

1.서론

7주차까지 배운 내용에 대해 실습을 해 보며 2차원 배열 반복문 조건문 등에 대해 알아보기 위해 실습을 진행하였고 목표는 TODO리스트를 만드는 것이다.

2.요구사항

사용자의 요구사항은 사용자가 할 일을 입력,삭제,출력,수정 할수 있는 프로그램의 생성이고 기능 요구사항 또한 앞에 나열한 것들과 동일한 기능을 요구하였다

3.설계 및 구현

1.기능 별 구현 사항

```
#define MAX_TASKS 10
#define CHAR_NUM 100
#include <string.h>

int main() {
    char tasks[MAX_TASKS][CHAR_NUM] = { "" }; // 할 일 목록을 저장하기 위한 2차원 배열
    int taskCount = 0; // 할 일의 수를 저장하기 위한 변수

    printf("TODO 리스트 시작! \n");

    while (1) {
        int choice = -1; // 사용자 입력 메뉴를 저장하기 위한 변수

        // 사용자에게 메뉴를 출력하고, 메뉴를 입력받기
        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("-----\n");
        scanf_s("%d", &choice);

        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
        int modifyIndex = -1;
        char ch; // 할 일 수정시 버퍼를 받기 위한 문자 변수
```

1.입력

- task[MAX_TASKS][CHAR_NUM] 10x100짜리 2차원 배열 생성
- taskCount라는 할 일의 수를 저장 하기 위한 변수 생성
- terminate라는 종료를 위한 flag 생성 하고 0을 할당
- delIndex라는 할 일 삭제를 위한 변수 생성
- changeIndex라는 할 일 수정을 위한 변수 생성
- modifyIndex라는 할 일 수정을 위한 변수 생성
- ch라는 버퍼를 받기 위한 변수 생성

```

switch (choice) {
case 1:
    printf("할 일을 입력하세요 (공백 없이 입력하세요): "); //사용자가 1을 입력하면 이런 문장이 출력이 된다.
    scanf_s("%s", tasks[taskCount], (int)sizeof(tasks[taskCount])); // 현재 task가 0 임으로 tasks 라는 2차원 배열의 1번째 줄에 입력받은 값을 저장
    printf("할 일 \"%s\"가 저장되었습니다.\n", tasks[taskCount]);
    taskCount++; //할일을 입력했으므로 taskCount가 1 추가 된다
    break;
case 2:
    // 할 일 삭제하는 코드 블록
    printf("삭제할 할 일의 번호를 입력해주세요. (1부터 시작):");
    scanf_s("%d", &delIndex); //입력받은 변수를 delIndex에 저장
    if (delIndex > taskCount || delIndex <= 0) { //delIndex라는 값이 taskCount보다 크면 아직 저장 되지 않은 일에 대한 삭제이고 0보다 작을수는 없으므로
        //저 둘중 하나라도 True가 나오면 아래의 printf문의 문장을 출력한다
        printf("삭제 범위가 벗어났습니다.\n");
    }
    else {
        printf("%d. %s : 할 일을 삭제합니다.\n", delIndex, tasks[delIndex - 1]); //delIndex 즉 삭제하고 싶은 번호가 1이라고 하면 그거는 tasks[0]에 저장되어
        //있으므로 tasks[delIndex - 1] 값을 반환한다
        strcpy_s(tasks[delIndex - 1], sizeof(tasks[delIndex - 1]), ""); // tasks[delIndex - 1] 자리에 ""인 공백을 넣는다

        for (int i = delIndex; i < taskCount + 1; i++) {
            strcpy_s(tasks[i - 1], sizeof(tasks[i]), tasks[i]); //반복문을 통해서 뒤에 있는 문자열을 한자리 앞으로 옮겨준다
        }
        taskCount -= 1; //할 일이 하나 삭제되었으므로 -1을 해준다
    }
    break;
}

