

C프로그래밍 및 실습

# 피부 관리 프로그램 개발

최종 보고서

제출일자: 2023-12-24

제출자명: 남궁윤아

제출자학번: 233603

## 1. 프로젝트 목표

### 1) 배경 및 필요성

불규칙적인 생활이나 패스트푸드를 자주 먹으면서 피부문제가 생긴 사람들이 많아지고 있다. 이러한 피부문제를 방치하면 더욱 더 심해져서 돌이킬 수 없게 된다. 그래서 하루빨리 관리를 시작하면서 피부를 다시 회복시킬 수 있도록 도와주는 프로그램을 개발하면 좋을 것 같다는 생각이 들었다. 피부가 안 좋으면 신경 쓰이고 힘든데 어떻게 해야할지 잘 알지 못해서 헤맸던 경험이 있어서 그런 걸 좀 잡아주고 상태를 기록할 수 있는 프로그램을 구상해보았다. 새로운 화장품을 사서 안맞아도 이게 여드름이 더 난건지 비교를 할 수 없었어서 안맞는 것을 계속 쓰다가 뒤늦게야 중단한 적도 여러 번 있었기에 이것도 방지하고자 한다.

### 2) 프로젝트 목표

사용자들의 피부타입을 분석하여 알려주고, 자신의 피부와 관련해 기록할 수 있고 약 먹는 날을 알려주고 사용자에게 맞는 화장품을 추천하는 프로그램을 만드는 것을 목표로 한다.

### 3) 차별점

기존의 프로그램 중에 '화해'라는 것이 있다. 하지만 이것은 단순히 화장품에 대한 리뷰를 모아둔 앱으로 자신의 피부타입을 모르는 상태에서 단지 다른 사람들의 리뷰만을 보고 사는 문제점이 있다. 나는 피부타입을 알 수 있고, 자신의 화장품과 피부에 대해서 기록하고 돌아볼 수 있는 것으로 다른 프로그램과 차별된다.

## 2. 기능 계획

### 1) 기능 1: 피부타입 진단

- 사용자의 피부타입이 무엇인지 알려주는 기능

### (1) 세부 기능 1: 사용자의 선택을 기반으로 피부타입 분석결과 도출

- 양자택일 질문을 해서 사용자가 선택하게 하고(1 또는 2를 입력받음) 배열에 담아 담긴 값을 for문으로 다 더한 뒤 if문으로 더한 값에 따라 지성인지 복합성인지 건성인지 분석해서 진단해준다.

## 2) 기능 2: 화장품 사용 이력 저장소

- 여러 기초화장품을 사다보면 뭐가 안 맞았는지 까먹을 수도 있기 때문에 자신만의 화장품 저장 공간을 만들어서 나만의 별점이랑 간단한 메모 느낌으로 작성할 수 있는 기능

### (1) 세부기능 1: 작성하기 (수정하기)

- 사용자로부터 별점과 메모를 입력 받는다. -> 별점, 화장품이름, 메모를 구조체를 사용하여 저장한다. (동적 메모리도 사용해서 낭비없이 잘 저장하기!) + 파일 3개로 만들기(헤더파일 사용)

### (2) 세부기능 2: 삭제하기

- 사용자가 입력했던 화장품의 이름, 별점, 메모를 삭제한다. 저장된 화장품이 없다면 없다는 문구를 출력한다. 삭제한 후에는 배열을 당겨오는 식으로 해서 배열(화장품 목록)의 중간에 빈 칸이 없도록 만든다. 삭제한 화장품은 할당된 동적 메모리를 해제시킨다.

### (3) 세부기능 3: 목록 불러오기

- 지금까지 사용자가 기록했던 목록을 불러온다.

## 3) 기능 3: 기초화장품 추천

- 새로 산 기초화장품이 사용자의 피부와 잘 맞지 않다면 과거 이력을 바탕으로 새로운 성분의 화장품을 추천해주는 기능 (기능1, 기능2번과 관련이 되어있다.)

### (1) 세부 기능 1

- 사용 이력 저장소에서 안 맞았던 화장품이 있다면 그 성분을 사용자로부터 입력받는다. 그후 그 성분 대신 새로운 성분의 다른 화장품을 추천해준다. 이때 새로운 성분의 화장

품 데이터들은 파일 입출력을 이용할 것이다. txt파일(데이터베이스)에 화장품들 이름과 대표성분을 미리 저장해둘 것이다. 만약 안 맞았던 화장품이 없거나 데이터베이스에 사용자로부터 입력받은 성분이 없으면 피부타입에 기반한 화장품을 추천해준다.

#### **4) 기능 4: 자신의 피부에 대해 기록하고 돌아보기**

- 사용자의 피부에 관련해 기록할 수 있는 기능

(1) 세부 기능 1: 오늘의 한줄일기

- 여드름이 더 났다거나 좀 진정됐다거나 세수를 안 했다가 등등 사용자가 특이사항을 기록할 수 있게 한다. (파일 입출력 사용!) 그리고 쓴 일기들을 열어볼 수 있도록 한다.

(2) 세부 기능 2: 약 기록 기능 + 먹는 날 알려주기

- 여드름약을 바르고 있거나 먹는 사용자들이 있을텐데 이들에 한번 먹거나 해야 하는 경우가 있는데 오늘이 먹는 날인지 헷갈릴 수 있으니까 날짜랑 복용 여부를 기록하게 해서 오늘이 먹는 날인지 알려주거나 다음 먹을 날을 계산해서 알려준다.

#### **5) 기능 5: 화장품 구매 사이트 바로가기**

- 화장품 구매 사이트인 올리브영의 링크를 출력해서 사용자가 바로갈 수 있게 만든다. (편리성을 높였다)

#### **6) 기능 6: 정보 출력**

- 여드름에 좋은 성분(티트리, 시카)들이나 기초화장품 바르는 순서, 좁쌀과 화농성의 차이 등등 정보를 출력해준다.

#### **7) 기능 7: 프로그램 종료**

- 종료문구와 함께 프로그램을 종료한다.

#### **8) 기능 8: 사용자로부터 작업 요청받기**

- if문을 사용하여 사용자가 기능 1~7에 해당하는 번호를 선택하면 그 번호로 이동할 수 있도록 만든다. 그 밖의 번호를 입력하면 에러 메시지를 출력한다.

### 3. 진척사항

#### 1) 기능 구현

##### (1) 기능8: 사용자로부터 작업 요청받기

- 입출력: med\_check값이 내려와서 if문을 작동시킴. 출력은 사용자가 입력해서 choice에 저장된 숫자를 밑에 있는 if문으로 전달해서 밑의 if문들을 작동하게 만든다.
- 설명: choice는 사용자가 고른 작업의 번호를 저장하기 위해서 만든 정수형 변수이다. 사용자가 입력한 값이 저장되어진다. 또한 메뉴들은 프로그램이 시작되면 바로 출력되어져서 사용자가 보고 원하는 것을 고를 수 있게 만들었다. while문 안에 있기 때문에 사용자가 프로그램 종료 메뉴를 선택하기 전까지 무한 반복된다. 이때, med\_check 변수를 통해서 출력을 분기시켰다. med\_check는 약을 매일 먹는다면 1이, 약을 이틀에 한번 먹는다면 2라는 값을 가진다. 만약 약을 안먹거나 매일 먹는다면 7. 프로그램 종료가 메뉴의 끝이고, 이틀에 한번 먹는다면 7. eat& splash 8. 프로그램 종료가 출력된다. 또한 약을 매일 먹는다면 메뉴의 밑에 항상 약 먹는(바르는)날이 맞다고 출력이 되고, 이틀이라면 다음 날짜를 계산해서 먹는 날짜가 출력이 되도록 만들었다.
- 적용된 배운 내용: 조건문, 입출력함수, 반복문

```
while (1) {
    int choice = 0;

    printf("-----\n");
    printf("번호를 선택해주세요!\n");
    printf(
        "1. 피부타입 진단하기\n2. 나의 화장품 서랍\n3. 오늘의 피부 일기\n4. "
        "올리브영 바로가기\n5. 여드름 관련 정보 모음\n");
    printf("6. 약 설정\n");
    if (med_check == 2) {
        printf("7. eat & splash\n8. 프로그램 종료\n");
    } else {
        printf("7. 프로그램 종료\n");
    }
    if (med_check == 1) {
        printf("*오늘은 약 먹는(바르는) 날이 맞습니다*\n");
    } else if (med_check == 2) {
        printf("*약을 먹어야(바라야) 하는 다음 날짜는 %d월 %d일입니다.*\n", month,
            day);
    }

    printf("-----\n");
    scanf_s("%d", &choice);
```

