

C프로그래밍및실습

# 로그인정보 저장 프로그램 개발

진척 보고서 #2

제출일자: 2023.12.10.(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: 비밀번호 추천**

- 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. 진척사항

### 1) 기능 구현

#### (1) 기능1 로그인 정보 저장

- 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

id = 아이디를 저장하는 구조체 변수

password = 비밀번호를 저장하는 구조체 변수

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 임시변수에 저장하고 동적할당 후 각 구조체 변수에 저장한다

3. 사이트 수를 저장하는 변수의 값을 1 증가한다

4. 로그인 정보가 저장되었다는 메시지를 출력한다

- 적용된 배운 내용: 조건문, 함수, 포인터배열, 동적할당, 구조체, 헤더파일

- 코드 스크린샷

```
if (choice == 1) {  
    /*기능 1. 로그인 정보 저장*/  
    /*기능 1.1 사용자로부터 로그인 정보 입력 받기*/  
    InitializeLogin(&login_info[site_count]);  
    site_count++;  
  
    printf("— 로그인 정보가 저장 되었습니다 —\n");  
}
```

```
/*로그인 정보를 구조체에 저장하는 함수 정의*/  
void InitializeLogin(LOGIN* login_info) {  
    // 사이트명 구조체 변수에 저장  
    printf("   사이트: ");  
    InitializeSite(login_info);  
    // 아이디 구조체 변수에 저장  
    printf("   아이디: ");  
    InitializeId(login_info);  
    // 비밀번호 구조체 변수에 저장  
    printf("   비밀번호: ");  
    InitializePassword(login_info);  
}  
  
// 사이트명을 구조체 변수에 저장하는 함수 정의  
void InitializeSite(LOGIN* login_info) {  
    char temp[100]; // 입력받은 값 임시 저장 변수  
    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];  
    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];  
    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

- 출력: 없음

- 설명: 1. 저장되어 있는 정보가 있는지 확인

2. 저장된 사이트의 목록을 순서대로 출력

- 적용된 배운 내용: 조건문, 반복문, 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

site\_count = 저장된 사이트의 수를 저장하는 변수

site\_choice = 로그인 정보를 출력할 사이트의 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. 로그인 정보 출력 기능의 메뉴 출력 및 사용자의 선택
  2. 1번 선택 시 전체 사이트의 로그인 정보를 순서대로 출력
  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, 반복문, 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*/
        PrintSiteLoginInfo(login_info, site_count, 0);
    } 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);
        PrintSiteLoginInfo(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 PrintSiteLoginInfo(LOGIN* p_login_info, int site_count, int index) {
    if (index == 0) {
        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);
        }
    } else {
        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 3.2 비밀번호 변경

- 입력: choice\_3 = 로그인 정보 수정 메뉴의 선택 번호를 저장하는 변수

change\_index = 변경할 사이트의 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

temp = 사용자가 입력한 정보를 임시 저장하는 변수

- 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, 반복문, 함수, 포인터 배열, 구조체, 헤더파일, 동적할당
- 코드 스크린샷

```

/*기능 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 선택해주세요: ");
    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 PrintSiteLoginInfo(LOGIN* p_login_info, int site_count, int index) {
    if (index == 0) {
        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);
        }
    } else {
        printf("사이트: %s\n아이디: %s\n비밀 번호: %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];
    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);
}
```

## (5) 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. 동적할당 해제

2. 프로그램 종료

- 적용된 배운 내용: 조건문, 동적할당

- 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;
}
```

## 2) 테스트 결과

### (1) 기능1 로그인 정보 저장

- 설명: 로그인 정보가 각각의 구조체변수에 잘 저장되었는지 확인

- 테스트 결과 스크린샷

```
번호: 1
  사이트: tkdlxm
  아이디: dkdlel
  비밀번호: qlalfqjsgh
-- 로그인 정보가 저장되었습니다 --
login_info[0].site_name: tkdlxm login_info[0].id: dkdlel login_info[0].password: qlalfqjsgh
```

## (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 3) --
  1. naver
  2. kakao
  3. google
-----
```

## (3) 기능2 로그인 정보 출력 - 기능2.2 로그인 정보 출력

- 설명: 전체 및 특정 사이트의 로그인 정보 출력 기능, 뒤로가기 기능 작동 확인

- 테스트 결과 스크린샷

```

1. 전체 사이트 로그인 정보
2. 특정 사이트의 로그인 정보
3. 뒤로 가기
-----
번호: 1
1. 사이트: naver 아이디: naverid 비밀번호: naverpw
2. 사이트: kakao 아이디: kakaoid 비밀번호: kakaopw
3. 사이트: google 아이디: googleid 비밀번호: googlepw
-----
원하시는 기능을 선택하세요.
1. 전체 사이트 로그인 정보
2. 특정 사이트의 로그인 정보
3. 뒤로 가기
-----
번호: 2
원하시는 사이트의 번호를 입력하세요: 2
----2번 로그인 정보----
사이트: kakao
아이디: kakaoid
비밀번호: kakaopw
-----
원하시는 기능을 선택하세요.
1. 전체 사이트 로그인 정보
2. 특정 사이트의 로그인 정보
3. 뒤로 가기
-----
번호: 3
=====
--메뉴를 선택해주세요.--
1. 로그인 정보 입력

```

#### (4) 기능3 로그인 정보 수정 - 기능 3.2 비밀번호 변경

- 설명: 전체 사이트 목록 출력, 선택한 사이트의 현재 로그인 정보 출력, 변경된 로그인 정보 출력 기능 작동 확인 및 변경된 비밀번호가 잘 저장되었는지 확인
- 테스트 결과 스크린샷

```

원하시는 기능을 선택하세요.
1. 로그인 정보 삭제
2. 로그인 정보 수정
3. 뒤로 가기
-----
번호: 2
-- 사이트 목록 (총 3) --
  1. naver
  2. kakao
  3. google
-----
비밀번호를 변경할 사이트를 선택해주세요: 2
----현재 로그인 정보----
사이트: kakao
아이디: kakaoaid
비밀번호: kakaopw
-----
새로운 비밀번호를 입력하세요: kakao_pw12!
---변경된 로그인 정보---
사이트: kakao
아이디: kakaoaid
비밀번호: kakao_pw12!
-----
login_info[0].site_name: naver login_info[0].id: nvaerid login_info[0].password: naverpw
login_info[1].site_name: kakao login_info[1].id: kakaoaid login_info[1].password: kakao_pw12!
login_info[2].site_name: google login_info[2].id: googleid login_info[2].password: googlepw

```

## (5) 기능5 프로그램 종료

- 설명: 프로그램이 정상적으로 종료되는지 확인
- 테스트 결과 스크린샷

```

번호: 5
프로그램을 종료합니다.

```

## 4. 계획 대비 변경 사항

### 1) 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

### 2) 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

### 3) 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

### 4) 전체 사이트목록 출력, 비밀번호 변경, 에러메시지 출력, 로그인 정보출력 코드 등 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

### 5) 추천받은 비밀번호를 바로 저장할 수 있는 기능 추가

- 이전: 추천받은 비밀번호를 바로 저장할 수 없었음
- 이후: 추천받은 비밀번호를 바로 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

## 5. 프로젝트 일정

업무		11/3	11/14	11/15	11/20	11/21	11/22	12/8	12/10	12/17	12/22
제안서 작성		완료									
기능1	-			완료							
기능2	세부기능1				완료						
	세부기능2				완료						
기능3	세부기능1					----->					
	세부기능2					완료					
기능4	세부기능1						진행중				
	세부기능2									----->	
기능5	-		완료								