

1. 서론

- A. 프로젝트 목적 및 배경 : TODO리스트를 보완하는 프로젝트형 실습을 위하여 진행
- B. 목표 : 사용자 요구사항에 맞는 TODO 리스트를 제작

2. 요구사항

- A. 사용자 요구사항 :
- B. 기능 요구사항
 - i. 사용자에게 작업 요청받기
 - ii. 요청 받은 작업에 따라 기능 수행
 - 1. 할 일 추가를 입력했을 경우, 사용자에게 할 일을 입력받고 저장
 - 2. 할 일 삭제를 입력했을 때, 인덱스를 입력받고 해당 할 일 삭제
 - 3. 목록 보기를 입력했을 때, 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
 - 6. 할 일이 10개로 다 찬 경우 할 일이 다 찼다고 출력하고 프로그램 종료

3. 설계 및 구현

- A. 기능 별 구현사항
 - i. 사용자에게 작업 요청받기

```
16      int choice = -1; // 사용자 입력을 저장받을 변수를 초기화함
17
18      // to do menu를 출력하고 입력받음
19      printf("-----\n");
20      printf("메뉴를 입력해주세요.\n");
21      printf("1. 할 일 추가\n2. 할 일 삭제\n3. 목록 보기\n4. 종료\n5. 할 일 수정\n");
22      printf("현재 할 일 수 = %d\n", taskCount);
23      printf("-----\n");
24      scanf_s("%d", &choice);
25
```

1. 입력

없음

2. 결과

1. 할 일 추가 2. 할 일 삭제 3. 목록 보기 4. 종료 5. 할 일 수정

3. 설명

사용자의 입력을 받기 위해 변수를 초기화 한 후 내용을 출력하고, 입력을 기다리는 내용

ii. 요청 받은 작업에 따라 기능 수행

1. 할 일 추가를 입력했을 경우, 사용자에게 할 일을 입력받고 저장

```
case 1:
    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. 입력

1

유저의 입력값(할일)

B. 결과

(유저 입력값) tasks[taskCount]에 저장

"할 일 (유저입력값) 가 저장되었습니다." 출력

taskCount++

C. 설명

유저의 할일 추가 내용을 문자열에 저장하고 taskCount 변수를 ++ 하여 파악

2. 할 일 삭제를 입력했을 때, 인덱스를 입력받고 해당 할 일 삭제

```

case 2:
    // 할 일 삭제하는 코드 블록
    printf("삭제할 할 일의 번호를 입력해주세요. (1부터 시작):");
    scanf_s("%d", &delIndex);
    if (delIndex > taskCount || delIndex <= 0) {
        printf("삭제 범위가 벗어났습니다.\n");
    }
    else {
        printf("%d. %s : 할 일을 삭제합니다.\n", 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]);
        }
        taskCount -= 1;
    }
    break;

```

A. 입력

2

(삭제할 항목 유저 입력값 = delIndex변수)

B. 결과

delIndex 변수 값이 유효하지 않으면

"삭제 범위가 벗어났습니다." 출력

delIndex값이 유효하면,

"%d. %s : 할 일을 삭제합니다" 출력 후

tasks[delIndex - 1] 의 값을 ""로 문자열 복사

반복문으로 그 이후에 존재하는 tasks 값들을 앞으로 땡겨옴

C. 설명

삭제할 항목이 정확하다면 해당 인덱스의 값을 삭제하고 후방에 위치한 값들을 앞으로 땡겨와서 재정렬한다.

3. 목록 보기를 입력했을 때, 전체 할 일 목록을 보여주기

```

case 3:
    // 할 일 목록을 출력하는 코드 블록
    printf("할 일 목록\n");
    for (int i = 0; i < taskCount; i++) {
        printf("%d. %s\n", i + 1, tasks[i]);
    }
    printf("\n");
    break;

```

A. 입력

3

B. 결과

반복문으로 할 일을 1. 2. 숫자와 함께 차례대로 출력

C. 설명

유저가 3 할 일 목록 을 선택했을 때, 할 일에 있는 내용을 반복문을 통하여 앞에 숫자를 동시에 출력하면서 출력

4. 종료를 입력했을 경우, 프로그램 종료

```
case 4:  
    // 종료하는 코드  
    terminate = 1;  
    break;
```

```
if (terminate == 1) {  
    //종료 변수 값이 1로 일치하면 프로그램을 종료  
    printf("종료를 선택하셨습니다. 프로그램을 종료합니다.\n");  
    break;  
}
```

A. 입력

4

Terminate = 1

B. 결과

"종료를 선택하셨습니다. 프로그램을 종료합니다." 를 출력

C. 설명

종료를 상징하는 변수 terminate를 1 로 바꾸고 만약 해당 변수 값이 1과 같다면, "종료를 선택하셨습니다. 프로그램을 종료합니다." 를 출력한 뒤 반복문을 탈출하여 프로그램을 종료함.

5. 할 일 수정을 입력했을 경우, 인덱스와 할 일(문자열)을 입력 받고, 해당 인덱스의 할 일 변경