## (2) 기능1: 피부타입 진단

- 입출력: choice에 사용자가 입력해서 저장된 1이 내려옴. ErrorDetect 함수의 매개변수로 입력받는 값은 사용자 입력값과 목록의 최대숫자이다. 반환값은 check이다. 1또는 0이 반환된다. skin\_type이 다음 블록들로 내려가서 사용된다.
- 설명: 조건문을 활용해 사용자가 입력한 값이 1일 경우 이 코드블럭으로 이동할 수 있게 만들었다. 피부타입 진단을 위한 기초정보와 설명을 출력하게 하였다. int type[5]배열을 만들어서 사용자 입력값인 1또는 2를 배열칸에 저장한다. 이때 사용자가 1 또는 2 이외의 값을 입력하면 오류가 뜨도록 ErrorDetect 함수를 호출하였다. (이렇게 해야 sum 계산해서 출력할 때 제대로 된 정보가 출력됨) 이 함수는 사용자입력값, 최대숫자를 받아서 조건문으로 최대숫자보다 큰지 또는 0보다 작은지 확인한 뒤, check를 반환하고 이것은 while(check)에 사용된다. while문은 1이면 참이라 계속 반복되는데 올바른 입력을 할 경우 함수에서 0이 반환되어 반복이 종료되게 만들었다. 그 후 반복문을 이용하여 배열칸의 값(사용자가 선택한 입력값)들을 모두 더한뒤 그 합에 따라 지성건성중성으로 분류출력 코드를 작성했다. 1번 보기는 지성 특징이고 2번 보기는 건성 특징이기 때문에 합으로 분류를 하였다. 그리고 문자열 복사함수로 skin\_type에 피부타입("지성", "복합성", "건성")을 저장할 수 있게 만들었다.
- 적용된 배운 내용: 조건문, 입출력함수(printf, scanf\_s), 반복문(while, for), 함수, 포인터, 배열, 문자열, 헤더파일

```
int type[5];
```

```
int check = 1;
```

```
int ErrorDetect(int user_num, int limit_num) {  
    int check=1;  
    if (user_num > limit_num || user_num <= 0) {  
        check = 1;  
        printf("잘못된 입력입니다. 1~%d 사이의 숫자를 입력해주세요!\n", limit_num);  
        // 조건파악  
    } else {  
        check = 0;  
    }  
    return check;  
}
```

```

if (choice == 1) {
    int sum = 0;
    printf("피부타입 진단을 시작할게요 :)#n");
    printf("피부타입은 크게 지성, 복합성(중성), 건성으로 나누어져요!#n");
    printf("다음 보기들을 보고 자신의 피부와 가까운 것을 고르면 됩니다.#n");
    do {
        printf(
            "1. 세수하고 나와서 아무것도 안바른 상태로 시간이 지나도 "
            "건조하지가 "
            "않다. 2. 세수하고 아무것도 안바른다면 얼굴이 건조하고 당기는 "
            "편이다.#n");
        scanf_s("%d", &type[0]);
        check = errordetect(*(type + 0), 2);
    } while (check);
    do {
        printf(
            "1. 오후가 되면 얼굴에 기름이 뜨고 반짝거린다. 2. 오후가 되어도 "
            "아침과 피부상태가 똑같거나 기름이 거의 없다.#n");
        scanf_s("%d", &type[1]);
        check = errordetect(*(type + 1), 2);
    } while (check);
    do {
        printf(
            "1. 가을겨울에도 피부의 건조함을 느끼지 못한다. 2. 가을겨울은 "
            "피부가 "
            "다른 계절보다 건조해진다.#n");
        scanf_s("%d", &type[2]);
        check = errordetect(*(type + 2), 2);
    } while (check);
}

```

```

do {
    printf(
        "1. 기름종이를 사용했을 때 기름이 잔뜩 흡수된다. 2. 기름종이를 "
        "썼는데 기름종이가 뽀송하다. (기름종이가 없다면 깨끗한 손으로 "
        "얼굴을 만져보고 묻은 양을 확인해보세요)#n");
    scanf_s("%d", &type[3]);
    check = errordetect(*(type + 3), 2);
} while (check);
do {
    printf("1. 여드름이 많이 나있다. 2. 여드름이 없는 편이다.#n");
    scanf_s("%d", &type[4]);
    check = errordetect(*(type + 4), 2);
} while (check);

for (int i = 0; i < 5; i++) {
    sum += *(type + i);
}

```

```

if (sum <= 6) {
    printf("피부 타입 진단 결과는 지성입니다!\n");
    printf(
        "지성은 일반적으로 유분(기름)이 많은 피부입니다. 꾸덕하거나 잘 "
        "흡수되지 않는 무거운 기초화장품을 사용하면 여드름이 더 심해지니 "
        "주의하세요! 또한 기름종이 사용을 추천합니다.\n");
    strcpy_s(skin_type, (int)sizeof(skin_type), "지성");
} else if (sum <= 8) {
    printf("피부 타입 진단 결과는 중성(복합성)입니다!\n");
    printf(
        "중성은 기름이 너무 많이 나오지도, 너무 적지도 않은 피부입니다. "
        "따라서 지금 상태를 유지하는 것이 좋습니다. 시중의 화장품 대부분이 "
        "잘 맞을 것으로 보입니다.\n");
    strcpy_s(skin_type, (int)sizeof(skin_type), "복합성");
} else {
    printf("피부 타입 진단 결과는 건성입니다!\n");
    printf(
        "건성은 얼굴에 기름이 별로 나오지 않는 피부입니다. 그렇기에 보습을 "
        "잘 해주는 것이 중요합니다. 피부가 건조해지지 않도록 꼭 "
        "기초화장품을 발라주세요.\n");
    strcpy_s(skin_type, (int)sizeof(skin_type), "건성");
}

```

### (3) 기능2: 화장품 사용 이력 저장소의 세부기능1 – 작성하기(수정하기)

- 입출력: choice 2가 내려와서 이 블록을 실행하게 된다. AddHwalInfo 함수는 struct HwalInfo 타입의 포인터를 입력으로 받는다. 반환값은 void이기 때문에 없다. Change HwalInfo 함수도 마찬가지로 struct HwalInfo 타입의 포인터를 입력으로 받는다. 대신 숫자도 입력받아서 브랜드, 이름, 별점, 메모 중 수정할 것을 고를 수 있게 하였다. 반환값은 void이기 때문에 없다. ErrorDetect 함수의 매개변수로 입력받는 값은 사용자 입력값과 목록의 최대숫자이고 반환값은 check이다.

