

서론

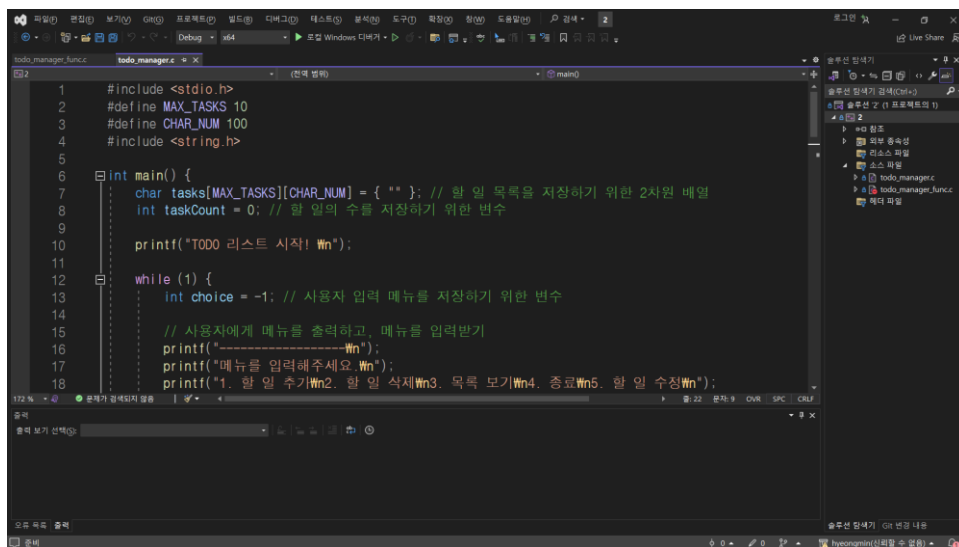
1. 프로젝트 목적 및 배경: 지금까지 배웠던 내용에 대한 실습과 코딩 실력을 기르기 위해 진행
2. 2. 목표: C언어를 이용해 TODO 리스트 만들기

요구사항

1. 사용자 요구사항: 사용자가 할 일 입력, 삭제, 출력할 수 있는 프로그램
2. 기능 요구사항 : 또한 같다

설계 및 구현

1. 기능 별 구현 사항 (요구 사항 별 코드)



```
1 #include <stdio.h>
2 #define MAX_TASKS 10
3 #define CHAR_NUM 100
4 #include <string.h>
5
6 int main() {
7     char tasks[MAX_TASKS][CHAR_NUM] = { "" }; // 할 일 목록을 저장하기 위한 2차원 배열
8     int taskCount = 0; // 할 일의 수를 저장하기 위한 변수
9
10    printf("TODO 리스트 시작! \n");
11
12    while (1) {
13        int choice = -1; // 사용자 입력 메뉴를 저장하기 위한 변수
14
15        // 사용자에게 메뉴를 출력하고, 메뉴를 입력받기
16        printf("-----\n");
17        printf("메뉴를 입력해주세요 \n");
18        printf("1. 할 일 추가\n2. 할 일 삭제\n3. 목록 보기\n4. 종료\n5. 할 일 수정\n");
19    }
```

#include <stdio.h>: printf와 scanf를 사용하기 위한 표준 입력 및 출력 라이브러리를 포함

#define MAX_TASKS 10: 이 지시문은 MAX_TASKS라는 상수를 정의합니다. 이 상수는 최대 할 일 수를 나타내고 10으로 초기화됨

