

C프로그래밍및실습

로그인정보 저장

프로그램 개발

최종 보고서

제출일자: 2023.12.24.(일)

제출자명: 유희원

제출자학번: 231583

1. 프로젝트 목표

1) 배경 및 필요성

인터넷과 온라인 시장이 확대되면서 여러 사이트별로 회원가입을 통한 서비스 사용이 이루어지고 있음. 하지만 사이트에 따라 자신이 평소 쓰던 아이디를 사용하기 어려운 경우가 생기기도 함. 특히 사이트별로 비밀번호 생성 규칙이 달라 같은 비밀번호를 사용하기 어렵기도 하며, 개인정보 보호하기 위해 사이트별로 모두 다 다른 비밀번호를 사용함. 이에 따라 많은 로그인 정보들(ID, PW 등)이 생기고 이를 다 기억하는 데에는 한계가 발생하고, 자주 사용하지 않는 사이트일 수록 로그인 정보를 찾아내는 데에 많이 시간이 소요됨. 또한 매번 다른 비밀번호를 생성하는 것도 번거로우며 이를 기억하기도 어려움. 이 문제를 해결하기 위해 사이트별 로그인 정보를 저장하는 프로그램이 필요함

2) 프로젝트 목표

사용자에게 사이트와 로그인 정보를 입력 받고 이를 저장하여, 사용자가 원하는 사이트의 로그인 정보를 쉽게 찾아볼 수 있도록 프로그램을 만드는 것을 목표로 함.

3) 차별점

기존의 프로그램들은 로그인 정보를 그저 저장하고 보여주는 기능만 함. 이는 사용자의 정보에만 의지하는 수동적인 프로그램임. 이에 반해 우리는 저장과 확인 기능뿐만아니라 비밀번호 추천 기능을 통해 개인정보 보호에 보다 능동적이라는 차별점이 있음.

2. 기능 계획

1) 기능 1: 사용자의 로그인 정보 저장

- 사용자에게 사이트와 ID, PW를 입력 받고 이를 저장한다.

2) 기능 2: 로그인 정보 출력

- 사용자가 원하는 로그인 정보를 출력한다.

(1) 세부 기능 1: 사이트 목록 출력

- 사용자가 저장해 놓은 사이트의 목록을 보여준다.

(2) 세부 기능 2: 로그인 정보(ID, PW) 출력

- 1. 저장되어 있는 전체 사이트의 로그인 정보를 제공한다
- 2. 사용자가 원하는 사이트를 지정하면 그 사이트의 로그인 정보를 제공한다

3) 기능 3: 로그인 정보 수정

- 사용자가 원하는 로그인 정보를 수정한다.

(1) 세부 기능 1: 로그인 정보 삭제

- 사용자가 지정한 사이트의 로그인 정보를 삭제한다

(2) 세부 기능 2: 비밀번호 변경

- 사용자가 지정한 사이트의 비밀번호를 다시 입력 받아 변경한다

4) 기능 4: 추천 비밀번호

- 보안에 강한 비밀번호를 생성하여 추천해준다.

(1) 세부 기능 1: 비밀번호 추천

- 알파벳 소문자, 대문자, 숫자, 특수문자로 조합된 비밀번호를 추천해준다

(2) 세부 기능 2: 추천 비밀번호 저장

- 추천받은 비밀번호를 바로 저장해준다

- 1. 새로운 로그인 정보에 저장

- 2. 기존 로그인 정보에 저장

5) 기능 5: 프로그램 종료

- 프로그램을 종료한다

3. 기능 구현

(1) 기능1 로그인 정보 저장

- 입력: choice = 사용자가 선택한 메인 기능 메뉴 번호를 저장하는 변수

site_count = 저장된 사이트의 수를 저장하는 변수

login_info = 사용자가 입력한 로그인 정보를 저장하는 구조체

site_name = 사이트명을 저장하는 구조체 변수

id = 아이디를 저장하는 구조체 변수

password = 비밀번호를 저장하는 구조체 변수

InitializeSite = 사이트명을 구조체 변수에 저장하는 함수

Initializeld = 아이디를 구조체 변수에 저장하는 함수

InitializePassword = 비밀번호를 구조체 변수에 저장하는 함수

temp = 사용자가 입력한 정보를 임시 저장하는 변수

- 출력: login_info[site_count].site_name에 입력받은 사이트명 저장

login_info[site_count].id에 입력받은 아이디 저장

login_info[site_count].password에 입력받은 비밀번호 저장

site_count + 1

- 설명: 1. 사용자에게 사이트명, 아이디, 비밀번호를 입력 받는다
2. 각각의 데이터를 구조체 변수에 저장하기 위한 함수를 호출한다.