- 설명: 구조체를 사용하여 화장품브랜드, 이름, 평점을 저장할 수 있게 만들었다. 그리고 hwa\_save 배열을 만들어서 브랜드랑 이름이랑 평점이랑 메모를 저장하고, 최대 30개까지 화장품을 작성할 수 있게 만들었다. 그리고 화장품을 추가할지 나갈지 고르게 하고 1번을 선택하면 AddHwalInfo함수를 호출하여 사용자가 브랜드, 이름, 평점을 입력할 수 있게 만들었다. 이때 동적 메모리 할당을 사용해서 낭비없이 저장할 수 있게 했다. strcpy로 복사하였다. 그리고 화장품이 추가되었으니 개수(count)를 1 더한다. 수정 코드는 화장품 저장소가 비어있다면 수정할 것이 없으니 화장품이 없다고 출력되게 했고, 나머지의 경우라면 수정할 화장품의 번호를 입력받아서 수정할 항목을 선택하게 한 뒤 수정내용을 입력받을 수 있게 했다. 이때 중요한 점은 사용자가 입력한 수정할 화장품 번호가 sujeong에 저장되었는데 저장소에서의 인덱스는 0번부터기 때문에 sujeong-1을 했다는



점이 중요하다.

- 적용된 배운 내용: 조건문, 입출력함수(printf, scanf\_s), 배열, 구조체, 함수, 포인터, 동적 메모리, 반복문, 헤더파일

```
struct HwalInfo {
    char* brandname;
    char* name;
    float star;
    char* write;
};

struct HwalInfo hwa_save[];

void AddHwalInfo(struct HwalInfo* hwa);
void DisplayHwalInfo(const struct HwalInfo* hwa, int count);
void ChangeHwalInfo(struct HwalInfo* hwa, int num);
void SubHwalInfo(struct HwalInfo* hwa, int count, int delete);
void Freemalloc(struct HwalInfo* hwa, int count);
```

```
else if (choice == 2) {
    int num = 0;

    DisplayHwalInfo(hwa_save, count);
    printf("-----\n");

    printf(
        "1. 화장품 추가\n2. 화장품 수정\n3. 화장품 삭제\n4. 화장품 "
        "추천 (BETA)\n5. 나가기\n");
    check = 1;
    while (check) {
        scanf_s("%d", &num);
        check = ErrorDetect(num, 5);
    }

    if (num == 1) {
        if (count < BUNHO) {
            AddHwalInfo(&hwa_save[count]); // 포인터
            count++;
        } else {
            printf("화장품 저장소가 꽉 찼어요. (최대 30개까지 저장 가능)\n");
        }
    } else if (num == 2) {
        int sujeong = 0;
        if (count == 0) {
            printf("화장품 저장소에 화장품이 없어요. \n");
        } else {
            printf("수정할 화장품 번호: ");
        }
    }
}
```

```

check = 1;
while (check) {
    scanf_s("%d", &sujeong);
    check = ErrorDetect(sujeong, count);
}

int num = 0;

check = 1;
while (check) {
    printf("수정할 항목을 선택하세요.\\n");
    printf("1. 화장품 브랜드 2.화장품 이름 3.별점 4. 메모\\n");
    scanf_s("%d", &num);
    check = ErrorDetect(num, 4);
}

ChangeHwaInfo(&hwa_save[sujeong - 1], num);

```

```

void AddHwaInfo(struct HwaInfo* hwa) {
    printf("화장품 브랜드를 입력하세요.(띄어쓰기 없이 입력): ");
    char temp[300];
    scanf_s("%s", temp, (int)sizeof(temp));
    hwa->brandname = (char*)malloc((strlen(temp) + 1) * sizeof(char));
    strcpy_s(hwa->brandname, strlen(temp) + 1, temp);

    printf("화장품 이름을 입력하세요.(띄어쓰기 없이 입력): ");
    scanf_s("%s", temp, (int)sizeof(temp));
    hwa->name = (char*)malloc((strlen(temp) + 1) * sizeof(char));
    strcpy_s(hwa->name, strlen(temp) + 1, temp);

    printf("별점을 입력하세요(5점 만점): ");
    scanf_s("%f", &hwa->star);

    printf("메모를 입력하세요(띄어쓰기 없이 입력): ");
    scanf_s("%s", temp, (int)sizeof(temp));
    hwa->write = (char*)malloc((strlen(temp) + 1) * sizeof(char));
    strcpy_s(hwa->write, strlen(temp) + 1, temp);
}

```

```

void ChangeHwaInfo(struct HwaInfo* hwa, int num) {
    char temp[300];
    if (num == 1) {
        printf("화장품 브랜드를 입력하세요.(띄어쓰기 없이 입력): ");
        scanf_s("%s", temp, (int)sizeof(temp));
        hwa->brandname = (char*)malloc((strlen(temp) + 1) * sizeof(char));
        strcpy_s(hwa->brandname, strlen(temp) + 1, temp);
    }
    if (num == 2) {
        printf("화장품 이름을 입력하세요.(띄어쓰기 없이 입력): ");
        scanf_s("%s", temp, (int)sizeof(temp));
        hwa->name = (char*)malloc((strlen(temp) + 1) * sizeof(char));
        strcpy_s(hwa->name, strlen(temp) + 1, temp);
    }
    if (num == 3) {
        printf("별점을 입력하세요(5점 만점): ");
        scanf_s("%f", &hwa->star);
    }
    if (num == 4) {
        printf("메모를 입력하세요(띄어쓰기 없이 입력): ");
        scanf_s("%s", temp, (int)sizeof(temp));
        hwa->write = (char*)malloc((strlen(temp) + 1) * sizeof(char));
        strcpy_s(hwa->write, strlen(temp) + 1, temp);
    }
}

```

#### (4) 기능2: 화장품 사용 이력 저장소의 세부기능2 - 삭제하기

- 입출력: num이 3일 때 내려와서 블록이 실행된다. SubHwaInfo 함수는 구조체 배열과 배열에 칸 들어있는 값의 개수(화장품 개수)와 사용자가 입력한 삭제할 번호에서 1 뺀 값을 입력받는다. 반환값은 void이기 때문에 없다. ErrorDetect 함수의 매개변수로 입력받는 값은 사용자 입력값과 목록의 최대숫자이고 반환값은 check이다. Freemalloc 함수는
- 설명: 사용자가 삭제할 화장품 번호를 입력하면 delete에 저장이 된다. 만약 화장품 저장소에 화장품이 1개라도 있을 때, 화장품 개수보다 큰 값을 입력하면 에러메시지를 출력하도록 하였다. 화장품 저장소가 아예 아무것도 없이 비어있을 경우는 화장품이 없어서 삭제를 못한다는 의미의 문구를 출력하도록 코드를 작성했다. 그리고 SubHwaInfo 함수를 호출하여 삭제를 진행한다. delete 값으로 삭제할 브랜드 이름과 화장품 이름 구조체에 접근을 해서 삭제한다는 문구를 출력한뒤 할당된 동적 메모리를 해제했다. 그 후에는 목록 중간에 구멍이 뚫리지 않도록 삭제할 번호부터 시작해서 총 화장품 개수까지 반복을 진행하도록 만들어서 한칸씩 당겨와서 덮어씌웠다. (이러면 삭제 한 곳이 비어지지 않고 쭉 빈칸이 없이 저장됨) 그리고 원래의 마지막 칸은 깔끔하게 NULL로 비웠다. 그리고 반복문으로 삭제를 한 후에는 count에서 1을 빼서 화장품 개수를 올바르게 맞추

었다.