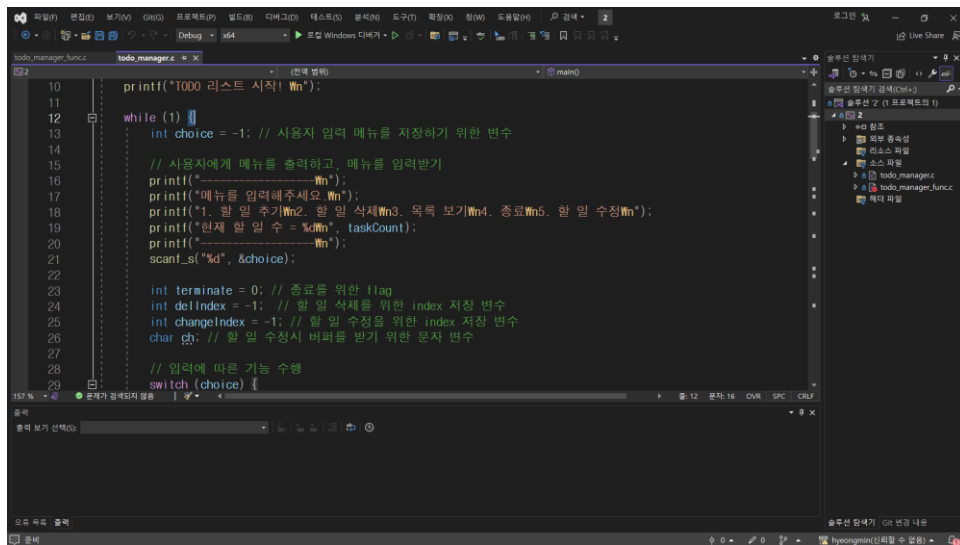
#define CHAR_NUM 100: 이 지시문은 CHAR_NUM이라는 상수를 정의합니다. 이 상수는 할 일 하나의 최대 문자 수를 나타내고 100으로 초기화됨

#include <string.h>: 이 지시문은 문자열 처리 함수를 사용하기 위한 라이브러리를 포함함

strcpy_s와 같은 문자열 복사 함수를 사용할 때 필요함

Main 함수 할 일 목록을 저장하기 위한 2차원 배열

TaskCount 는 할 일의 수를 저장하기 위한 변수



```
10 printf("TODO 리스트 시작! \n");
11
12 while (1) {
13     int choice = -1; // 사용자 입력 메뉴를 저장하기 위한 변수
14
15     // 사용자에게 메뉴를 출력하고, 메뉴를 입력받기
16     printf("-----\n");
17     printf("메뉴를 입력해주세요.\n");
18     printf("1. 할 일 추가\n2. 할 일 삭제\n3. 목록 보기\n4. 종료\n5. 할 일 수정\n");
19     printf("현재 할 일 수 = %d\n", taskCount);
20     printf("-----\n");
21     scanf_s("%d", &choice);
22
23     int terminate = 0; // 종료를 위한 flag
24     int delIndex = -1; // 할 일 삭제를 위한 index 저장 변수
25     int changeIndex = -1; // 할 일 수정을 위한 index 저장 변수
26     char ch; // 할 일 수정시 버퍼를 받기 위한 문자 변수
27
28     // 입력에 따른 기능 수행
29     switch (choice) {
```