```

case 5:
    //수정을 위한 코드
    printf("수정할 할 일의 번호를 입력해주세요. (1부터 시작):");
    scanf_s("%d", &changeIndex);
    //사용자에게 인덱스 번호를 받고
    ch = getchar();
    //받은 번호가, 테스크 카운트보다 같거나 적고, 0보다 클때만 동작
    if (changeIndex > 0 && changeIndex <= taskCount) {
        printf("수정할 일을 입력하세요 (공백 없이 입력하세요): ");
        //뭐라고 수정할지 입력받고 해당 위치의 기존 값으로 덮어쓰기 함
        scanf_s("%s", tasks[changeIndex - 1], (int)sizeof(tasks[changeIndex - 1]));
        //strcpy_s(tasks[changeIndex - 1], sizeof(tasks[changeIndex - 1]), tasks[changeIndex - 1]);
        printf("할 일 \"%s\"로 수정 되었습니다\n\n", tasks[changeIndex - 1]);
        break;
    }
    //아니라면 잘못된 선택이라면서 다시 복귀시킴
    else {
        printf("수정할 할 일이 없습니다. 잘못된 선택입니다.\n");
        break;
    }
}

```

A. 입력

5

할 일의 번호(changeIndex)

바꿀 일

B. 결과

"수정할 할 일의 번호를 입력해주세요. (1부터 시작):"

"수정할 일을 입력하세요 (공백 없이 입력하세요): "

"할 일 \"%s\"로 수정 되었습니다\n\n"

"수정할 할 일이 없습니다. 잘못된 선택입니다.\n"

C. 설명

수정할 일의 번호를 유저에게 입력 받은 뒤, 해당 인덱스가 taskCount보다 작거나 같고, 0보다 크다면, 어떻게 수정할지 받은 뒤 해당 내용을 덮어쓰기로 수정한다.

이후 수정이 완료되었다는 메시지를 출력한다.

만약 입력받은 인덱스 값이 일치하지 않는다면, "수정할 할 일이 없습니다. 잘못된 선택입니다" 라고 출력한다.

6. 할 일이 10개로 다 찬 경우 할 일이 다 찼다고 출력하고 프로그램 종료

```

if (taskCount == 10) {
    //taskcount가 10 일 때 모두 종료되었다는 문구와 함께 프로그램을 종료
    printf("할 일이 모두 찼습니다. 프로그램을 종료합니다.\n");
    break;
}

```

A. 입력

없음

B. 결과

taskCount 가 10일 때.

"할 일이 모두 찻습니다. 프로그램을 종료합니다."

break

C. 설명

입력받은 할 일의 수가 10개가 충족되면, 할 일이 모두 찻다는 메시지를 출력하고 프로그램을 종료한다.

4. 테스트

A. 기능 별 테스트 결과

i. 사용자에게 작업 요청받기

```
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
-----
```

ii. 요청 받은 작업에 따라 기능 수행

1. 할 일 추가를 입력했을 경우, 사용자에게 할 일을 입력받고 저장

```
-----
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
```

2. 할 일 삭제를 입력했을 때, 인덱스를 입력받고 해당 할 일 삭제

```

-----
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):2
2. c언어공부2 : 할 일을 삭제합니다.

```

3. 목록 보기를 입력했을 때, 전체 할 일 목록을 보여주기

```

-----
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. c언어공부
2. c언어공부3

```

4. 종료를 입력했을 경우, 프로그램 종료

```

-----
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. 할 일 수정을 입력했을 경우, 인덱스와 할 일(문자열)을 입력 받고, 해당 인덱스의 할 일 변경

```

3
할 일 목록
1. 111
2. 222
3. 333

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

-----
5
수정할 할 일의 번호를 입력해주세요. (1부터 시작):3
수정할 일을 입력하세요 (공백 없이 입력하세요): 555
할 일 로 수정 되었습니다

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

-----
3
할 일 목록
1. 111
2. 222
3. 555

```

6. 할 일이 10개로 다 찬 경우 할 일이 다 찼다고 출력하고 프로그램 종료

```

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

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

할 일이 모두 찼습니다. 프로그램을 종료합니다.

```

B. 최종 테스트 스크린샷


```

1
할 일을 입력하세요 (공백 없이 입력하세요): 234
할 일 234가 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

1
할 일을 입력하세요 (공백 없이 입력하세요): 3456
할 일 3456가 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

3
할 일 목록
1. 123
2. 234
3. 3456

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

5
수정할 할 일의 번호를 입력해주세요. (1부터 시작):3
수정할 일을 입력하세요 (공백 없이 입력하세요): 654321
할 일 로 수정 되었습니다

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

4
종료를 선택하셨습니다. 프로그램을 종료합니다.

C:\Users\A\\Desktop\c 프로젝트 실습\64\Debug\c 프로젝트 실습.exe(프로세스 11552개)이(가) 종료되었습니다(코드: 0개).
이 창을 닫으려면 아무 키나 누르세요...

```

5. 결과 및 결론

- A. 프로젝트 결과 : 할 일을 추가하고, 삭제하고, 목록을 제공하며, 수정도 가능한 할 일 관리 프로그램을 보완하고 수정하면서, c언어의 이해를 높일 수 있었다.
- B. 느낀점 : 충분한 시간이 주어진다면 해낼 수 있을 것 같은데, 자꾸 시간에 쫓기는 압박에 빠져 할 수 있던것도 못하는 것 같아 아쉽습니다. 추후 개인 프로젝트때는 충분한 시간을 할애하여 완벽한 프로그램으로 제작해보겠습니다.