- 적용된 배운 내용: 조건문, 입출력함수(printf, scanf\_s), 반복문, 배열, 구조체, 함수, 포인터, 동적메모리, 헤더파일

```
} else if (num == 3) {  
    int delete = 0;  
    if (count > 0) {  
        printf("삭제할 화장품 번호: ");  
        check = 1;  
        while (check) {  
            scanf_s("%d", &delete);  
            check = ErrorDetect(delete, count);  
        }  
        SubHwalInfo(hwa_save, count, delete - 1);  
        count = count - 1;  
    } else {  
        printf("화장품 저장소에 화장품이 없어요.\n");  
    }  
}
```

```
void SubHwalInfo(struct HwalInfo* hwa, int count, int delete) {  
    printf("%s | %s 화장품을 삭제할게요.\n", hwa[delete].brandname,  
        hwa[delete].name);  
  
    free(hwa[delete].brandname);  
    free(hwa[delete].name);  
    free(hwa[delete].write);  
  
    for (int i = delete; i < count; i++) {  
        hwa[i] = hwa[i + 1];  
    }  
    hwa[count - 1].brandname = NULL;  
    hwa[count - 1].name = NULL;  
    hwa[count - 1].write = NULL;  
}
```

## (5) 기능2: 화장품 사용 이력 저장소의 세부기능3 - 목록 불러오기

- 입출력: choice 2가 내려와서 이 블록을 실행하게 된다. DisplayHwalInfo함수는 구조체 배열과 배열에 칸 들어있는 값의 개수(화장품 개수)를 입력받는다. 반환값은 void이기 때문에 없다.
- 설명: 사용자가 2번을 입력하면 화장품 목록이 바로 뜨게된다. 구조체를 사용하여 화장품 브랜드, 이름, 평점을 저장할 수 있게 만들었다. 그리고 hwa\_save 배열을 만들어서 브랜드랑 이름이랑 평점을 저장하고, 최대 30개까지 화장품을 저장할 수 있게 만들었다.

count에는 화장품의 개수가 저장되도록 했다. 이때 빈 배열은 출력이 되지 않도록 반복문을 사용해 화장품 개수만큼만 딱 정확히 목록 출력이 되게 만들었다. 화장품 브랜드, 이름, 별점, 메모가 순서대로 출력된다. 별점은 소수점 첫째자리까지 출력되도록 했다.

- 적용된 배운 내용: 조건문, 입출력함수(printf, scanf\_s), 반복문, 배열, 구조체, 함수, 포인터, 동적메모리, 헤더파일

```
struct HwalInfo {
    char* brandname;
    char* name;
    float star;
    char* write;
};

struct HwalInfo hwa_save[];

void AddHwalInfo(struct HwalInfo* hwa);
void DisplayHwalInfo(const struct HwalInfo* hwa, int count);
```

```
} else if (choice == 2) {
    int num = 0;

    DisplayHwalInfo(hwa_save, count);
    printf("-----\n");
```

```
void DisplayHwalInfo(const struct HwalInfo* hwa, int count) {
    printf("[화장품 목록]\n");
    for (int i = 0; i < count; i++) {
        printf("%d. %s | %s (%.1f/5)\n", i + 1, hwa[i].brandname, hwa[i].name,
            hwa[i].star);
        printf("%s\n", hwa[i].write); // 한줄씩 출력
    }
}
```

## (6) 기능3: 기초화장품 추천

- 입출력: choice가 2일 때 num이 4면 이 블록이 실행된다. SkinTypeHwa 함수는 피부타입 포인터를 입력받고 반환값은 void이기 때문에 없다.

- 설명: 기초 화장품을 추천하는 메뉴이다. 먼저 화장품 저장소에서 맞지 않았던 화장품 성분이 있는지 여부를 입력받고 만약 있다면 성분을 입력받아 component에 넣는다. 문자열 비교함수로 사용자가 입력한 성분과 티트리 혹은 병풀이 일치하면 파일입출력을 사용해서 새로운 성분의 화장품을 추천한다. 만약 티트리를 입력했다면 병풀 화장품을 저장할 파일을 열고 사용자 컴퓨터에 저장한 뒤(데이터 베이스 느낌) 다시 닫고 읽기모드로 열어 새로운 성분의 화장품 목록을 출력하도록 했다. 병풀을 입력했을 경우에도 마찬가지로

가지이다. 만약 사용자가 입력한 성분이 없다면 추천 데이터베이스에 없다는 문구를 출력하고 그 대신 사용자의 피부타입에 따른 화장품 추천을 하도록 했다. SkinTypeHwa 함수를 호출하고 이때 피부타입 진단을 하지 않아서 skin\_type이 비어있다면 먼저 피부타입 진단을 하면 추천해준다고 출력한다. 그 외에는 문자열 비교로 지성, 복합성, 건성으로 분류해서 그에 맞는 추천 화장품을 출력하도록 했다.

```
else if (num == 4) {
    char one;
    char component[15];
    printf("기초 화장품 추천 탭입니다!\n");
    printf(
        "화장품 저장소에서 맞지 않았던 화장품 성분이 있다면 y를 없으면 n를 "
        "입력해주세요.\n");
    scanf_s(" %c", &one, 1);
    getchar();
    if (one == 'y' || one == 'Y') {
        printf("맞지 않았던 성분 한가지를 입력해주세요: ");
        scanf_s("%s", component, (int)sizeof(component));

        if (strcmp(component, "티트리") == 0) {
            char* fileName2 = "./best_hwa_byeongful.txt";
            FILE* file3;

            printf("%s 성분 대신 [병풀] 성분의 화장품을 추천드립니다\n",
                component);
            fopen_s(&file3, fileName2, "w");
            fputs(
                "1. 라운드랩 소나무 진정 앰플\n2. 토리든 밸런스풀 병풀 진정 "
                "크림\n",
                file3);
            fclose(file3);
            fopen_s(&file3, fileName2, "r");

            while (1) {
                char* result = fgets(temp, MAX, file3);
                if (result == NULL) {
                    break;
                }
            }
        }
    }
}
```

```

    printf("%s", temp);
}
fclose(file3);
} else if (strcmp(component, "병풀") == 0) {
    FILE* file2;
    fopen_s(&file2, "./best_hwa_teetree.txt", "w");
    if (file2 != NULL) {
        printf("%s 성분 대신 [티트리] 성분의 화장품을 추천드립니다!\n",
            component);
        fputs(
            "1. 메디힐 티트리 진정 수분 앰플\n2. 린제이 모델링마스크 "
            "쿨티트리\n3. 브링그린 티트리수딩토너\n4. 파파레서피 "
            "티트리 "
            "수딩 크림\n",
            file2);
        fclose(file2);
        fopen_s(&file2, "./best_hwa_teetree.txt", "r");

        while (1) {
            char* result = fgets(temp, MAX, file2);
            if (result == NULL) {
                break;
            }
            printf("%s", temp);
        }
        fclose(file2);
    }
} else {
    printf(
        "성분이 추천 데이터베이스에 존재하지 않습니다. 그 대신 "
        "피부타입에 따른 화장품을 추천해드릴게요!\n");
    SkinTypeHwa(skin_type);
}

else {
    printf("그렇다면 피부타입에 따른 화장품을 추천해드릴게요!\n");
    SkinTypeHwa(skin_type);
}

```

```

void SkinTypeHwa(char* skin_type) {
    if (strcmp(skin_type, "지성") == 0) {
        printf("[지성 피부]\n");
        printf("1. 독도토너\n2. 에스네이처 아쿠아 오아시스 수분크림\n");
    } else if (strcmp(skin_type, "복합성") == 0) {
        printf("[중성(복합성) 피부]\n");
        printf(
            "1. 이즈엔트리 초저분자 히알루론산 토너\n2. 메이크프렘 인테카 "
            "수분크림\n");
    } else if (strcmp(skin_type, "건성") == 0) {
        printf("[건성 피부]\n");
        printf("1. 일리운 더마 스킨\n2. 토리든 솔리드인 세라마이드 크림\n");
    } else {
        printf("먼저 피부타입 진단을 하시면 그에 맞는 화장품을 추천드려요.\n");
    }
}
}