While 반복문을 계속 작동하도록 함

Printf 로 프로그램의 기본적인 틀 출력

메뉴 입력 1. 할 일 추가\n2. 할 일 삭제\n3. 목록 보기\n4. 종료\n5. 할 일 수정 메뉴선택

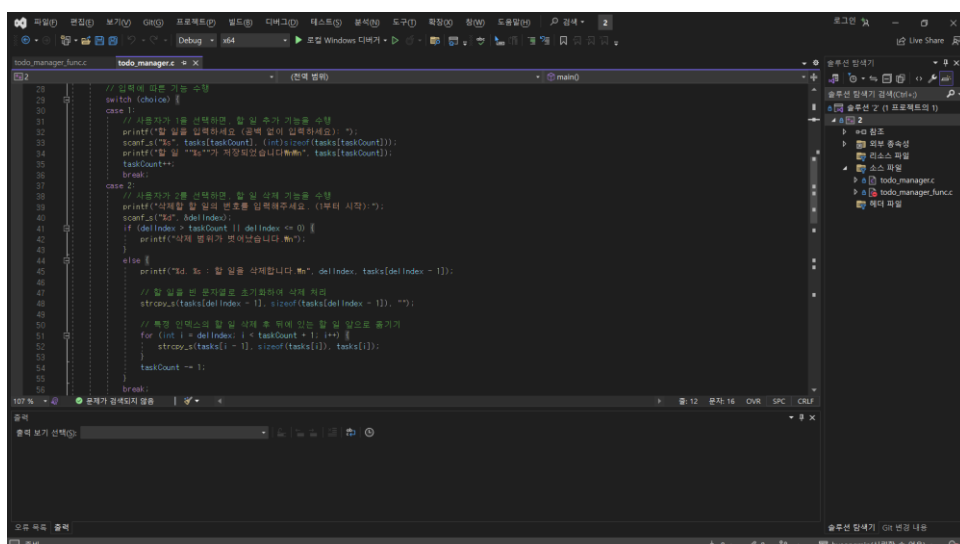
Scanf_s로 내가 선택한거 출력

int terminate = 0; // 종료를 위한 flag

int delIndex = -1; // 할 일 삭제를 위한 index 저장 변수

int changeIndex = -1; // 할 일 수정을 위한 index 저장 변수

char ch; // 할 일 수정시 버퍼를 받기 위한 문자 변수



```
28 // 입력에 따른 기능 수행
29 switch (choice) {
30     case 1:
31         // 사용자가 1을 선택하면 할 일 추가 기능을 수행
32         printf("할 일을 입력하세요 (숫자 없이 입력해주세요.) : ");
33         scanf_s("%s", tasks[taskCount], (int)sizeof(tasks[taskCount]));
34         printf("할 일 \"%s\"가 저장되었습니다.\n", tasks[taskCount]);
35         taskCount++;
36         break;
37     case 2:
38         // 사용자가 2를 선택하면 할 일 삭제 기능을 수행
39         printf("삭제할 할 일의 번호를 입력해주세요 (부터 시작) : ");
40         scanf_s("%d", &delIndex);
41         if (delIndex > taskCount || delIndex <= 0) {
42             printf("삭제 범위가 벗어났습니다.\n");
43         }
44         else {
45             printf("Id %d : 할 일을 삭제합니다.\n", delIndex, tasks[delIndex - 1]);
46
47             // 할 일을 arr 문자열로 초기화하여 삭제 처리
48             strcpy_s(tasks[delIndex - 1], sizeof(tasks[delIndex - 1]), "");
49
50             // 특정 인덱스의 할 일 삭제 후 뒤에 있는 할 일 앞으로 옮기기
51             for (int i = delIndex; i < taskCount; i++) {
52                 strcpy_s(tasks[i - 1], sizeof(tasks[i]), tasks[i]);
53             }
54             taskCount--;
55         }
56         break;
57 }
```

switch (choice) { }: 이 코드는 choice 변수의 값을 기반으로 다양한 경우(case)를 처리하는 스위치 문을 시작

choice는 사용자가 입력한 정수값 사용자가 어떤 작업을 선택했는지를 결정하는 데 사용

case 1:: 이 부분은 choice 변수가 1일 때 실행됩니다. 1을 선택했을 때 다음 코드 블록이 실행

printf("할 일을 입력하세요 (공백 없이 입력하세요): ") 사용자에게 메시지를 표시하여 할 일을 입력하라는 요청

scanf_s("%s", tasks[taskCount], (int)sizeof(tasks[taskCount])) 이 줄은 사용자로부터 입력을 받기 scanf_s 함수는 포맷 문자열 "%s"를 사용하여 문자열을 입력 받기 입력된 문자열은 tasks[taskCount] 배열에 저장

taskCount 변수는 현재 할 일 목록에 저장된 항목의 수를 추적하기 위해 사용

printf("할 일 \"%s\"가 저장되었습니다.\n\n", tasks[taskCount]) 이 줄은 사용자가 입력한 할 일을 화면에 표시합니다. %s 형식 지정자를 사용하여 사용자가 입력한 할 일 문자열을 출력. 그리고 tasks[taskCount] 배열에 저장된 문자열을 표시

taskCount++; 이 줄은 taskCount 변수를 증가

이것은 사용자가 입력한 할 일을 배열에 저장 다음 할 일을 저장할 위치를 추적하는 데 사용
끝으로, break; 문은 case 1: 블록을 종료합니다.