3. 각각의 데이터를 임시변수에 저장하고 동적할당 후 각 구조체 변수에 저장한다

4. 사이트 수를 저장하는 변수의 값을 1 증가한다

5. 로그인 정보가 저장되었다는 메시지를 출력한다

- 적용된 배운 내용: 조건문(if), 함수, 문자열, 포인터배열, 동적할당, 구조체, 헤더파일

- 코드 스크린샷

```
if (choice == 1) {
    /*기능 1. 로그인 정보 저장*/
    /*기능 1.1 사용자로부터 로그인 정보 입력 받기*/
    InitializeSite(&login_info[site_count]);
    InitializeId(&login_info[site_count]);
    InitializePassword(&login_info[site_count]);
    site_count++;

    printf("-- 로그인 정보가 저장 되었습니다 --\n");
}
```

```
// 사이트명을 구조체 변수에 저장하는 함수 정의
void InitializeSite(LOGIN* login_info) {
    char temp[100]; // 입력받은 값 임시 저장 변수
    printf("    사이트: ");
    scanf_s("%s", temp, (int)sizeof(temp));

    // 동적할당 & 구조체에 저장
    login_info->site_name = (char*)malloc((strlen(temp) + 1) * sizeof(char));
    strcpy_s(login_info->site_name, strlen(temp) + 1, temp);
}

// 아이디를 구조체 변수에 저장하는 함수 정의
void InitializeId(LOGIN* login_info) {
    char temp[100];
    printf("    아이디: ");
    scanf_s("%s", temp, (int)sizeof(temp));
    login_info->id = (char*)malloc((strlen(temp) + 1) * sizeof(char));
    strcpy_s(login_info->id, strlen(temp) + 1, temp);
}

// 비밀번호를 구조체 변수에 저장하는 함수 정의
void InitializePassword(LOGIN* login_info) {
    char temp[100];
    printf("    비밀번호: ");
    scanf_s("%s", temp, (int)sizeof(temp));
    login_info->password = (char*)malloc((strlen(temp) + 1) * sizeof(char));
    strcpy_s(login_info->password, strlen(temp) + 1, temp);
}
```

(2) 기능2 로그인 정보 출력 - 기능2.1 저장된 사이트 목록 출력

- 입력: choice = 사용자가 선택한 메뉴 번호를 저장하는 변수

site_count = 저장된 사이트의 수를 저장하는 변수

login_info = 사용자가 입력한 로그인 정보를 저장하는 구조체

site_name = 사이트명을 저장하는 구조체 변수

PrintSiteList = 전체 사이트 목록을 출력하는 함수

- 출력: 없음

- 설명: 1. 저장되어 있는 정보가 있는지 확인

2. 저장된 정보가 없을 경우(site_count=0) 오류 문구 출력

3. 저장된 정보가 있을 경우 저장된 사이트의 목록을 순서대로 출력

- 적용된 배운 내용: 조건문(if), 반복문(for), 함수, 포인터 배열, 구조체, 헤더파일

- 코드 스크린샷

```
else if (choice == 2) {
    /*기능 2. 로그인 정보 출력*/
    if (site_count == 0) {
        // 저장된 정보가 있는지 확인. 없을 경우 오류 문구 출력
        printf("저장된 로그인 정보가 없습니다. 로그인 정보를 추가하세요.\n");
    } else {
        /*기능 2.1 저장된 사이트 목록 출력*/
        PrintSiteList(login_info, site_count);
    }
}
```

```
/*전체 사이트 목록을 출력하는 함수 정의*/
void PrintSiteList(LOGIN* p_login_info, int site_count) {
    printf("-- 사이트 목록 (총 %d) --\n", site_count);
    for (int i = 0; i < site_count; i++) {
        printf("  %d. %s\n", i + 1, p_login_info[i].site_name);
    }
    printf("-----\n");
}
```

(3) 기능2 로그인 정보 출력 - 기능2.2 로그인 정보 출력

- 입력: choice_2 = 로그인 정보 출력 메뉴의 선택 번호를 저장하는 변수

PrintLoginInfo = 로그인 정보 출력하는 함수

login_info = 사용자가 입력한 로그인 정보를 저장하는 구조체

site_count = 저장된 사이트의 수를 저장하는 변수

site_choice = 로그인 정보를 출력할 사이트의 번호를 저장하는 변수

site_name = 사이트명을 저장하는 구조체 변수

id = 아이디를 저장하는 구조체 변수

password = 비밀번호를 저장하는 구조체 변수

PrintErrorMessage = 잘못된 번호 입력시 에러 메시지 출력하는 함수

- 출력: 없음
- 설명: 1. 로그인 정보 출력 기능의 메뉴 출력 및 사용자의 선택
 2. 1번 선택 시 전체 사이트의 로그인 정보를 순서대로 출력 및 메인 기능 선택 화면으로 이동
 3. 2번 선택 시 사용자가 지정한 사이트의 로그인 정보를 출력
 4. 3번(뒤로가기) 선택 시 메인 메뉴로 이동
 5. 1,2,3 이외의 번호 선택 시 에러메시지 출력
- 적용된 배운 내용: 조건문(if), 반복문(while, for), 함수, 포인터 배열, 구조체, 헤더파일
- 코드 스크린샷

```
/*기능2.2 로그인 정보 출력*/
while (1) {
    int choice_2 = 0;
    printf("-----\n");
    printf("원하시는 기능을 선택하세요.\n");
    printf(
        "1. 전체 사이트 로그인 정보\n2. 특정 사이트의 로그인 정보\n3. "
        "뒤로 "
        "가기\n");
    printf("-----\n");
    printf("번호: ");
    scanf_s("%d", &choice_2);

    if (choice_2 == 1) {
        /*기능 2.2.1 전체 사이트의 로그인 정보 출력*/
        PrintLoginInfo(login_info, site_count, 0);
        break;
    } else if (choice_2 == 2) {
        /*기능 2.2.2 특정 사이트의 로그인 정보 출력*/
        int site_choice;
        printf("원하시는 사이트의 번호를 입력하세요: ");
        scanf_s("%d", &site_choice);
        printf("--- %d번 로그인 정보 ---\n", site_choice);
        PrintLoginInfo(login_info, site_count, site_choice);
    } else if (choice_2 == 3) {
        // 메인 기능 선택지로 이동
        break;
    } else {
        // 잘못된 번호 입력시 오류 메세지 출력
        PrintErrorMessage(3);
    }
}
```

```

/*로그인 정보를 출력하는 함수 정의*/
void PrintLoginInfo(LOGIN* p_login_info, int site_count, int index) {
    if (index == 0) {
        // 전체 로그인 정보 출력
        printf("--- 전체 로그인 정보 ---\n");
        for (int i = 0; i < site_count; i++) {
            printf("%d. 사이트: %s 아이디: %s 비밀번호: %s\n", i + 1,
                p_login_info[i].site_name, p_login_info[i].id,
                p_login_info[i].password);
        }
        printf("-----\n");
    } else {
        // 특정 사이트 로그인 정보 출력
        printf("사이트: %s\n아이디: %s\n비밀번호: %s\n",
            p_login_info[index - 1].site_name, p_login_info[index - 1].id,
            p_login_info[index - 1].password);
        printf("-----\n");
    }
}