```

(7) 기능4: 자신의 피부에 대해 기록하고 돌아보기의 세부기능1 - 오늘의 한줄 일

## 기

- 입출력: 위에서 사용자가 입력했던 choice 3이 내려옴. txt파일이 만들어지고 이어서 써진다.
- 설명: 파일입출력을 사용하여 hwa\_diary.txt 파일을 만들고 거기에 사용자가 쓴 일기를 저장해서 출력이 가능하게 만들었다. 피부일기 쓰기, 보기로 메뉴를 출력해서 고르도록 했고 월과 일은 간편하게 숫자만 입력하도록 했다. 사용자가 일기를 쪽 쓰다가 다 썼으면 '나가기'를 입력하여 나갈 수 있게 만들었다. 2번을 입력하면 날짜와 지금까지 썼던 일기가 모조리 다 출력되도록 만들었다. a+를 사용해서 처음에는 파일이 없을 테니 만들고, 그 후에는 일기가 전에 썼던 내용에 이어서 작성되도록 하였다.
- 적용된 배운 내용: 조건문, 입출력함수(printf, scanf\_s), 파일입출력, 반복문, 헤더파일

```
else if (choice == 3) {
    int num = 0;

    fopen_s(&file, fileName, "a+");
    if (file == NULL) {
        printf("실패\n");
        return 1;
    }
    printf("1. 피부 일기 쓰기\n2. 피부 일기 보기\n");
    check = 1;
    while (check) {
        scanf_s("%d", &num);
        check = ErrorDetect(num, 2);
    }

    if (num == 1) {
        int month = 0;
        int day = 0;
        printf("월을 입력하세요.(1~12)\n");
        scanf_s("%d", &month);
        printf("일을 입력하세요.(1~31)\n");
        scanf_s("%d", &day);
        printf("%d월 %d일", month, day);
        fprintf(file, "%d월 %d일 ", month, day);
        printf("일기 입력(작성이 완료되면 나가기를 입력하세요): ");
        while (1) {
            scanf_s("%[^\n]", content, (int)sizeof(content));
            getchar();

            if (strcmp(content, "나가기") == 0) {
                printf("일기가 저장되었습니다.\n");
                break;
            }
            fputs(content, file);
            fputs("\n", file);
        }
        fclose(file);
    }
}
```



```

} else if (num == 2) {
    while (1) {
        char* result = fgets(temp, MAX, file);
        if (result == NULL) {
            break;
        }
        printf("%s", temp);
    }
    fclose(file);
}

```

#### (8) 기능4: 자신의 피부에 대해 기록하고 돌아보기의 세부기능2 - 약 기록 기능 + 먹는 날 알려주기

- 입출력: MedCalCulator 함수는 포인터 변수 3개를 매개변수로 입력받는다. med\_check는 약 복용 변수, month와 day는 각각 월과 일을 가리킨다. 반환값은 void이기 때문에 없다.

- 설명: 프로그램이 시작되면 초기 설정 단계로 바로 여드름약 복용 여부를 입력하게 해서 약 알리미 기능에 활용한다. 만약 지금 당장은 약을 먹지 않더라도 나중에 먹을 수 있기 때문에 6. 약 설정 메뉴(choice == 6)에서 추후에도 설정할 수 있도록 만들었다. MedCalCulator 함수를 호출해서 여드름약 복용 여부를 입력받고 y(복용을 한다)를 입력하면 약 먹는(바르는) 주기를 선택하도록 한다. 만약 매일 복용한다면 med\_check에 1의 값을 저장, 이틀이라면 오늘 날짜 정보를 수집해서 날짜 계산에 활용하고 med\_check에 2의 값을 저장하고 기타의 경우는 지원하지 않았다. 여드름약은 매일아니면 이틀 복용이 99%이기 때문에 이 둘만 지원하기로 했다. 기타와 아예 복용하지 않는 경우에는 med\_check에 0을 저장한다.

이때 만약 이틀의 주기로 약을 복용한다면 전체 메뉴 밑에 약을 먹어야(발라야)하는 다음 날짜가 출력되는데 사용자가 약을 오늘 먹어서 7번(choice==7)을 입력하면 날짜가 계산이 되도록 만들었다. month와 day는 사용자가 입력한 날짜가 저장되어있는 상태다. 이틀 주기이기 때문에 day에는 2를 더하도록 만들었다. 이때 미리 달의 날짜수를 month\_day 배열에 담아놓았다. 그래서 그 달에 담긴 날짜수보다 day가 크면 day를 1로 만들고 달에는 1을 더해서 다음달로 넘어가게 만들었다. 만약 달이 12월을 넘어가면 다시 1월로 시작되게 만들었다.

- 적용된 배운 내용: 입출력함수(printf, scanf\_s), 조건문, 함수, 포인터, 헤더파일

```
int med_check = 0;
int month, day;
int month_day[12] = {31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31};
```

```
int main() {
    char* fileName = "./hwa_diary.txt"; // 월
    FILE* file;
    printf("피부관리 프로그램 ver1.0 \n");
    printf("초기 설정 단계입니다!\n");
    MedCalculator(&med_check, &month, &day);
}
```

```
} else if (choice == 6) {
    MedCalculator(&med_check, &month, &day);
}
if (med_check == 2) {
    if (choice == 7) {
        day += 2;

        int EndDay = month_day[month - 1];
        if (day > EndDay) {
            day -= EndDay; // 1일로 만들기
            month++;
            if (month > 12) {
                month = 1;
            }
        }
    }
}
```

```
void MedCalculator(int* med_check, int* Month, int* Day) {
    char one = ' ';

    printf(
        "여드름약을 먹거나 바르고 계신가요? 맞다면 y를, 틀리면 n을 입력해주세요. "
        "이 정보는 약 알리미 기능에 사용됩니다!\n");
    scanf_s("%c", &one, 1);
    getchar();
    if (one == 'y' || one == 'Y') {
        int med;
        printf("약 먹는(바르는) 주기가 어떻게 되나요?\n1. 매일 2. 이틀 3. 기타\n");
        scanf_s("%d", &med);

        if (med == 1) {
            printf("답변 감사합니다 :) \n");
            *med_check = 1;
        } else if (med == 2) {
            printf("답변 감사합니다 :) \n");
            printf("날짜 정보를 수집할게요! \n");
            printf("오늘 월을 입력하세요: ");
            scanf_s("%d", Month);
            printf("일을 입력하세요: ");
            scanf_s("%d", Day);
            *med_check = 2;
            printf(
                "앞으로 약을 먹거나 발랐다면 메뉴 번호인 7번을 입력해주세요! 약 "
                "알리미에 날짜 계산이 실시간 "
                "반영되게 됩니다. \n");
        } else if (med == 3) {
            printf(
                "매일, 이틀 외는 알림기능을 지원하고 "
                "있지 않습니다. \n");
            *med_check = 0;
        } else {
            printf("잘못된 입력입니다. 약 설정 메뉴에서 다시 설정해주세요. \n");
        }
    } else {
        *med_check = 0;
    }
}
```

## (9) 기능5: 화장품 구매 사이트 바로가기

- 입출력: 위에서 사용자가 입력했던 choice 4가 내려옴. 마찬가지로 이 코드블럭 밑으로 전달되는 것은 없다.
- 설명: 사용자가 4를 입력하면 이 블럭으로 오게됨. 올리브영 링크를 출력하여 사용자가 편하게 복사해서 원하는 때에 이동할 수 있게 만들었다.
- 적용된 배운 내용: 입력함수(printf), 조건문

```

} else if (choice == 4) {
    printf(
        "올리브영 링크입니다! (링크 바로가기는 추후 업데이트 예정입니다) "
        "복사(ctrl+c) 후 주소 창에 붙여넣기(ctrl+v) 하세요\n");
    printf("www.oliveyoung.co.kr\n");
} else if (choice == 5) {

