

[TODO 리스트 프로그램 보고서]

이름 : 정지우

학번 : 181155

학과 : 통계학과

강의명 : C 프로그래밍및실습

<목차>

1. 서론

1) 프로젝트 목적 및 배경

2) 목표

2. 요구사항

1) 사용자 요구사항

2) 기능 요구사항

3. 설계 및 구현

1)기능별 구현사항

4.테스트

1)기능별 테스트 결과

2)최종 테스트 스크린샷

5.결과 및 결론

1)프로젝트 결과

2)느낀점

1.서론

1) 프로젝트 목적 및 배경

:C프로그래밍 및 실습 강의시간에 2주차부터 7주차까지 배운 반복문,조건문,배열,함수,문자열 등을 활용하여 프로그램을 만들어보는 실습을 위해 할 일을 입력하면 저장해주고 수정,삭제 등이 가능한 TODO리스트 프로그램을 만들어보았다.

2) 목표

:프로그램에 할 일을 입력하여 저장하고 수정, 삭제, 할 일 목록을 출력 등을 해주는 TODO 리스트 프로그램 만든다.

2.요구사항

1) 사용자 요구사항

: 프로그램에 할 일을 입력하여 저장하고 수정을 원하는 할 일을 다른 일로 수정하며, 원하는 목록을 삭제, 할 일 목록을 출력 등을 해주는 프로그램

2) 기능 요구사항

:할 일을 입력하면 할 일 목록에 저장해주는 입력기능, 저장된 할 일 목록 중에서 원하는 목록을 수정할 수 있게 해주는 수정기능, 저장된 할 일 목록을 전부 보여주는 출력기능, 원하는 할 일 목록을 없애 주는 삭제 기능, 제한된 입력가능 개수를 초과하면 자동으로 프로그램을 종료해주는 기능 등이 요구된다.

3.설계 및 구현

1) 기능 별 구현 사항

<입력 기능-할 일 추가하기>

```
case 1:
    printf("할 일을 입력하세요 (공백 없이 입력하세요): ");
    //할일을 2차원 배열 tasks에 입력받는 scanf_s문
    scanf_s("%s", tasks[taskCount], (int)sizeof(tasks[taskCount]));
    printf("할 일 \"%s\"가 저장되었습니다\n\n", tasks[taskCount]);
    //할 일 목록이 추가되었으므로 할 일 갯수를 저장하는
    //taskCount변수에 1을 더한다.
    taskCount++;
    break;
```

[1]코드블록/함수 스크린샷(좌)

[2]입력

-taskCount는 현재 작업수를 의미한다.

-tasks는 할 일 목록이 저장될 2차원 배열을 의미한다.

[3]반환값

-함수가 아니므로 반환값은 없다.

[4]결과

-scanf_s문을 통해 입력된 할 일이 2차원 배열 tasks에 추가된다.

[5]설명

-프로그램이 시작되면 1번을 입력하면 할 일을 입력하라고 출력이 된다.

-그 후에 할 일을 입력하라는 문구가 출력된다.

-할 일을 입력하면 2차원 배열 tasks에 할 일이 추가되고 다시 메뉴를 고르는 것으로 넘어간다.

<삭제기능-할 일 삭제하기>

case 2:

```
// 할 일 삭제하는 코드 블록
printf("삭제할 할 일의 번호를 입력해주세요. (1부터 시작):");
scanf_s("%d", &delIndex); //삭제할 할일의 인덱스를 입력받는 scanf_s문

//만약 입력한 삭제할 일의 번호가 저장된 할 일의 갯수인
//taskCount보다 크면 삭제범위가 벗어났음을 알려주는 조건문 작성
if (delIndex > taskCount || delIndex <= 0) {
    printf("삭제 범위가 벗어났습니다.\n");
}
else {
    printf("%d. %s : 할 일을 삭제합니다.\n", delIndex, tasks[delIndex - 1]);

    // 배열에 문자 배열인 문자열의 대입이 불가능하므로
    //문자열 복사 함수strcpy_s를 이용해서 삭제
    strcpy_s(tasks[delIndex - 1], sizeof(tasks[delIndex - 1]), "");

    // 앞에서 할 일 삭제 후 뒤에 있는 할 일을 앞으로 옮겨서 빈 칸 채우기
    for (int i = delIndex; i < taskCount + 1; i++) {
        strcpy_s(tasks[i - 1], sizeof(tasks[i]), tasks[i]);
    }
    taskCount -= 1;
}
break;
```

[1]코드블록/함수 스크린샷(좌)

[2]입력

-taskCount는 현재 입력된 작업 수를 의미하는 변수이다.