2를 누르면

printf("삭제할 할 일의 번호를 입력해주세요. (1부터 시작):") 이 줄은 사용자에게 삭제할 할 일의 번호를 입력하라는 메시지를 표시

scanf_s("%d", &delIndex); 이 줄은 사용자로부터 정수값을 입력받음

delIndex 변수에 사용자의 입력값이 저장됩니다.

if (delIndex > taskCount || delIndex <= 0) 은 입력한 번호(delIndex)가 유효한 범위 내에 있는지 확인 delIndex가 할 일 목록의 크기(taskCount)를 초과하거나 1 이하의 값이라면, 삭제 범위가 벗어난것

else 이 부분은 if 문의 조건이 거짓일 때 실행

printf("%d. %s : 할 일을 삭제합니다.\n", delIndex, tasks[delIndex - 1]) 이 줄은 삭제할 할 일을 화

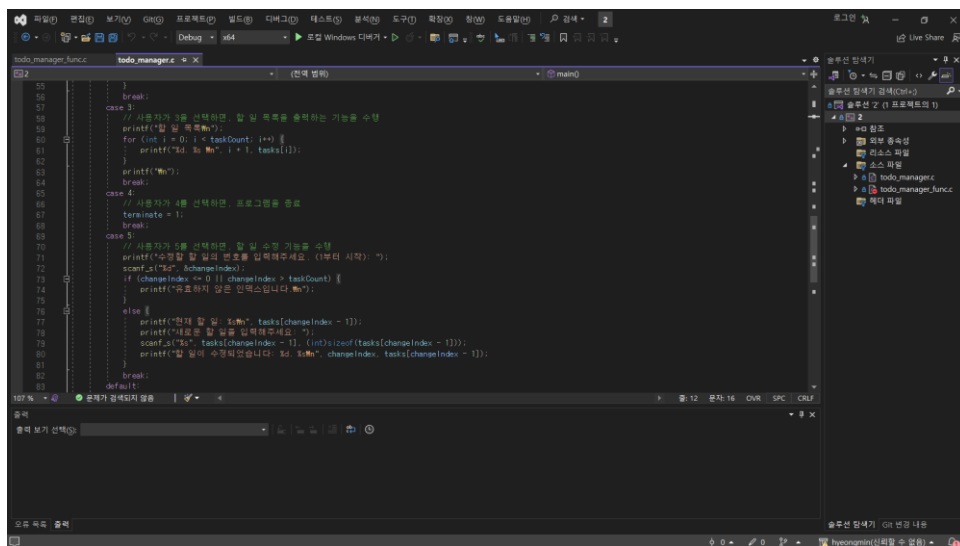
면에 표시 삭제할 할 일의 번호와 해당 할 일 내용을 출력.

`strcpy_s(tasks[delIndex - 1], sizeof(tasks[delIndex - 1]), "");`이 줄은 선택된 할 일을 삭제하기 위해 해당 배열 요소(`tasks[delIndex - 1]`)를 빈 문자열("")로 초기화

`for (int i = delIndex; i < taskCount + 1; i++) { ... }:` 이 부분은 삭제된 할 일 뒤에 있는 할 일들을 앞으로 이동시키는 작업을 수행 for 루프를 사용하여 삭제된 할 일의 인덱스(`delIndex`) 이후의 할 일을 한 칸씩 앞으로 옮김. 이렇게 하면 삭제한 할 일의 공간이 비워지지 않고 다른 할 일이 앞으로 이동됨

`taskCount -= 1;` 삭제한 후에 할 일 목록의 크기(`taskCount`)를 1 감소

`break;` case 2: 블록을 종료하고 스위치문에서 빠져나가요



case 3: - 할 일 목록 출력

`printf("할 일 목록\n");` 이 줄은 사용자에게 할 일 목록을 출력한다는 메시지를 표시

`for (int i = 0; i < taskCount; i++) { ... }:` 은 현재 저장된 모든 할 일을 출력하는 반복문

`taskCount` 변수는 현재 할 일 목록에 저장된 항목의 수를 나타내며, for 루프를 사용하여 각 할 일을 번호와 함께 출력합니다. `i`는 배열의 인덱스를 나타냄

`printf("%d. %s\n", i + 1, tasks[i]);` 이 줄은 현재 할 일의 번호와 내용을 출력

`i + 1`은 1부터 시작하는 번호를 표시하고, `tasks[i]`는 해당 할 일의 내용을 나타냄

`printf("\n");` 각 할 일을 출력한 후에 빈 줄을 출력하여 각 할 일 사이에 공백을 만들

`break;` case 3: 블록을 종료하고 스위치문에서 빠져나갑니다.

case 4: - 프로그램 종료

terminate = : 이 줄은 terminate라는 변수를 1로 설정하여 프로그램을 종료하도록 표시 이것은 일반적으로 루프나 메인 함수에서 사용되며, 프로그램이 종료될 때 terminate 값이 확인되어 종료 동작이 수행됩니다.

break;; case 4: 블록을 종료하고 스위치문에서 빠져나갑니다. 이렇게 함으로써 프로그램은 종료됩니다.

