

C프로그래밍및실습

8주차 프로젝트형

실습

프로젝트 보고서

제출일자: 2023-10-27

제출자명: 최지원

제출자학번: 224571

1. 서론

1) 프로젝트 목적 및 배경

7주차까지 배운 내용에 대한 실습을 위해 진행, 할 일 관리를 유용하고 편리하기 위해 프로그램이 필요함

2) 목표

TODO 리스트 만들기

2. 요구사항

1) 사용자 요구사항:

사용자가 할 일을 입력, 삭제, 출력할 수 있는 프로그램

2) 기능 요구사항

- 할 일 추가
- 할 일 삭제
- 할 일 목록 보기
- 할 일 수정
- 프로그램 종료

3. 설계 및 구현

1) 기능 별 구현 사항

■ 할 일 추가 블록

1. 코드블록 스크린샷

```
switch (choice) {  
case 1: //1선택시  
    printf("할 일을 입력하세요 (공백 없이 입력하세요): ");  
    scanf_s("%s", tasks[taskCount], (int)sizeof(tasks[taskCount])); //tasks배열에 할일 저장  
    printf("할 일 \"%s\"가 저장되었습니다\n\n", tasks[taskCount]);  
    taskCount++; //1 선택할때마다 배열 index 증가  
    break;
```

2. 입력 (블록/함수에 입력되는 변수, 값들과 설명)

- taskCount = 현재 작업 수
- tasks = 할일 목록 저장 2차원 배열
- choice = 기능 선택을 위해 사용자가 입력하는 변수(기능번호)

3. 반환값 (함수의 경우 작성)

- 함수가 아니므로 없음

4. 결과 (블록/함수가 종료된 결과)

- 할 일 <사용자가 추가한 할 일>가 저장되었습니다 문구 출력

5. 설명 (코드 내 작동 순서, 내용 등 추가 설명)

- while문에서 메뉴 1번 입력 시 실행
- 사용자에게 할 일을 입력 받는다
- 할일을 tasks배열 1열에 추가한다
- 사용자가 입력한 할 일이 저장되었다는 문구를 표시한다
- taskCount를 증가시켜 다음 입력을 받을 때 다음 열에 추가하도록 한다

■ 할 일 삭제 블록

1. 코드블록 스크린샷

```
break;
case 2:
    // 할 일 삭제하는 코드 블록
    printf("삭제할 할 일의 번호를 입력해주세요. (1부터 시작): ");
    scanf_s("%d", &delIndex);
    if (delIndex > taskCount || delIndex <= 0) { //할일 번호 잘못 입력시 실행
        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;
```

2. 입력 (블록/함수에 입력되는 변수, 값들과 설명)

- taskCount = 현재 작업 수
- tasks = 할일 목록 저장 2차원 배열
- delIndex = 할 일 삭제를 위한 인덱스 번호

3. 반환값 (함수의 경우 작성)

- 함수가 아니므로 없음

4. 결과 (블록/함수가 종료된 결과)

- <삭제할 할 일 번호>. <할 일> : 할 일을 삭제합니다 문구 표시

5. 설명 (코드 내 작동 순서, 내용 등 추가 설명)

- while문에서 메뉴 2번 입력 시 실행
- 사용자에게 삭제할 할 일 번호를 입력 받는다

- 삭제할 할 일 번호가 기존 배열에 있던 열 보다 숫자가 크면 if문 실행
- 크지 않으면 할 일 삭제 문구 표시
- strcpy_s를 사용하여 tasks[delIndex-1] 배열을 ""로 복사
- ➔ 배열간 대입이 불가능하기 때문에 strcpy_s 를 사용
- 할 일 삭제 후에 뒤에 있는 할 일을 앞으로 가져오기
- 할 일을 앞으로 가져왔기 때문에 taskCount를 -1씩 줄여 다음 할 일 입력 시 다음 인덱스에 입력이 되도록 함

■ 할 일 목록 블록

1. 코드블록 스크린샷

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

2. 입력 (블록/함수에 입력되는 변수, 값들과 설명)