-tasks는 입력된 할 일 목록들이 저장된 2차원 배열이다.

-delIndex는 삭제하고 싶은 인덱스의 번호를 입력받아 저장한 변수이다.

[3]반환값

-함수가 아니므로 반환값이 존재하지 않는다.

[4]결과

-기존 tasks배열이 입력한 인덱스의 할 일이 삭제된 2차원 배열 tasks로 바뀐다.

[5]설명

-메뉴 선택 화면에서 2번을 누르면 삭제하고 싶은 일의 인덱스를 입력받는 문구가 출력된다.

-삭제하고 싶은 할 일의 인덱스를 입력받는다.

-입력받은 인덱스에서 -1을 한 인덱스의 할 일이 배열에서 제거 된다.

-제거한 후 뒤에 있던 할 일들을 앞으로 당겨와서 중간에 빈 행이 없도록 한다.

-삭제가 끝나면 switch문을 탈출해서 다시 메뉴를 고르는 것으로 넘어간다.

<출력기능-할 일 목록 보여주기>

case 3:

```
printf("할 일 목록\n");
//반복문을 통해 tasks배열을 전부 출력
for (int i = 0; i < taskCount; i++) {
    printf("%d. %s\n", i + 1, tasks[i]);
}
printf("\n");
break;
```

[1]코드블록/함수 스크린샷(좌)

[2]입력

-taskCount는 현재 입력된 할 일의 개수를 의미하는 변수이다.

-tasks는 입력된 할 일 목록이 저장된 2차원 배열이다.

[3]반환값

-함수가 아니므로 반환값이 존재하지 않는다.

[4]결과

-현재 2차원 배열 tasks에 저장된 할 일 목록을 전부 출력하여 사용자에게 보여다.

[5]설명

-메뉴선택 화면에서 3번을 누르면 현재 tasks에 저장된 모든 할 일 목록이 출력된다.

-출력이 끝나면 switch문을 탈출해서 다시 메뉴를 고르는 것으로 넘어간다.

<종료기능-TODO리스트 종료>

case 4:

```
//종료를 위한 변수 terminate에 1을 입력하여
//뒤에 나오는 조건문을 이용하여 프로그램 종료
terminate = 1;
break;

//terminate변수에 1이 입력되면 프로그램을 종료시키는 조건문
if (terminate == 1) {
    printf("종료를 선택하셨습니다. 프로그램을 종료합니다.\n");
    break;
```

[1]코드블록/함수 스크린샷(좌)

[2]입력

-terminate는 종료를 위한 변수로 초기값은 0으로 되어있으며 1이 입력되면 조건문에 의해 프로그램이 종료된다.

[3]반환값

-함수가 아니므로 반환값은 존재하지 않는다.

[4]결과

-terminate에 1이 입력되고 terminate가 1이 되면 while문을 break하는 조건문에 의해 프로그램이 종료된다.

[5]설명

-메뉴선택 화면에서 4번을 누르면 종료를 위한 변수인 terminate 변수에 1이 저장된다.

-terminate변수에 1이 저장되면. 조건문에 의해 프로그램이 종료된다.

<수정기능-할 일 목록 수정>

case 5:

```
printf("수정할 할 일의 번호를 입력해주세요. (1부터 시작): ");
scanf_s("%d", &changeIndex); //수정할 할 일의 인덱스를 입력받음
//입력된 버퍼를 getchar()를 이용해 ch에 저장하는 방식으로
//다음 코드가 작동하는데 방해되지 않도록함
ch = getchar();
printf("수정할 내용을 입력해주세요 : ");
//수정할 내용을 입력받는 scanf_s문
scanf_s("%s", tasks[changeIndex - 1], (int)sizeof(tasks[taskCount]));
printf("할일이 수정되었습니다.");
printf("\n");
break;
```

[1]코드블록/함수 스크린샷(좌)

[2]입력

-changeIndex는 바꾸고 싶은 할 일의 인덱스 번호를 입력받아 저장하는 변수이다.

-taskCount는 현재 입력된 할 일의 개수를 의미하는 변수이다.

-tasks는 입력된 할 일 목록이 저장된 2차원 배열이다.