```

```

/*잘못된 번호 입력시 에러 메시지 출력 함수 정의*/
void PrintErrorMessage(int size) {
    printf("잘못된 입력입니다. 1~%d 사이의 번호를 입력하세요.\n", size);
}

```

(4) 기능3 로그인 정보 수정 – 기능 3.1 로그인 정보 삭제

- 입력: choice = 사용자가 선택한 메뉴 번호를 저장하는 변수

site_count = 저장된 사이트의 수를 저장하는 변수

choice_3 = 로그인 정보 수정 메뉴의 선택 번호를 저장하는 변수

PrintSiteList = 전체 사이트 목록을 출력하는 함수

login_info = 사용자가 입력한 로그인 정보를 저장하는 구조체

DelLoginInfo = 로그인 정보를 삭제하는 함수

site_name = 사이트명을 저장하는 구조체 변수

id = 아이디를 저장하는 구조체 변수

password = 비밀번호를 저장하는 구조체 변수

del_index = 삭제할 사이트 번호 저장 변수

- 출력: login_info[del_index]의 값이 삭제되고 그 뒤의 인덱스의 저장 값들이 하나씩 앞으로 복사되어 저장

- 설명: 1. 저장되어 있는 정보가 있는지 확인
 2. 저장된 정보가 없을 경우(site_count=0) 오류 문구 출력
 3. 저장된 정보가 있을 경우 세부 기능 선택 받기
 4. 저장된 전체 사이트 목록을 순서대로 출력
 5. 삭제할 사이트 번호 입력 받기
 6. 삭제 확인 메시지 출력
 7. 삭제할 정보 뒤의 값을 하나씩 앞으로 복사
 8. site_count -1
 9. 메인 기능 화면으로 이동
- 적용된 배운 내용: 조건문(if), 반복문(while, for), 함수, 포인터 배열, 구조체, 헤더파일
- 코드 스크린샷

```

else if (choice == 3) {
    if (site_count == 0) {
        // 저장된 정보가 있는지 확인. 없을 경우 오류 문구 출력
        printf("저장된 로그인 정보가 없습니다. 로그인 정보를 추가하세요.\n");
    } else {
        while (1) {
            /*기능 3. 로그인 정보 수정*/
            int choice_3;
            printf("-----\n");
            printf("원하시는 기능을 선택하세요.\n");
            printf(
                "1. 로그인 정보 삭제\n2. 로그인 정보 수정\n3. 뒤로 "
                "가기\n");
            printf("-----\n");
            printf("번호: ");
            scanf_s("%d", &choice_3);

            /*기능 3.1 로그인 정보 삭제*/
            if (choice_3 == 1) {
                // 전체 사이트 목록 출력
                PrintSiteList(login_info, site_count);

                // 로그인 정보 삭제
                DelLoginInfo(login_info, site_count);
                site_count--;

                break;
            }
        }
    }
}

```

```

/*전체 사이트 목록을 출력하는 함수 정의*/
void PrintSiteList(LOGIN* p_login_info, int site_count) {
    printf("-- 사이트 목록 (총 %d) --\n", site_count);
    for (int i = 0; i < site_count; i++) {
        printf("    %d. %s\n", i + 1, p_login_info[i].site_name);
    }
    printf("-----\n");
}

/*로그인 정보를 삭제하는 함수 정의*/
void DelLoginInfo(LOGIN* login_info, int site_count) {
    int del_index; // 삭제할 사이트 번호 저장 변수

    printf("삭제를 원하시는 사이트의 번호를 입력하세요: ");
    scanf_s("%d", &del_index);

    // 삭제 확인 메시지 출력
    printf("%s'의 로그인 정보가 정상적으로 삭제 되었습니다.\n",
        login_info[del_index - 1].site_name);

    // 삭제할 정보 뒤의 값들을 하나씩 앞으로 복사
    for (int i = del_index - 1; i < site_count - 1; i++) {
        memcpy_s(&login_info[i], sizeof(LOGIN), &login_info[i + 1], sizeof(LOGIN));
    }
}