```

## (10) 기능6: 정보 출력

- 입출력: 위에서 choice 값을 받음. 이게 5이면 이 코드블럭이 실행된다. 끝나면 아래쪽으로 보내는 변수는 딱히 없다고 볼 수 있다. 코드블럭 내에서의 문구들이 잘 출력된다.
- 설명: choice가 5(사용자 입력5)이면 printf로 정보들의 목록이 좌르륵 출력된다. 이때 사용자로부터 원하는 정보의 번호를 입력받고, if문을 사용해서 번호에 맞는 정보를 출력한다. 이때 정보 번호인 1, 2, 3 외의 번호를 입력한다면 잘못된 입력임을 알려준다.
- 적용된 배운 내용: 조건문, 입출력함수(printf, scanf\_s)

```

} else if (choice == 5) {
    int num = 0;
    printf("(1) 여드름에 좋은 화장품 성분들\n");
    printf("(2) 기초화장품 바르는 순서?\n");
    printf("(3) 쏘팔여드름과 화농성 여드름의 차이\n");
    scanf_s("%d", &num);
    if (num == 1) {
        printf(
            "'티트리' - 티트리는 진정에 매우 탁월하여 예민해진 피부에 도움을 "
            "줄 수 있습니다.\n");
        printf(
            "'병풀' - 병풀은 상처입은 호랑이가 병풀이 있는 곳에서 뒹굴어 "
            "'호랑이 풀'이라고도 불리며 마데카식애씨드 등의 성분이 피부 진정과 "
            "수분관리에 도움을 줍니다.\n");
    } else if (num == 2) {
        printf(
            "기초화장품은 가벼운 것부터 점차 무겁고 꾸덕한 순서로 바르면 "
            "됩니다! 간단하게 설명하면 [스킨(토너) - 앰플 - 로션 or 수분크림 - "
            "스팟연고] 순서로 바르면 됩니다. ");
        printf(
            "바를 때는 뽀뽀 힘줘서 바르면 안되고 살살 발라야 자극이 덜해서 "
            "트러블을 막을 수 있습니다.\n");
    } else if (num == 3) {
        printf(
            "쏘팔 여드름은 하얗고 안에 쌀알같은 것이 차있는 것이고 화농성 "
            "여드름은 염증이 생겨서 빨갛고 만지면 아프고 부어올라있는 "
            "여드름입니다.\n");
    } else {
        printf(
            "잘못된 입력입니다. 목록에 있는 번호를 입력해 주세요.\n"); // +
    }
}

```

## (11) 기능7: 프로그램 종료

- 입출력: 위에서 choice라는 값이 7또는 8이라는 것을 받음. return; 이 있어서 더 이상 while문을 돌지 않고(다음으로 진행X) 곧바로 프로그램 종료. Freemalloc의 입력받는 매개 변수는 구조체, count(화장품 개수)를 입력받고 void형 함수이기 때문에 반환하는 값은 없다.
- 설명: med\_check는 위에서 설명한 것처럼 사용자 약 복용과 관련된 변수이다. med\_check가 2일 때 즉, 약을 이틀에 한번 복용한다면 choice가 7번일 때 약 복용 체크 메뉴이고 8번이 프로그램 종료이다. 그 외의 경우라면 7번이 프로그램 종료이다. 메모리 해제 함수로 프로그램 종료 전에 구조체에서 할당됐던 메모리를 해제하도록 만들었다. Freemalloc함수에서 반복문을 이용해서 각각 저장됐던 화장품 브랜드와 이름, 메모에 동적할당된 메모리를 해제한다.
- 적용된 배운 내용: 조건문, 출력함수, 반복문, 함수, 동적메모리, 헤더파일

```

if (med_check == 2) {
    if (choice == 7) {
        day += 2;

        int EndDay = month_day[month - 1];
        if (day > EndDay) {
            day -= EndDay; // 1일로 만들기
            month++;
            if (month > 12) {
                month = 1;
            }
        }
    } else if (choice == 8) {
        printf("프로그램을 종료합니다.\n");
        Freemalloc(hwa_save, count);
        return 0;
    } else {
        ErrorDetect(choice, 8);
    }
} else {
    if (choice == 7) {
        printf("프로그램을 종료합니다.\n");
        Freemalloc(hwa_save, count);
        return 0;
    } else {
        ErrorDetect(choice, 7);
    }
}
}

```

```

void Freemalloc(struct HwalInfo* hwa, int count) {
    for (int i = 0; i < count; i++) {
        free(hwa[i].brandname);
        free(hwa[i].name);
        free(hwa[i].write);
    }
}

```

## <헤더파일>

```

struct HwalInfo {
    char* brandname;
    char* name;
    float star;
    char* write;
};

struct HwalInfo hwa_save[];

void AddHwalInfo(struct HwalInfo* hwa);
void DisplayHwalInfo(const struct HwalInfo* hwa, int count);
void ChangeHwalInfo(struct HwalInfo* hwa, int num);
void SubHwalInfo(struct HwalInfo* hwa, int count, int delete);
void Freemalloc(struct HwalInfo* hwa, int count);
void MedCalculator(int* MedCheck, int* month, int* day);
void SkinTypeHwa(char* SkinType);
int ErrorDetect(int user_num, int limit_num);

```

## 2) 테스트 결과

### (1) 기능8: 사용자로부터 작업 요청받기

- 프로그램이 시작되면 초기 설정 문구가 출력되면서 y, n을 잘 입력할 수 있는지 확인했다. 그리고 이들을 선택했을 때는 메뉴 번호가 7. eat & splash와 8. 프로그램 종료로 제대로 출력되는지 테스트하였다. 만약 이틀이 아닌 그 외의 경우(매일, 약X)는 7번까지 메뉴가 출력되는 걸 테스트했는데 잘 됐다. 입력 범위 에러 메시지도 잘 뜨는 것을 확인했다.

```
피부관리 프로그램 ver1.0
초기 설정 단계입니다!
여드름약을 먹거나 바르고 계신가요? 맞다면 y를, 틀리면 n을 입력해주세요. 이 정보는 약 알리미 기능에 사용됩니다!
y
약 먹는(바르는) 주기가 어떻게 되나요?
1. 매일 2. 이틀 3. 기타
2
답변 감사합니다 :)
날짜 정보를 수집할게요!
오늘 월을 입력하세요: 12
일을 입력하세요: 24
앞으로 약을 먹거나 발랐다면 메뉴 번호인 7번을 입력해주세요! 약 알리미에 날짜 계산이 실시간 반영되게 됩니다.
-----
번호를 선택해주세요!
1. 피부타입 진단하기
2. 나의 화장품 서랍
3. 오늘의 피부 일기
4. 올리브영 바로가기
5. 여드름 관련 정보 모음zip
6. 약 설정
7. eat & splash
8. 프로그램 종료
*약을 먹어야(발라야) 하는 다음 날짜는 12월 24일입니다.*
```

```
n
-----
번호를 선택해주세요!
1. 피부타입 진단하기
2. 나의 화장품 서랍
3. 오늘의 피부 일기
4. 올리브영 바로가기
5. 여드름 관련 정보 모음zip
6. 약 설정
7. 프로그램 종료
-----
8
잘못된 입력입니다. 1~7 사이의 숫자를 입력해주세요!
```

<- n (약X)을 입력했을 경우

### (2) 기능1: 피부타입 진단

- 1번을 입력했을 때 사용자에게 설명이 잘 출력되는지 확인하였다. 그리고 보기들을 보고 숫자를 입력했을 때 1, 2 외의 숫자를 입력하면 에러메시지가 잘 출력되는지, 최종 결과인 지성중성건성이 고른 결과에 따라 잘 출력되는지 확인하였다.