[3]반환값

-함수가 아니므로 반환값은 존재하지 않는다

[4]결과

-사용자가 입력한 인덱스 번호의 할 일이 사용자가 입력한 할 일로 바뀌어서 2차원 배열tasks에 저장된다.

[5]설명

-메뉴선택 화면에서 5번을 누르면 현재 tasks에 저장된 할 일 중에서 수정하고 싶은 할 일의 인덱스를 입력받는 문구가 출력된다.

-수정하고 싶은 할 일의 인덱스를 입력하면 할 일을 무엇으로 수정할지 입력받는 문구가 출력된다.

-기존 할 일을 수정할 할 일까지 입력되면 입력된 인덱스에서 -1을 한 인덱스에 있는 할 일이 입력한 할 일로 수정된다.

<자동 종료 기능>

```
// TODO: 할 일이 다 찼는지 체크하는 코드=>다 차면 프로그램 종료
if (taskCount == MAX_TASKS) {
    printf("할일이 다 찼습니다.\n");
    break;
}
```

[1]코드블록/함수 스크린샷(좌)

[2]입력

-MAX_TASKS는 최대 입력가능한 할 일의 개수를 의미하는 변수이다.

-taskCount는 현재 입력된 할 일의 개수를 의미하는 변수이다.

[3]반환값

-함수가 아니므로 반환값은 존재하지 않는다

[4]결과

-사용자가 입력한 할 일의 개수가 최대 입력가능한 개수에 도달하면 프로그램이 자동 종료된다.

[5]설명

-1번 입력기능을 통해 할 일을 계속해서 저장하다가 최대 입력가능 개수인 10개에 도달하게 되면 프로그램이 자동으로 즉시 종료된다.

4.테스트

1)기능별 테스트 결과

<입력기능-할 일 추가하기>

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): 영화감상
할 일 영화감상이 저장되었습니다

<삭제기능>

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. 산책 : 할 일을 삭제합니다.

<출력기능>

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. 영화감상
2. 영화감상

<종료기능>

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.

<수정기능>

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

수정할 내용을 입력해주세요 : C프로그래밍과제

할일이 수정되었습니다.

<자동 종료 기능>

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): 10

할 일 10가 저장되었습니다

할일이 다 찼습니다.

2.최종 테스트 스크린샷

<1~5번 기능 테스트>

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): 산책
할 일 산책가 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

3

할 일 목록

1. 산책

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
수정할 내용을 입력해주세요 : c프로그래밍과제
할일이 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

2

삭제할 할 일의 번호를 입력해주세요. (1부터 시작):1

1. c프로그래밍과제 : 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.

<자동 종료 기능 테스트>

메뉴를 입력해주세요.

1. 할 일 추가
2. 할 일 삭제
3. 목록 보기
4. 종료

5. 할 일 수정

현재 할 일 수 = 7

1

할 일을 입력하세요 (공백 없이 입력하세요): 8

할 일 8가 저장되었습니다

메뉴를 입력해주세요.

1. 할 일 추가
2. 할 일 삭제
3. 목록 보기
4. 종료

5. 할 일 수정

현재 할 일 수 = 8

1

할 일을 입력하세요 (공백 없이 입력하세요): 9

할 일 9가 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

1

할 일을 입력하세요 (공백 없이 입력하세요): 10

할 일 10가 저장되었습니다

할 일이 다 찼습니다.

5.결과 및 결론

1)프로젝트 결과

:할 일을 입력하여 저장하고 수정, 삭제, 목록 출력 등으로 관리가 가능한 할 일 관리 프로그램 TODO리스트를 만들었다. 세부적인 기능으로 할 일 입력, 입력된 할 일 삭제 및 수정, 목록 출력, 프로그램 수동 종료, 자동종료 등의 기능이 있으며 최대 입력가능 할 일의 개수는 10개이며 10개를 입력할 시 할 일이 다 찼다는 메시지와 함께 자동으로 종료된다.

2)느낀 점:

: C프로그래밍 및 실습 강의시간에 배운 C언어의 기초적인 함수, 배열, 조건문, 반복문 등을 이용하여 직접 TODO리스트 프로그램을 만들면서 이론으로 배웠던 내용들이 확실히 이해가 되었다. 또한 프로그래밍이라고 하면 무조건적으로 복잡한 코드들과 복잡한 이론들이 필요할거라는 선입견이 있었는데 기초적인 함수와 조건문, 반복문들만 가지고도 유용한 프로그램이 만들어지는 것을 보고 그런 선입견이 사라지는 계기가 되었다.