```

(5) 기능3 로그인 정보 수정 - 기능 3.2 비밀번호 변경

- 입력: choice_3 = 로그인 정보 수정 메뉴의 선택 번호를 저장하는 변수

change_index = 변경할 사이트의 번호를 저장하는 변수

PrintSiteList = 전체 사이트 목록을 출력하는 함수

login_info = 사용자가 입력한 로그인 정보를 저장하는 구조체

site_count = 저장된 사이트의 수를 저장하는 변수

PrintLoginInfo = 로그인 정보 출력하는 함수

InitializeNewPassword = 비밀번호 변경 후 구조체 변수에 저장하는 함수

site_name = 사이트명을 저장하는 구조체 변수

id = 아이디를 저장하는 구조체 변수

password = 비밀번호를 저장하는 구조체 변수

temp = 사용자가 입력한 정보를 임시 저장하는 변수

InitializePassword = 비밀번호를 구조체 변수에 저장하는 함수

- 출력: login_info[change_index-1].password에 새로운 비밀번호 저장

- 설명: 1. 전체 사이트의 목록 출력

2. 비밀번호를 변경할 사이트의 번호 입력 받기 및 change_index에 번호 저장

3. 선택한 사이트의 현재 로그인 정보 출력

4. 변경할 비밀번호 입력 받아 저장하기

5. 변경된 로그인 정보를 출력하기

- 적용된 배운 내용: 조건문(if), 반복문(for, while), 함수, 문자열, 포인터 배열, 구조체, 헤더파일, 동적할당

- 코드 스크린샷

```
/*가능 3.2 비밀번호 변경*/
else if (choice_3 == 2) {
    int change_index; // 변경할 사이트의 번호를 저장하는 변수

    // 전체 사이트 목록 출력
    PrintSiteList(login_info, site_count);

    // 비밀번호를 변경할 사이트의 번호를 입력받기
    printf("비밀번호를 변경할 사이트를 선택해 주세요: ");
    scanf_s("%d", &change_index);

    // 선택한 사이트의 현재 로그인 정보 출력
    printf("----현재 로그인 정보----\n");
    PrintSiteLoginInfo(login_info, site_count, change_index);

    // 변경할 비밀번호 입력받기
    InitializeNewPassword(&login_info[change_index-1]);

    // 변경된 로그인 정보 출력
    printf("---변경된 로그인 정보---\n");
    PrintSiteLoginInfo(login_info, site_count, change_index);
}
```

```
/*전체 사이트 목록을 출력하는 함수 정의*/
void PrintSiteList(LOGIN* p_login_info, int site_count) {
    printf("-- 사이트 목록 (총 %d) --\n", site_count);
    for (int i = 0; i < site_count; i++) {
        printf("    %d. %s\n", i + 1, p_login_info[i].site_name);
    }
    printf("-----\n");
}
```

```

/*로그인 정보를 출력하는 함수 정의*/
void PrintLoginInfo(LOGIN* p_login_info, int site_count, int index) {
    if (index == 0) {
        // 전체 로그인 정보 출력
        printf("--- 전체 로그인 정보 ---\n");
        for (int i = 0; i < site_count; i++) {
            printf("%d. 사이트: %s 아이디: %s 비밀번호: %s\n", i + 1,
                p_login_info[i].site_name, p_login_info[i].id,
                p_login_info[i].password);
        }
        printf("-----\n");
    } else {
        // 특정 사이트 로그인 정보 출력
        printf("사이트: %s\n아이디: %s\n비밀번호: %s\n",
            p_login_info[index - 1].site_name, p_login_info[index - 1].id,
            p_login_info[index - 1].password);
        printf("-----\n");
    }
}

```

```

/*비밀번호 변경 후 구조체 변수에 저장하는 함수 정의*/
void InitializeNewPassword(LOGIN* login_info) {
    // 변경할 사이트의 비밀번호 동적할당 해제
    free(login_info->password);

    // 새로운 비밀번호 입력받기
    printf("새로운 비밀번호를 입력하세요: ");
    InitializePassword(login_info);
}

```

```

// 비밀번호를 구조체 변수에 저장하는 함수 정의
void InitializePassword(LOGIN* login_info) {
    char temp[100];
    printf("    비밀번호: ");
    scanf_s("%s", temp, (int)sizeof(temp));
    login_info->password = (char*)malloc((strlen(temp) + 1) * sizeof(char));
    strcpy_s(login_info->password, strlen(temp) + 1, temp);
}

```

(6) 기능4 추천 비밀번호 – 기능 4.1 비밀번호 추천

- 입력: choice = 사용자가 선택한 메인 기능 메뉴 번호를 저장하는 변수

password_len = 추천 비밀번호 길이를 입력받을 변수

choice_4 = 추천 비밀번호 저장 메뉴의 선택 번호를 저장하는 변수

rec_arr = 비밀번호 추천하는데 사용할 문자의 배열(알파벳 소문자, 대문자, 특수 문자, 숫자)

rec_password = 추천 비밀번호를 저장할 포인터 배열

- 출력: rec_password에 추천 비밀번호 저장

- 설명: 1. 사용자로부터 추천받을 비밀번호 길이 입력 받기

2. 랜덤 수를 추천 문자 배열의 길이로 나눈 나머지의 인덱스 위치에 있는 문자를 추천 비밀번호 문자열에 순서대로 저장
3. 추천 비밀번호 문자열 맨 뒤 인덱스에 null 값 저장
4. 추천 비밀번호 출력
5. 추천 비밀번호 바로 저장 메뉴 출력 및 번호 입력 받기

- 적용된 배운 내용: 조건문(if), 반복문(for), 문자열, 포인터 배열, 동적할당
- 코드 스크린샷

```

else if (choice == 4) {
/*기능 4. 추천 비밀번호 생성*/
int password_len = 0; // 추천 비밀번호 길이를 입력받을 변수
int choice_4; // 세부기능 선택 번호 저장 변수

// 비밀번호 추천에 사용할 문자 배열
char rec_arr[] =
    "abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ~!@#$%^&*_-/"
    "?0123456789";

// 비밀번호 길이 입력받기
printf("몇 자리의 비밀번호를 추천해 드릴까요? ");
scanf_s("%d", &password_len);
char* rec_password = (char*)malloc(password_len + 1);
printf(" 추천 비밀번호: ");

/*랜덤수의 나머지 위치의 인덱스에 맞는 문자를 출력을 반복하여 추천
비밀번호 생성 및 저장*/
for (int i = 0; i < password_len; i++) {
    rec_password[i] = rec_arr[rand() % strlen(rec_arr)];
}
rec_password[password_len] = '\0'; // 맨 뒤의 자리에 null값 저장
printf("%s", rec_password);
printf("\n");

printf("-----\n");
printf(
    "1. 새로운 로그인 정보에 추가\n2. 기존 로그인 정보에 추가\n3. 뒤로
    가기\n");
printf("-----\n");
printf("번호: ");
scanf_s("%d", &choice_4);