1  
 피부타입 진단을 시작할게요 :)  
 피부타입은 크게 지성, 복합성(중성), 건성으로 나누어져요!  
 다음 보기들을 보고 자신의 피부와 가까운 것을 고르면 됩니다.  
 1. 세수하고 나와서 아무것도 안바른 상태로 시간이 지나도 건조하지가 않다. 2. 세수하고 아무것도 안바른다면 얼굴이 건조하고 당기는 편이다.  
 3  
 잘못된 입력입니다. 1~2 사이의 숫자를 입력해주세요!  
 1. 세수하고 나와서 아무것도 안바른 상태로 시간이 지나도 건조하지가 않다. 2. 세수하고 아무것도 안바른다면 얼굴이 건조하고 당기는 편이다.  
 1  
 1. 오후가 되면 얼굴에 기름이 뜨고 반짝거린다. 2. 오후가 되어도 아침과 피부상태가 똑같거나 기름이 거의 없다.  
 1  
 1. 가을겨울에도 피부의 건조함을 느끼지 못한다. 2. 가을겨울은 피부가 다른 계절보다 건조해진다.  
 2  
 1. 기름종이를 사용했을 때 기름이 잔뜩 흡수된다. 2. 기름종이를 썼는데 기름종이가 뽕송하다. (기름종이가 없다면 깨끗한 손으로 얼굴을 만져보고 묻은 양을 확인해보세요)  
 1  
 1. 여드름이 많이 나있다. 2. 여드름이 없는 편이다.  
 1  
 피부 타입 진단 결과는 지성입니다!  
 지성은 일반적으로 유분(기름)이 많은 피부입니다. 꾸덕하거나 잘 흡수되지 않는 무거운 기초화장품을 사용하면 여드름이 더 심해지니 주의하세요! 또한 기름종이 사용을 추천합니다.

### (3) 기능2: 화장품 사용 이력 저장소의 세부기능1 – 작성하기(수정하기)

- 메뉴가 잘 출력되고 띄어쓰기 없이 입력하면 오류없이 잘 저장되는 것을 확인했다.

[화장품 목록]  
 -----  
 1. 화장품 추가  
 2. 화장품 수정  
 3. 화장품 삭제  
 4. 화장품 추천(BETA)  
 5. 나가기  
 1  
 화장품 브랜드를 입력하세요.(띄어쓰기 없이 입력): 토리든  
 화장품 이름을 입력하세요.(띄어쓰기 없이 입력): 수분광토퍼  
 별점을 입력하세요(5점 만점): 4.76  
 메모를 입력하세요(띄어쓰기 없이 입력): 수분이많음

<- 작성하기

2  
 수정할 화장품 번호: 1  
 수정할 항목을 선택하세요.  
 1. 화장품 브랜드 2.화장품 이름 3.별점 4. 메모  
 3  
 별점을 입력하세요(5점 만점): 5

<- 수정하기

### (4) 기능2: 화장품 사용 이력 저장소의 세부기능2 – 삭제하기

- 화장품 삭제 번호인 3번을 입력한 후 삭제할 화장품 번호를 입력하면 화장품이 잘 삭제되는지 확인하였다. 만약 삭제할 화장품이 없다면 없다고 출력되는지도 테스트하였다.

```
3
삭제할 화장품 번호: 1
토리든 | 수분광토너 化粧품을 삭제할게요.
```

```
2
[化粧품 목록]
-----
```

<- 삭제했더니 목록에서도 잘 지워지는 모습

```
[化粧품 목록]
-----
1. 化粧품 추가
2. 化粧품 수정
3. 化粧품 삭제
4. 化粧품 추천 (BETA)
5. 나가기
3
化粧품 저장소에 化粧품이 없어요.
```

<- 化粧품 저장소에 化粧품이 없을 경우 출력

## (5) 기능2: 化粧품 사용 이력 저장소의 세부기능3 - 목록 불러오기

- 化粧품 이름 저장 배열이 빈칸일 때는 출력이 안되는지 테스트하였는데 제대로 구현되었다. 化粧품 추가를 하고 수정했을때도 化粧품 브랜드, 이름이 잘 출력되었고 별점도 소수 첫째자리까지 잘 출력된 것을 확인했다. 메모도 사용자가 입력한 그대로 잘 출력되었다.

```
2
[化粧품 목록]
-----
```

<- 비어져있을 때

```
[化粧품 목록]
1. 토리든 | 수분광토너 (5.0/5)
수분이 많음
```

## (6) 기능3: 기초 化粧품 추천

- 化粧품 추천 메뉴를 선택하면 맞지 않았던 化粧품 성분이 있는지 없는지 입력받았다. 성분을 입력했을 때 데이터베이스(txt파일)에 있는 경우에는 그 성분 대신 다른 성분의 化粧품이 잘 출력되는지 확인했다. 만약에 맞지 않았던 化粧품 성분이 데이터베이스에 없는 경우에는 피부타입 기반 추천인데, 만약 사용자가 진단을 하지 않아서 정보가 없을 때는 먼저 하라는 메시지가 잘 출력되는지 확인하였다. 만약 진단을 해서 skin\_type에 피부타입이 담겨있으면 피부타입에 맞는 化粧품이 잘 출력되는지 확인했는데 잘 되었다.



```

1. 화장품 추가
2. 화장품 수정
3. 화장품 삭제
4. 화장품 추천 (BETA)
5. 나가기
4
기초 화장품 추천 탭입니다!
화장품 저장소에서 맞지 않았던 화장품 성분이 있다면 y를 없으면 n를 입력해주세요.
y
맞지 않았던 성분 한가지를 입력해주세요: 티트리
티트리 성분 대신 [병풀] 성분의 화장품을 추천드립니다
1. 라운드랩 소나무 진정 앰플
2. 토리든 밸런스풀 병풀 진정 크림

```

```

화장품 저장소에서 맞지 않았던 화장품 성분이 있다면 y를 없으면 n를 입력해주세요.
n
그렇다면 피부타입에 따른 화장품을 추천해드릴게요!
먼저 피부타입 진단을 하시면 그에 맞는 화장품을 추천드려요.

```

```

n
그렇다면 피부타입에 따른 화장품을 추천해드릴게요!
[지성 피부]
1. 독도토너
2. 에스네이처 아쿠아 오아시스 수분크림

```

#### (7) 기능4: 자신의 피부에 대해 기록하고 돌아보기의 세부기능1 - 오늘의 한줄 일기

- 3을 입력했을 때 오늘의 피부일기 메뉴로 이동을 잘 하는지 테스트했다. 그리고 쓰기, 보기가 잘 출력되는지와 사용자가 월과 일을 제대로 입력할 수 있는지 확인했다. 일기를 다 썼을 때 나가기를 입력하면 일기쓰기가 끝나고 잘 저장되는지도 테스트했다. 일기 보기를 했을때도 날짜와 함께 사용자가 쓴 일기가 제대로 저장되었다.

```

3
1. 피부 일기 쓰기
2. 피부 일기 보기
1
월을 입력하세요.(1~12)
12
일을 입력하세요.(1~31)
24
12월 24일일기 입력(작성이 완료되면 나가기를 입력하세요): 오늘은 피부가 좀 좋아진 것 같다^^
나가기
일기가 저장되었습니다.