- taskCount = 현재 작업 수
- tasks = 할일 목록 저장 2차원 배열

3. 반환값 (함수의 경우 작성)

- 함수가 아니므로 없음

4. 결과 (블록/함수가 종료된 결과)

- 할 일 목록이 출력됨

5. 설명 (코드 내 작동 순서, 내용 등 추가 설명)

- main문에서 메뉴 3번 입력 시 실행
- for 반복문을 사용하여 tasks배열에 있는 목록들을 출력함

■ 할 일 종료 블록

1. 코드블록 스크린샷

```
case 4: //4 선택시 terminate 변수를 1로 저장 -> 종료
{
    terminate = 1;
    break;
}
```

2. 입력 (블록/함수에 입력되는 변수, 값들과 설명)

- terminate = 프로그램 종료를 위한 변수

3. 반환값 (함수의 경우 작성)

- 함수가 아니므로 없음

4. 결과 (블록/함수가 종료된 결과)

- 프로그램이 종료됨

5. 설명 (코드 내 작동 순서, 내용 등 추가 설명)

- while문에서 메뉴 4번 입력 시 실행

➔ while문에서 terminate가 1이 될 시 프로그램을 종료하는 if문이 있어서 메뉴 4를 선택 시 프로그램은 종료된다

■ 할 일 수정 블록

1. 코드블록 스크린샷

```

break;
case 5: //할일 수정
    printf("수정할 할 일의 번호를 입력해주세요. (1부터 시작):");
    scanf_s("%d", &changeIndex); //수정할 index 입력받음
    if (changeIndex > taskCount || changeIndex <= 0) { //잘못된 번호 입력 시 실행
        printf("수정 범위가 벗어났습니다.\n");
    }
    else { //할일을 수정하기 위해 기존 배열에 있던 내용 strcpy_s를 이용하여 삭제
        printf("%d. %s : 할 일을 수정합니다.\n", changeIndex, tasks[changeIndex - 1]);
        strcpy_s(tasks[changeIndex - 1], sizeof(tasks[changeIndex - 1]), "");
        printf("수정할 내용을 입력하세요\n"); //scanf_s를 이용해서 삭제했던 배열에 다시 내용 넣어줌
        scanf_s("%s", tasks[changeIndex - 1], (int)sizeof(tasks[changeIndex - 1]));
        printf("%d번 할 일이 %s 으로 수정되었습니다.\n", changeIndex, tasks[changeIndex - 1]);

        break;
    }
}

```

2. 입력 (블록/함수에 입력되는 변수, 값들과 설명)

- taskCount = 현재 작업 수
- tasks = 할일 목록 저장 2차원 배열
- changeIndex = 할 일 수정을 위한 인덱스 번호

3. 반환값 (함수의 경우 작성)

- 함수가 아니므로 없음

4. 결과 (블록/함수가 종료된 결과)

- <할 일 번호>번 할 일이 <수정한 할 일>으로 수정되었습니다 출력

5. 설명 (코드 내 작동 순서, 내용 등 추가 설명)

- while문에서 메뉴 5번 입력 시 실행
- 사용자에게 수정할 할 일 번호를 입력 받는다
- 수정할 할 일 번호가 기존 배열에 있던 열 보다 숫자가 크면 if문 실행
- 크지 않으면 else문 실행
- strcpy_s를 사용하여 tasks[changeIndex-1] 배열을 ""로 복사
- ➔ 배열간 대입이 불가능하기 때문에 strcpy_s 를 사용
- 복사 후 사용자에게 할 일을 입력 받고 tasks[changeIndex-1]에 저장

■ if문: 할 일 종료 블록

1. 코드블록 스크린샷

```
if (terminate == 1) { //종료 코드 case 4 선택시 종료됨
    printf("종료를 선택하셨습니다. 프로그램을 종료합니다.\n");
    break;
}
```

2. 입력 (블록/함수에 입력되는 변수, 값들과 설명)

- terminate = 프로그램 종료를 위한 변수

3. 반환값 (함수의 경우 작성)