```

(7) 기능4 추천 비밀번호 – 기능 4.2.1 새로운 로그인 정보에 저장

- 입력: choice_4 = 추천 비밀번호 저장 메뉴의 선택 번호를 저장하는 변수

InitializeSite = 사이트명을 구조체 변수에 저장하는 함수

Initializeld = 아이디를 구조체 변수에 저장하는 함수

site_count = 저장된 사이트의 수를 저장하는 변수

login_info = 사용자가 입력한 로그인 정보를 저장하는 구조체

site_name = 사이트명을 저장하는 구조체 변수

id = 아이디를 저장하는 구조체 변수

rec_password = 추천 비밀번호를 저장할 포인터 배열

password = 비밀번호를 저장하는 구조체 변수

temp = 사용자가 입력한 정보를 임시 저장하는 변수

site_count = 저장된 사이트의 수를 저장하는 변수

PrintLoginInfo = 로그인 정보 출력하는 함수

- 출력: login_info[site_count].site_name에 입력받은 사이트명 저장

login_info[site_count].id에 입력받은 아이디 저장

login_info[site_count].password에 추천 비밀번호 저장

site_count + 1

- 설명: 1. 사이트명 입력 받기 및 구조체 변수에 저장

2. 아이디 입력 받기 및 구조체 변수에 저장

3. 추천받은 비밀번호를 구조체 변수에 저장

4. site_count + 1

5. 로그인 정보 저장 확인 메시지 출력

- 적용된 배운 내용: 조건문(if), 함수, 포인터 배열, 구조체, 헤더파일, 동적할당

- 코드 스크린샷

```

if (choice_4 == 1) {
    /*새로운 로그인 정보에 추가*/
    InitializeSite(&login_info[site_count]); // 사이트 입력 및 저장
    Initializeld(&login_info[site_count]); // 아이디 입력 및 저장

    // 추천 받은 비밀번호 저장
    login_info[site_count].password =
        (char*)malloc((strlen(rec_password) + 1) * sizeof(char));
    strcpy_s(login_info[site_count].password, strlen(rec_password) + 1,
        rec_password);
    site_count++;
    printf("- 다음 로그인 정보가 저장되었습니다 -\n");

    PrintLoginInfo(login_info, site_count, site_count);

    // 사이트명을 구조체 변수에 저장하는 함수 정의
void InitializeSite(LOGIN* login_info) {
    char temp[100]; // 입력받은 값 임시 저장 변수
    printf("    사이트: ");
    scanf_s("%s", temp, (int)sizeof(temp));

    // 동적할당 & 구조체에 저장
    login_info->site_name = (char*)malloc((strlen(temp) + 1) * sizeof(char));
    strcpy_s(login_info->site_name, strlen(temp) + 1, temp);
}

// 아이디를 구조체 변수에 저장하는 함수 정의
void Initializeld(LOGIN* login_info) {
    char temp[100];
    printf("    아이디: ");
    scanf_s("%s", temp, (int)sizeof(temp));
    login_info->id = (char*)malloc((strlen(temp) + 1) * sizeof(char));
    strcpy_s(login_info->id, strlen(temp) + 1, temp);
}
}

```

```

/*로그인 정보를 출력하는 함수 정의*/
void PrintLoginInfo(LOGIN* p_login_info, int site_count, int index) {
    if (index == 0) {
        // 전체 로그인 정보 출력
        printf("--- 전체 로그인 정보 ---\n");
        for (int i = 0; i < site_count; i++) {
            printf("%d. 사이트: %s 아이디: %s 비밀번호: %s\n", i + 1,
                p_login_info[i].site_name, p_login_info[i].id,
                p_login_info[i].password);
        }
        printf("-----\n");
    } else {
        // 특정 사이트 로그인 정보 출력
        printf("사이트: %s\n아이디: %s\n비밀번호: %s\n",
            p_login_info[index - 1].site_name, p_login_info[index - 1].id,
            p_login_info[index - 1].password);
        printf("-----\n");
    }
}
}

