭제
3. 목록 보기
4. 종료
5. 할 일 수정
현재 할 일 수 = 8
-----

할 일을 입력하세요 (공백 없이 입력하세요):
1
할 일 11가 저장되었습니다
-----

메뉴를 입력해주세요.
1. 할 일 추가
2. 할 일 삭제
3. 목록 보기
4. 종료
5. 할 일 수정
현재 할 일 수 = 9
-----

할 일을 입력하세요 (공백 없이 입력하세요): 1
할 일 1가 저장되었습니다

할 일 10개를 넘었습니다. 프로그램을 종료하였습니다.
```

2. 최종 테스트 스크린샷: (프로그램 전체 동작 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): CPP공부
할 일 CPP공부가 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): C공부
할 일 C공부가 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

```

5
할 일 수정할 번호를 입력하세요:
2
2 : 할 일을 수정합니다.
할 일을 입력하세요 (공백 없이 입력하세요): 알고리즘공부
할 일 알고리즘공부가 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. CPP공부
2. 알고리즘공부

-----
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

1. 프로젝트 결과: 투두 관리프로그램을 만들었다. 수정 코드를 추가하고 10개가 넘으면 종료하는 프로그램을 만들었다.

2. 느낀 점: 처음부터 짜기에는 너무 어려움이있지만 case이 문을 좀 더 공부해야겠다는 생각이 들었다.

3.추가로 함수문으로 한번 해보기를 진행하였는데 함수가 더 깔끔하다는 것을 깨달았다.