Case 5 : 할일 수정

printf("수정할 할 일의 번호를 입력해주세요. (1부터 시작): ") 은 사용자에게 수정할 할 일의 번호를 입력하라는 메시지를 표시 사용자는 1부터 시작하는 번호를 입력해야 함

scanf_s("%d", &changeIndex) 은 사용자로부터 정수값을 입력받음

changeIndex 변수에 사용자의 입력값이 저장됨

if (changeIndex <= 0 || changeIndex > taskCount) { ... }: 이 부분은 입력한 번호(changeIndex)가 유효한 범위 내에 있는지 확인

만약 입력된 번호가 할 일 목록의 범위를 벗어나거나 1보다 작거나 같은 경우, "유효하지 않은 인덱스입니다."라는 메시지를 출력

else { ... }: 이 부분은 if 문의 조건이 거짓일 때 실행

printf("현재 할 일: %s\n", tasks[changeIndex - 1]); 은 선택된 할 일을 표시합니다. 현재 선택한 할 일의 내용을 출력

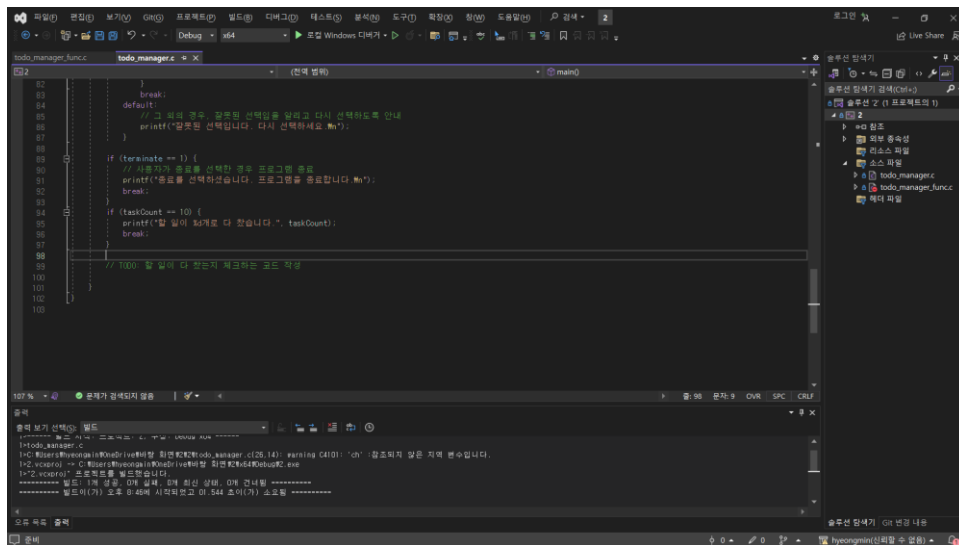
printf("새로운 할 일을 입력해주세요: ");: 사용자에게 새로운 할 일을 입력하라는 메시지를 표시

scanf_s("%s", tasks[changeIndex - 1], (int)sizeof(tasks[changeIndex - 1]))은 사용자가 입력한 새로운 할 일을 받아와서 tasks 배열의 해당 위치(changeIndex - 1)에 저장함

printf("할 일이 수정되었습니다: %d. %s\n", changeIndex, tasks[changeIndex - 1]) 수정이 완료된 할 일을 표시 새로운 할 일의 내용과 수정된 할 일의 번호를 출력

break;; case 5: 블록을 종료하고 스위치문에서 빠져나감

default: 와 printf("잘못된 선택입니다. 다시 선택하세요.\n"); 잘못된 선택을 하면 다시 선택하도록 안내함



if (terminate == 1) { ... } 은 terminate 변수가 1로 설정되었을 때 실행 terminate 변수는 이전에 case 4:에서 사용자가 프로그램을 종료하겠다고 선택했을 때 설정된 변수로, 프로그램 종료를 표시