```

(8) 기능4 추천 비밀번호 - 기능 4.2.2 기존 로그인 정보에 저장

- 입력: choice_4 = 추천 비밀번호 저장 메뉴의 선택 번호를 저장하는 변수

change_index = 변경할 사이트의 번호를 저장하는 변수

PrintSiteList = 전체 사이트 목록을 출력하는 함수

login_info = 사용자가 입력한 로그인 정보를 저장하는 구조체

site_count = 저장된 사이트의 수를 저장하는 변수

PrintLoginInfo = 로그인 정보 출력하는 함수

site_name = 사이트명을 저장하는 구조체 변수

id = 아이디를 저장하는 구조체 변수

password = 비밀번호를 저장하는 구조체 변수

rec_password = 추천 비밀번호를 저장할 포인터 배열

temp = 사용자가 입력한 정보를 임시 저장하는 변수

- 출력: login_info[chage_index-1].password에 새로운 비밀번호 저장
- 설명: 1. 전체 사이트의 목록 출력
 - 2. 비밀번호를 변경할 사이트의 번호 입력 받기 및 change_index에 번호 저장
 - 3. 선택한 사이트의 이전 로그인 정보 출력
 - 4. 추천 비밀번호로 저장하기
 - 5. 변경된 로그인 정보를 출력하기
- 적용된 배운 내용: 조건문(if), 반복문(for), 함수, 포인터 배열, 구조체, 헤더파일, 동적할당
- 코드 스크린샷


```

else if (choice_4 == 2) {
/*기존 로그인 정보에 추가*/
int change_index; // 변경할 사이트의 번호를 저장하는 변수

// 전체 사이트 목록 출력
PrintSiteList(login_info, site_count);

// 비밀번호를 변경할 사이트의 번호를 입력받기
printf("비밀번호를 변경할 사이트를 선택해주세요: ");
scanf_s("%d", &change_index);

// 선택한 사이트의 현재 로그인 정보 출력
printf("----이전 로그인 정보----\n");
PrintLoginInfo(login_info, site_count, change_index);

// 변경할 비밀번호 동적할당 해제
free(login_info[change_index - 1].password);

// 추천 비밀번호 동적할당 및 저장
login_info[change_index - 1].password =
(char*)malloc((strlen(rec_password) + 1) * sizeof(char));
strcpy_s(login_info[change_index - 1].password,
strlen(rec_password) + 1, rec_password);

// 변경된 로그인 정보 출력
printf("---변경된 로그인 정보---\n");
PrintLoginInfo(login_info, site_count, change_index);
}

```

```

/*전체 사이트 목록을 출력하는 함수 정의*/
void PrintSiteList(LOGIN* p_login_info, int site_count) {
printf("-- 사이트 목록 (총 %d) --\n", site_count);
for (int i = 0; i < site_count; i++) {
printf("    %d. %s\n", i + 1, p_login_info[i].site_name);
}
printf("-----\n");
}

```

```

/*로그인 정보를 출력하는 함수 정의*/
void PrintLoginInfo(LOGIN* p_login_info, int site_count, int index) {
if (index == 0) {
// 전체 로그인 정보 출력
printf("--- 전체 로그인 정보 ---\n");
for (int i = 0; i < site_count; i++) {
printf("%d. 사이트: %s 아이디: %s 비밀번호: %s\n", i + 1,
p_login_info[i].site_name, p_login_info[i].id,
p_login_info[i].password);
}
printf("-----\n");
} else {
// 특정 사이트 로그인 정보 출력
printf("사이트: %s\n아이디: %s\n비밀번호: %s\n",
p_login_info[index - 1].site_name, p_login_info[index - 1].id,
p_login_info[index - 1].password);
printf("-----\n");
}
}

```

(8) 기능5 프로그램 종료

- 입력: choice = 사용자가 선택한 메뉴 번호를 저장하는 변수

login_info = 사용자가 입력한 로그인 정보를 저장하는 구조체

site_name = 사이트명을 저장하는 구조체 변수

id = 아이디를 저장하는 구조체 변수

password = 비밀번호를 저장하는 구조체 변수

- 출력: 없음
- 설명: 1. 저장된 정보의 동적할당 해제
2. 프로그램 종료 문구 출력 및 종료
- 적용된 배운 내용: 조건문(if), 동적할당, 반복문(for)
- 코드 스크린샷

```
else if (choice == 5) {  
    /*프로그램 종료*/  
  
    // 동적 할당 해제  
    for (int i = 0; i < site_count; i++) {  
        free(login_info[i].site_name);  
        free(login_info[i].id);  
        free(login_info[i].password);  
    }  
    free(login_info);  
  
    // 프로그램 종료  
    printf("프로그램을 종료합니다.");  
    break;  
}
```

4. 테스트 결과

(1) 기능1 로그인 정보 저장

- 설명: 로그인 정보가 각각의 구조체 변수에 잘 저장되었는지 확인 및 총 사이트 수 확인
- 테스트 결과 스크린샷

```
번호: 1  
사이트: naver  
아이디: naverid  
비밀번호: naverpassword  
-- 로그인 정보가 저장 되었습니다 --  
login_info[0].site_name: naver login_info[0].id: naverid login_info[0].password: naverpassword  
site_count: 1
```

(2) 기능2 로그인 정보 출력 - 기능2.1 저장된 사이트 목록 출력

- 설명: 저장된 정보가 있는 지 확인 및 저장된 전체 사이트 목록 출력 확인
- 테스트 결과 스크린샷

```
번호 : 2
저장된 로그인 정보가 없습니다. 로그인 정보를 추가하세요.
```

```
번호 : 2
-- 사이트 목록 (총 2) --
1. naver
2. kakao
-----
```

(3) 기능2 로그인 정보 출력 - 기능2.2.1 전체 사이트의 로그인 정보 출력

- 설명: 전체 사이트의 로그인 정보 출력 및 메인 기능으로 이동 확인
- 테스트 결과 스크린샷

```
-----
원하시는 기능을 선택하세요.
1. 전체 사이트 로그인 정보
2. 특정 사이트의 로그인 정보
3. 뒤로 가기
-----
번호 : 1
--- 전체 로그인 정보 ---
1. 사이트: naver 아이디: naverid 비밀번호: naverpassword
2. 사이트: kakao 아이디: kakaoaid 비밀번호: kakaopassword
-----
=====
--메뉴를 선택해주세요.--
1. 로그인 정보 입력
2. 로그인 정보 보기
3. 로그인 정보 변경
```

(4) 기능2 로그인 정보 출력 - 기능2.2.2 특정 사이트의 로그인 정보 출력

- 설명: 사용자가 특정한 사이트의 로그인 정보 출력 및 뒤로가기 기능 작동 확인
- 테스트 결과 스크린샷

```

-----
원하시는 기능을 선택하세요.
1. 전체 사이트 로그인 정보
2. 특정 사이트의 로그인 정보
3. 뒤로 가기
-----
번호: 2
원하시는 사이트의 번호를 입력하세요: 1
--- 1번 로그인 정보 ---
사이트: naver
아이디: naverid
비밀번호: naverpassword
-----
원하시는 기능을 선택하세요.
1. 전체 사이트 로그인 정보
2. 특정 사이트의 로그인 정보
3. 뒤로 가기
-----
번호: 3
=====
--메뉴를 선택해주세요.--
1. 로그인 정보 입력
2. 로그인 정보 보기
3. 로그인 정보 변경

```

(5) 기능3 로그인 정보 수정

- 설명: 저장된 정보가 있는지 확인 및 세부 메뉴 출력 확인
- 테스트 결과 스크린샷

```

번호: 3
저장된 로그인 정보가 없습니다. 로그인 정보를 추가하세요.