- 함수가 아니므로 없음

4. 결과 (블록/함수가 종료된 결과)

- 프로그램이 종료됨

5. 설명 (코드 내 작동 순서, 내용 등 추가 설명)

- switch문에서 case 4번 실행시 terminate 값이 1이 되면서 if문이 실행됨
- break가 실행되어 while문이 종료되어 프로그램이 종료됨

■ 할 일 개수 제한 블록

1. 코드블록 스크린샷

```
// TODO: 할 일이 다 있는지 체크하는 코드 작성
if (taskCount >= 10)
{
    printf("할 일이 다 찼습니다");
    break;
}
```

2. 입력 (블록/함수에 입력되는 변수, 값들과 설명)

- taskCount = 현재 작업 수

3. 반환값 (함수의 경우 작성)

- 함수가 아니므로 없음

4. 결과 (블록/함수가 종료된 결과)

- '할 일이 다 찼습니다' 문구 출력

5. 설명 (코드 내 작동 순서, 내용 등 추가 설명)

- taskCount가 10이상이 될 시 실행

4. 테스트

1) 기능 별 테스트 결과

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): 공부

할 일 공부가 저장되었습니다

2. 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 = 1

2

삭제할 할 일의 번호를 입력해주세요. (1부터 시작):1

1. 공부 : 할 일을 삭제합니다.

3. 목록 보기

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. 공부
2. 밥먹기

4. 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 수정

```
-----
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):2
2. 숨쉬기 : 할 일을 수정합니다.
수정할 내용을 입력하세요
잠자기
2번 할 일이 잠자기 으로 수정되었습니다.
```

5. 할 일 개수 제한

```
-----
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): qwer
할 일 qwer가 저장되었습니다

할 일이 다 찹니다
```

(10번째 할 일 입력 후 프로그램이 종료됨)

2) 최종 테스트 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): 공부
할 일 공부가 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): 숨쉬기
할 일 숨쉬기가 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

수정할 할 일의 번호를 입력해주세요. (1부터 시작): 2

2. 숨쉬기 : 할 일을 수정합니다.

수정할 내용을 입력하세요

잠자기

2번 할 일이 잠자기 으로 수정되었습니다.

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): 가나다
할 일 가나다가 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
-

1
할 일을 입력하세요 (공백 없이 입력하세요): 라마바
할 일 라마바가 저장되었습니다

메뉴를 입력해주세요.

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

1
할 일을 입력하세요 (공백 없이 입력하세요): 사아자
할 일 사아자가 저장되었습니다

메뉴를 입력해주세요.

1. 할 일 추가
2. 할 일 삭제
3. 목록 보기
4. 종료

5. 할 일 수정

현재 할 일 수 = 5

1

할 일을 입력하세요 (공백 없이 입력하세요): 차카타
할 일 차카타가 저장되었습니다

메뉴를 입력해주세요.

1. 할 일 추가
2. 할 일 삭제
3. 목록 보기
4. 종료

5. 할 일 수정

현재 할 일 수 = 6

1

할 일을 입력하세요 (공백 없이 입력하세요): 파하
할 일 파하가 저장되었습니다

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): abc
할 일 abc가 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

```

1
할 일을 입력하세요 (공백 없이 입력하세요): cde
할 일 cde가 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): qwer
할 일 qwer가 저장되었습니다

할 일이 다 찼습니다
C:\Users\binwo\source\repos\week8\x64\Debug\week8.exe(프로세스 5488개)이(가) 종료되었습니다(코드: 0개).
이 창을 닫으려면 아무 키나 누르세요...

```

5. 결과 및 결론

1) 프로젝트 결과

할 일 추가, 수정, 삭제가 되고 할일 목록을 볼 수 있는 투두 관리프로그램을 만들었다.

2) 느낀 점

7주차까지 배운 내용에 대해서 종합적으로 실습을 한 것 같아 배운 내용을 한 번 복습한 것 같습니다. 2차원 배열 사용에 대해서 더 잘 이해가 된 것 같습니다