이 조건은 사용자가 프로그램을 종료하겠다고 선택한 경우 해당 메시지를 출력하고 break를 사용하여 스위치문에서 빠져나감

if (taskCount == 10) { ... }: 은 taskCount 변수가 10으로 설정되었을 때 실행

taskCount 변수는 현재 할 일 목록에 저장된 항목의 수를 나타내며, 이 조건은 할 일 목록이 최대 10개로 다 찼을 때 해당 메시지를 출력하고 break를 사용하여 스위치문에서 빠져나감

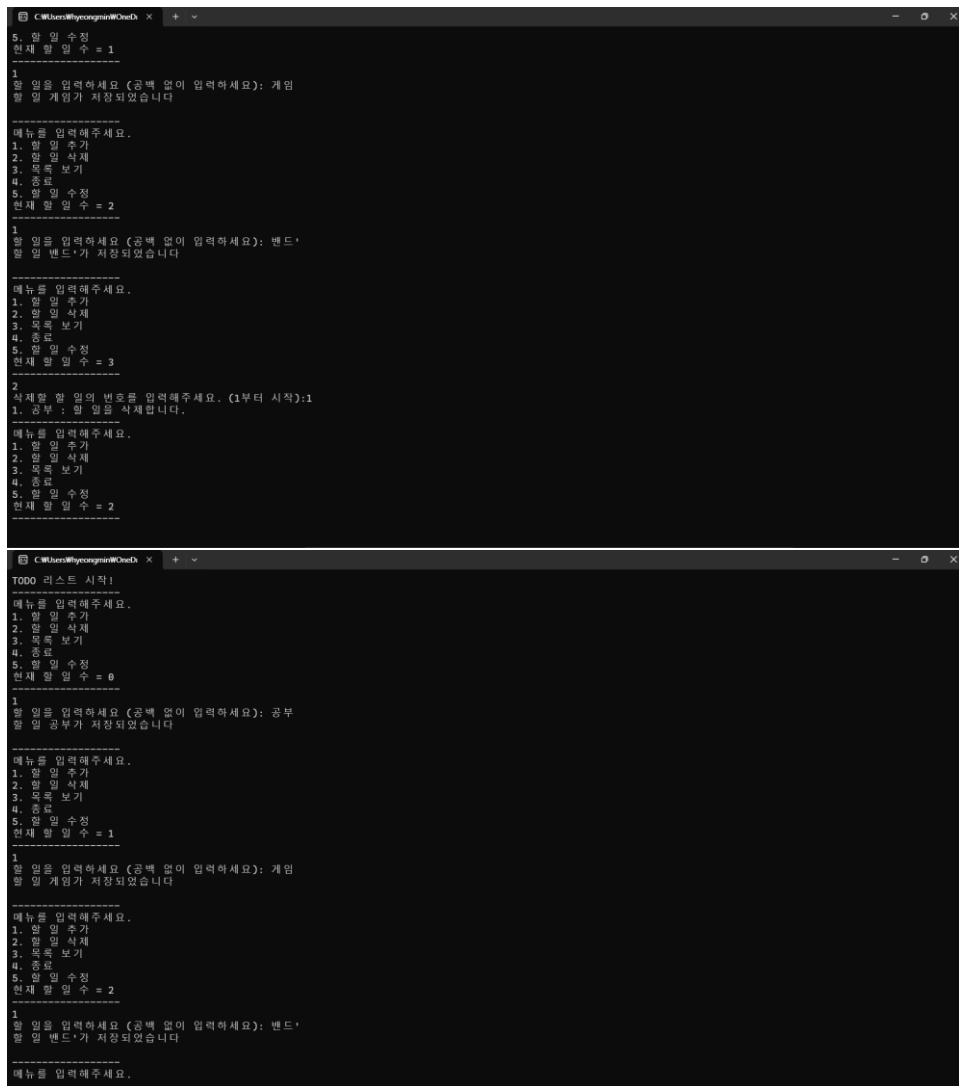
테스트

1. 기능별 테스트 결과

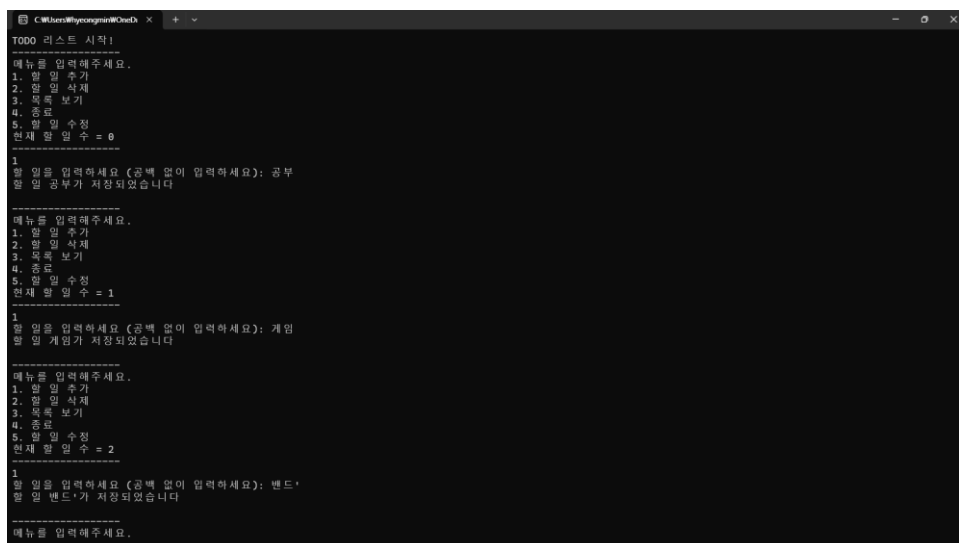
```
Microsoft Visual Studio 12비트 x
메뉴를 입력해주세요.
1. 할 일 추가
2. 할 일 삭제
3. 목록 보기
4. 종료
5. 할 일 수정
현재 할 일 수 = 2
-----
3
할 일 목록
1. 게임
2. 밴드
-----
메뉴를 입력해주세요.
1. 할 일 추가
2. 할 일 삭제
3. 목록 보기
4. 종료
5. 할 일 수정
현재 할 일 수 = 2
-----
5
수정할 할 일의 번호를 입력해주세요. (1부터 시작): 1
현재 할 일: 게임
새로운 할 일을 입력해주세요: 친구들과만나기
할 일이 수정되었습니다: 1. 친구들과만나기
-----
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.
C:\Users\hyeonjin\OneDrive\바탕 화면\2\64\Debug\2.exe(프로세스 2384개)이(가) 종료되었습니다(코드: 0개).
이 창을 닫으려면 아무 키나 누르세요...]
```

```
C:\Users\hyeonjin\OneDr