```

```

=====
--메뉴를 선택해주세요.--
1. 로그인 정보 입력
2. 로그인 정보 보기
3. 로그인 정보 변경
4. 추천 비밀번호
5. 프로그램 종료
=====
번호: 3
-----
원하시는 기능을 선택하세요.
1. 로그인 정보 삭제
2. 로그인 정보 수정
3. 뒤로 가기
-----
번호: |

```

(6) 기능3 로그인 정보 변경 - 기능3.1 로그인 정보 삭제

- 설명: 사용자가 선택한 사이트의 로그인 정보 삭제 확인, 총 사이트 수 감소 확인, 메인 기능으로 이동 확인
- 테스트 결과 스크린샷

```

번호: 3
-----
원하시는 기능을 선택하세요.
1. 로그인 정보 삭제
2. 로그인 정보 수정
3. 뒤로 가기
-----
번호: 1
-- 사이트 목록 (총 2) --
    1. naver
    2. kakao
-----
삭제를 원하시는 사이트의 번호를 입력하세요: 1
'naver'의 로그인 정보가 정상적으로 삭제 되었습니다.
site_count: 1
1. 사이트: kakao 아이디: kakaoaid 비밀번호: kakaopassword
=====
--메뉴를 선택해주세요.--
1. 로그인 정보 입력
2. 로그인 정보 보기

```

(6) 기능3 로그인 정보 변경 - 기능3.2 비밀번호 변경

- 설명: 전체 사이트 목록 출력, 선택한 사이트의 현재 로그인 정보 출력, 새로운 비밀번호 입력 받기, 변경된 로그인정보 출력, 뒤로가기 기능 확인
- 테스트 결과 스크린샷

```

1. 로그인 정보 삭제
2. 로그인 정보 수정
3. 뒤로 가기
-----
번호: 2
-- 사이트 목록 (총 1) --
    1. kakao
-----
비밀번호를 변경할 사이트를 선택해주세요: 1
----현재 로그인 정보----
사이트: kakao
아이디: kakaoaid
비밀번호: kakaopassword
-----
새로운 비밀번호: kakao_pw12
---변경된 로그인 정보---
사이트: kakao
아이디: kakaoaid
비밀번호: kakao_pw12
-----
원하시는 기능을 선택하세요.
1. 로그인 정보 삭제
2. 로그인 정보 수정
3. 뒤로 가기
-----
번호: 3
=====
--메뉴를 선택해주세요.--
1. 로그인 정보 입력

```

(7) 기능4 추천 비밀번호 - 기능4.1 비밀번호 추천

- 설명: 추천 비밀번호가 저장 변수에 저장 및 출력 확인
- 테스트 결과 스크린샷

```
3. 로그인 정보 변경
4. 추천 비밀번호
5. 프로그램 종료
=====
번호: 4
몇 자리의 비밀번호를 추천해 드릴까요? 10
rec_password: PrIzSXdHLo
추천 비밀번호: PrIzSXdHLo
=====
```

(8) 기능4 추천 비밀번호 - 기능4.2.1 새로운 로그인 정보에 저장

- 설명: 새로운 사이트, 아이디 입력 받기, 추천 비밀번호 로그인 정보에 저장 및 출력, 메인 기능으로 이동 확인
- 테스트 결과 스크린샷

```
1. 새로운 로그인 정보에 추가
2. 기존 로그인 정보에 추가
3. 뒤로 가기
-----
번호: 1
  사이트: google
  아이디: gooleid
- 다음 로그인 정보가 저장되었습니다 -
사이트: google
아이디: gooleid
비밀번호: PrIzSXdHLo
-----
=====
--메뉴를 선택해주세요.--
1. 로그인 정보 입력
```

(9) 기능4 추천 비밀번호 - 기능4.2.2 기존 로그인 정보에 저장

- 설명: 전체 사이트 목록 출력, 변경할 사이트 입력 받기, 해당 사이트의 이전 로그인 정보, 변경 후 로그인 정보 출력, 메인 기능으로 이동 확인
- 테스트 결과 스크린샷

```

1. 새로운 로그인 정보에 추가
2. 기존 로그인 정보에 추가
3. 뒤로 가기
-----
번호: 2
-- 사이트 목록 (총 2) --
  1. kakao
  2. google
-----
비밀번호를 변경할 사이트를 선택해주세요: 1
----이전 로그인 정보----
사이트: kakao
아이디: kakaoid
비밀번호: kakao_pw12
-----
---변경된 로그인 정보---
사이트: kakao
아이디: kakaoid
비밀번호: fuFB_P5rBK1@c
-----
=====
--메뉴를 선택해주세요.--
1. 로그인 정보 입력
2. 로그인 정보 보기

```

(10) 기능5 프로그램 종료

- 설명: 프로그램이 정상적으로 종료되는 지 확인
- 테스트 결과 스크린샷

```

2. 로그인 정보 보기
3. 로그인 정보 변경
4. 추천 비밀번호
5. 프로그램 종료
=====
번호: 5
프로그램을 종료합니다.
C:\Users\dbgml\OneDrive\바탕 화면\login_inform

```

5. 계획 대비 변경 사항

1) 로그인 정보 출력 기능 세분화

- 이전: 사용자가 지정하는 사이트만 출력 가능
- 이후: 사용자가 지정하는 사이트만 출력하는 기능과 전체 사이트의 로그인 정보를 출력하는 기능으로 세분화함

- 사유: 저장된 로그인 정보 전체를 보고자 할 때, 이전의 경우에는 사용자가 하나하나 계속 입력을 해줘야만 출력이 되기 때문에 번거로웠음. 이러한 이유로 한 번에 볼 수 있게 전체 사이트의 로그인 정보를 출력하는 기능을 추가하여 세분화함.