```

1.설명

- case1설명 앞서 1을 입력 받게 되면 할 일을 입력하라는 문장이 출력된다
- 문자를 입력 받게 되면 task라는 배열에 저장이 된다
- 할 일을 입력 하고 배열에 저장이 되었으면 할 일이 추가 되었으므로 taskCount는 +1이 된다
- case2 설명 삭제할 번호를 입력 받고 delIndex라는 변수에 저장 해준다
- if문을 통해 delIndex가 taskCount보다 크거나 delIndex가 0보다 작거나같은경우는 범위가 벗어났다고 알려준다
- 그 이외의 경우에는 삭제하려고 하는 번호가 3 이면 task[2]에 그 문자가 저장되어있으므로 strcpy_s를 통해서 tasks[delIndex-1]부분을 공백으로 채워준다
- for문을 통해 공백 자리에 뒤에있는 문자를 한칸 앞으로 뺄겨주어야 됴므로 delIndex가 3 이라고 하면 tasks[2]에 tasks[3]을 넣어주는 과정인데 tasks[2]에는 현재 공백이 저장되어 있음으로 한칸 앞으로 뺄겨주게 된다
- 삭제가 되었으므로 taskCount의 값은 -1을 해주어 할 일의 개수가 하나 줄어들게 된다

```

case 3:
    printf("할 일 목록\n");
    for (int i = 0; i < taskCount; i++) { //taskCount가 3이라한다면
        printf("%d. %s\n", i + 1, tasks[i]); //1 ,tasks[0]이 출력 되고 ~~3,tasks[2] 까지 되어 3개의 목록을 다 보여주게 된다
    }
    printf("\n");
    break;
case 4:
    terminate = 1; //4번을 누르게 되면 terminate값이 1이 되게 되고 switch문을 빠져나가게 된다
    break;
case 5:
    printf("수정할 할 일의 번호를 입력해주세요. (1부터 시작): ");
    scanf_s("%d", &modifyIndex); //사용자가 수정하고자 하는 번호를 modifyIndex 변수에 저장
    ch = getchar();
    printf("새로운 할 일을 입력해주세요");
    scanf_s("%s", tasks[modifyIndex - 1], (int)sizeof(tasks[modifyIndex - 1])); //새로 입력받은 값을 저장해 준다
    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; //4번을 누르게 되면 break로 switch문을 빠져 나오면 여기로 오게 되는데 여기서는 종료 문장을 출력하고 while 문을 빠져나와 프로그램이 종료된다
}

```

1. 설명

- case3 할 일의 목록을 보기 위해서 이고 for문을 통해서 taskCount가 3이라고 하면 출력은 1값에다 1더해진 1이 나오고 task[i] 는 task[0]을 출력 되게 되어 쌍이 맞게 된다
- case4 4를 입력 받게 되면 terminate 값이 1로 되게 되고 switch문을 빠져 나가게 된다 빠져 나가게 되어 아래의 if문으로 가서 종료되었다는 문장을 프린트 하게 되고 while문 자체를 빠져나가 프로그램이 끝나게 된다
- case5 5를 입력 받게 되면 수정할 번호를 입력하라고 뜨고 입력 하면 modifyIndex라는 변수에 저장하게 되고 getchar를 통해 공백은 제거해 준다
- 이제 새로운 할 일을 입력 해 주라고 말하게 된다
- 위에서 수정할 번호를 5를 입력했다고 하게 되면 modifyIndex는 5가 되고 새로운 할 일의 운동을 입력 했다고 하면 현재 5번째에 입력되어있는 값은 task[4]에 저장되어있으므로 새로운 입력값을 task[modifyIndex-1]에 저장 해 준다
- 그리고 print를 통해 확인 시켜 준다
- 처음에 1 2 3 4 5 이외의 값을 입력 받으면 default로 가 잘못된 선택이라고 알려준다

4.테스트

1. 할 일 추가

```

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): c공부
할 일 c공부가 저장되었습니다

```

2. 할 일 목록

```
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. c 공부
```

3. 할 일 수정

```
5
수정할 할 일의 번호를 입력해주세요. (1부터 시작): 1
새로운 할 일을 입력해주세요. 운동
새로운 할 일이 추가되었습니다: 1. 운동
-----
```

```
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. 할 일 삭제

```
2
삭제할 할 일의 번호를 입력해주세요. (1부터 시작): 1
1. 운동 : 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
```

5. 할 일 종료

```
4
종료를 선택하셨습니다. 프로그램을 종료합니다.
```

5.결과 및 결론

1.프로젝트 결과

```
1000 리즈트 시작1
-----
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): c언어공부
할 일 c언어공부가 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
-----
5
수정할 할 일의 번호를 입력해주세요. (1부터 시작): 1
새로운 할 일을 입력해주세요.운동
새로운 할 일이 추가되었습니다: 1. 운동

-----
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. 운동 : 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
-----
4
종료를 선택하셨습니다. 프로그램을 종료합니다.

C:\Users\장예성\source\repos\Project13\x64\Debug\Project13.exe(프로세스 15800개)이(가) 종료되었습니다(코드: 0개).
이 창을 닫으려면 아무 키나 누르세요...|
```

2. 느낀점: 2차원 배열 반복문 조건문 그리고 다양한 변수들을 활용한 코드를 직접 만들어보면서 이해해 보니 하나의 개념만 가지고 문제를 풀때보다 더 코딩 실력이 향상이 되는 것 같았다.