2
삭제할 할 일의 번호를 입력해주세요. (1부터 시작):1
1. 공부 : 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 = 2
-----
3
할 일 목록
1. 게임
2. 밴드
-----
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 할 일의 번호를 입력해주세요. (1부터 시작): 1
현재 할 일: 게임
새로운 할 일을 입력해주세요: 친구들과만나기
할 일이 수정되었습니다: 1. 친구들과만나기
-----
메뉴를 입력해주세요.
1. 할 일 추가
2. 할 일 삭제
3. 목록 보기
4. 종료
5. 할 일 수정
현재 할 일 수 = 2
-----
```

```
C:\Users\hyeonjin\OneDr
현재 할 일 수 = 2
-----
1
할 일을 입력하세요 (공백 없이 입력하세요): 밴드
할 일 밴드'가 저장되었습니다
-----
메뉴를 입력해주세요.
1. 할 일 추가
2. 할 일 삭제
3. 목록 보기
4. 종료
5. 할 일 수정
현재 할 일 수 = 3
-----
2
삭제할 할 일의 번호를 입력해주세요. (1부터 시작):1
1. 공부 : 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 = 2
-----
3
할 일 목록
1. 게임
2. 밴드
-----
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. 최종 테스트 스크린샷



결과 및 결론

1. 프로젝트 결과 : TODO 프로그램을 만들었다.
2. 느낀점: c언어 코딩 실력이 부족해서 아직 여러가지 구현하지 못하지만 이 강의를 통해 조금이나마 실력을 길러야겠다.
- 3.