2) 로그인 정보 수정 기능 추가

- 이전: 한번 입력한 로그인 정보를 수정할 수 없었음
- 이후: 로그인 정보의 삭제 및 비밀번호 변경 가능
- 사유: 이전의 계획에서는 로그인 정보의 수정 기능이 없어 한번 입력하면 내용을 변경할 수 없었음. 그러나 잘못된 정보 입력, 사이트의 탈퇴 및 비밀번호 변경 등과 같은 이유로 로그인 정보의 수정 기능의 필요성을 인지하였고, 그에 따라 해당 기능을 추가함.

3) 프로그램 종료 기능 추가

- 이전: 프로그램을 종료할 수 없었음
- 이후: 프로그램 사용 후 종료 가능
- 사유: 이전의 계획에서는 프로그램을 사용 후 종료할 수 없어서 계속해서 프로그램 실행되어 있었음. 따라서 프로그램을 종료하는 기능을 추가하여 더 이상 다른 기능이 수행되지 않도록 함.

4) 구조체로 로그인 정보 관리

- 이전: 구조체를 사용하지 않고 각 로그인 정보를 따로따로 관리함
- 이후: 구조체를 통해 각 로그인 정보를 하나의 묶음으로 관리
- 사유: 구조체로 사용자의 로그인 정보를 한 번에 관리함으로써 효율성을 높이고, 코드를 단순화하기 위해 사용

5) 입력받은 데이터를 동적 할당을 통해 관리

- 이전: 정적 메모리 할당을 사용하여 입력 가능 개수에 제한과 메모리 공간을 비효율적으로 사용함

- 이후: 동적 메모리 할당을 사용하여 저장 가능한 정보의 개수에 제한 두지 않고 사용할 수 있음
- 사유: 메모리 공간을 보다 효율적으로 관리하고 사용자가 저장할 수 있는 데이터의 양에 제한을 두지 않기 위해 사용

6) 헤더파일 생성

- 이전: 사용자 정의 함수를 기본 코드들과 한 파일 안에서 관리함
- 이후: 헤더 파일을 통해 사용자 정의 함수와 기본 코드를 다른 파일로 분리해서 관리
- 사유: 헤더파일을 사용함으로써 코드의 길이를 줄이고, 함수, 구조체 등의 재사용을 편리하게 하기 위해 사용

7) 전체 사이트목록 출력, 비밀번호 변경, 에러메시지 출력, 로그인 정보출력 코드 등 반복되는 코드 함수화

- 이전: 함수화 하지 않고 사용하여 여러 중복된 코드들이 존재하였음
- 이후: 해당 기능들을 함수화하여 코드 단순화
- 사유: 중복 사용되는 코드들을 함수화하여 코드를 단순화하기 위해 사용

8) 추천받은 비밀번호를 바로 저장할 수 있는 기능 추가

- 이전: 추천받은 비밀번호를 바로 저장하지 못하고 사용자가 직접 복사하여 저장해야 했음.
- 이후: 추천받은 비밀번호를 새 로그인 정보나 기존의 로그인 정보에 바로 추가가 가능함
- 사유: 이전에는 추천받은 비밀번호를 저장하려면 직접 복사하여 원하는 기능으로 이동하여 추가해야 했었음. 이에 대한 번거로움이 있다고 판단하여 비밀번호를 추천받고 그 자리에서 새 로그인 정보나 기존 로그인 정보에 바로 저장할 수 있도록 하기 위해 해당 기능을 추가함

6. 느낀점

- 하나의 프로그램을 원하는 작동을 하게 하기 위해 기능 설계부터 구현까지 직접 하나 하나 계획하며 만들어 보면서 이 과정이 생각보다 쉽지 않았다는 것을 알았다. 비슷한 코드이지만 함수화하는 과정에서 조금씩 다른 부분이 있어 그 부분을 어떻게 할 것인지, 코드를 어떻게 해야 길이를 줄일 수 있는지, 어떤 알고리즘을 사용해야 더 단순화할 수 있는지 등 여러 고민이 많았다. 오류를 찾아 해결하는 과정은 특히 어려웠다. 디버그를 사용해 봐도 왜 오류가 발생하는지 알 수 없는 경우도 있었기 때문이다. 프로그램 기능별 구현 일정을 세우는 과정에서도 자신의 실력을 잘 알고 그에 맞게 계획을 세워야 일정에 맞춰 구현 가능하다는 것을 느꼈다. 나의 경우 초기 일정보다 구현이 계속 늦어져 특히 더 중요함을 깨달았다. 비록 이 수업은 이것으로 끝났지만 이후에도 계속 스스로 여러 프로그램을 만들면서 어떻게 해야 코드를 단순화하여 깔끔하게 짤 수 있을지, 어떤 코드를 어떻게 짜야 할 지 등 코딩 실력을 늘려 나가야겠다.