```

```

3
1. 피부 일기 쓰기
2. 피부 일기 보기
2
12월 24일
오늘은 피부가 좀 좋아진 것 같다^^

```

#### (8) 기능4: 자신의 피부에 대해 기록하고 돌아보기의 세부기능2 - 약 기록 기능

## + 먹는 날 알려주기

- 시작하자마자 잘 뜨는지, 사용자가 잘 입력 가능한지 확인하였다. 그리고 처음에 n을 했어도 나중에 6. 약 설정 메뉴에 들어가면 다시 잘 설정이 되는지 확인하였다. 약을 이틀에 한번 먹는 사용자라고 가정하고 먹는 날 알려주기 기능도 테스트했는데 사용자가 7번을 입력하면 날짜가 2일이 잘 더해진 것을 확인할 수 있었다. 달도 잘 변경되는지 확인했는데 완벽하게 12월에서 1월로 변경되었다. 매일 먹는 사람의 경우도 출력이 잘 되는 것을 확인하였다.

```
피부관리 프로그램 ver1.0
초기 설정 단계입니다!
여드름약을 먹거나 바르고 계신가요? 맞다면 y를, 틀리면 n을 입력해주세요. 이 정보는 약 알리미 기능에 사용됩니다!
n
```

```
6
여드름약을 먹거나 바르고 계신가요? 맞다면 y를, 틀리면 n을 입력해주세요. 이 정보는 약 알리미 기능에 사용됩니다!
y
약 먹는(바르는) 주기가 어떻게 되나요?
1. 매일 2. 이틀 3. 기타
2
답변 감사합니다 :)
날짜 정보를 수집할게요!
오늘 월을 입력하세요: 12
일을 입력하세요: 24
앞으로 약을 먹거나 발랐다면 메뉴 번호인 7번을 입력해주세요! 약 알리미에 날짜 계산이 실시간 반영되게 됩니다.
```

```
*약을 먹어야(발라야) 하는 다음 날짜는 12월 24일입니다.*
-----
7
-----
번호를 선택해주세요!
1. 피부타입 진단하기
2. 나의 화장품 서랍
3. 오늘의 피부 일기
4. 올리브영 바로가기
5. 여드름 관련 정보 모음zip
6. 약 설정
7. eat & splash
8. 프로그램 종료
*약을 먹어야(발라야) 하는 다음 날짜는 12월 26일입니다.*
-----
```

<- 이틀 날짜 계산 테스트 사진

```
*약을 먹어야(발라야) 하는 다음 날짜는 12월 30일입니다.*
-----
7
-----
번호를 선택해주세요!
1. 피부타입 진단하기
2. 나의 화장품 서랍
3. 오늘의 피부 일기
4. 올리브영 바로가기
5. 여드름 관련 정보 모음zip
6. 약 설정
7. eat & splash
8. 프로그램 종료
*약을 먹어야(발라야) 하는 다음 날짜는 1월 1일입니다.*
-----
```

<- 달도 잘 넘어가는 모습

```
*오늘은 약 먹는(바르는) 날이 맞습니다*
```

<- 매일 먹는 사용자의 출력문구

## (9) 기능5: 화장품 구매 사이트 바로가기

- 4번을 입력했을 때 올리브영 링크와 이동방법이 잘 출력되는지 확인했는데 제대로 나왔다.

```
4
올리브영 링크입니다! (링크 바로가기는 추후 업데이트 예정입니다) 복사(ctrl+c) 후 주소 창에 붙여넣기(ctrl+v) 하세요
www.oliveyoung.co.kr
```

## (10) 기능6: 정보 출력

- scanf\_s로 사용자가 5번을 입력했을 때 정보 목록이 잘 뜨는지, 원하는 목록의 번호를 입력했을 때 정보 출력이 잘 되는지 테스트하였다.

```
5
1) 여드름에 좋은 화장품 성분들
2) 기초화장품 바르는 순서?
3) 좁쌀여드름과 화농성 여드름의 차이
1
'티트리' - 티트리는 진정에 매우 탁월하여 예민해진 피부에 도움을 줄 수 있습니다.
'병풀' - 병풀은 상처입은 호랑이가 병풀이 있는 곳에서 뒹굴어 호랑이 풀이라고도 불리며 마데카식애씨드 등의 성분이 피부 진
정과 수분관리에 도움을 줍니다.
```

## (11) 기능7: 프로그램 종료

- 사용자가 매일 약을 복용하거나 복용하지 않을 경우에는 7번을 입력했을 때 문구와 함께 프로그램이 제대로 종료되는지 테스트하였다. 이틀을 선택했을 경우에는 8번을 입력했을 때 종료가 잘 되는지 테스트했는데 완벽하게 종료가 되었다.

```
7. 프로그램 종료
-----
7
프로그램을 종료합니다.
```

```
7. eat & splash
8. 프로그램 종료
*약을 먹어야(발라야) 하는 다음 날짜는 12월 24일입니다.*
-----
8
프로그램을 종료합니다.
```

## 4. 계획 대비 변경 사항

### 1) 기능8(사용자로부터 작업 요청받기) 추가

- 이전: X
- 이후: if문을 사용하여 사용자가 기능 1~7에 해당하는 번호를 선택하면 그 번호로 이동

할 수 있도록 만든다.

- 사유: 제안서 작성 때 작업 요청받을 수 있는 if문까지는 쓰지 않아서 새로 추가하였다.

## 2) 기능1(피부타입 진단)을 양자택일 합에 따라 분류하는 것으로 변경

- 이전: 양자택일의 개수를 비교하여 ~
- 이후: 양자택일(1 또는 2)의 합에 따라 ~, 배열에 담는다. for, if문 사용
- 사유: 개수를 비교하는 것보다는 다 더한 합에 따라서 나뉘는 게 더 직관적이고 나중에 유지보수하기에도 더 좋을 것이라 생각하였다. 그리고 배열에 담는다는 것과 for, if문 사용을 한다는 것까지 추가적으로 자세히 설명하였다.

## 3) 기능2(화장품 사용 이력 저장소)에 내용 추가

- 이전: 사용자로부터 별점과 메모를 입력 받는다.
- 이후: + 별점, 화장품이름, 메모는 구조체로 저장한다. (동적 메모리도 사용해서 낭비없이 잘 저장하기!) 파일 3개로 만들기(헤더파일 사용) 추가
- 사유: 어떤 배운 내용을 쓸 건지를 명확하게 추가적으로 표시하였다.

## 4) 기능3, 기능4에 파일 입출력으로 만들기 계획 추가

- 이전: 파일 입출력 관련 내용이 없었음.
- 이후: 기능3에 파일 입출력 사용해서 txt파일에 화장품들 이름과 대표성분을 미리 저장해둘 것이다. / 기능4 한줄 일기 쓰기, 보기에 파일 입출력 사용하기
- 사유: 기능3은 화장품 데이터를 어떻게 저장할지 안썼어서 추가했고, 기능4도 추가적으로 자세히 계획을 추가했다.

## 5) 기능5(화장품 구매 사이트 바로가기)를 링크 출력으로 수정

- 이전: 사이트가 바로 열리는 것으로 계획했음.
- 이후: 링크 출력(특정 상황 가정)으로 바꿈
- 사유: 사이트가 바로 열리는 것은 예기치 못한 상황이 벌어질 수 있어서 링크를 출력하고 사용자가 정확히 원하는 시기에 이동할 수 있게 했다.

## 6) 기능3(기초화장품 추천) 계획 수정 및 추가

- 이전: 기능2번과만 관련있었다. 또한 안 맞았던 화장품을 구조체에서 직접 선택하도록 했고 그 화장품의 대표 성분을 바탕으로 새로운 성분의 화장품을 추천했다.
- 이후: 기능1번과도 관련이 있다. 그리고 성분을 사용자로부터 직접 입력받는 것으로 수정하였다.
- 사유: 사용 이력 저장소에서 직접 선택하는 것보다는 성분을 직접 입력 받는 것이 직접적이기에 더 정확하고 좋을 것이라고 판단했다. 또한 입력한 대표성분이 정보가 별로 없어서 데이터베이스에 존재하지 않을 수 있기 때문에 추가적으로 지성중성건성과 같이 피부타입에 따라서 화장품 추천을 하는 것을 추가하였다.

## 7) 기능4(자신의 피부에 대해 기록하고 돌아보기)의 한줄일기 기능 수정

- 이전: 날짜수정, 일기 수정을 선택하여 수정할 수 있게 만들었다. (진행중)
- 이후: 쓰기, 보기로만 구성함.
- 사유: 일기 수정은 없어도 된다고 생각했다. 어차피 사용자가 형식에 제한받지 않고 있는 그대로 내 피부에 대해서 마구쓸 수 있는 공간으로 마련한 거기 때문이다.

## 8) 파일 3개 만들기 (헤더파일 추가하기)

- 이전: Hwa\_main.c 파일 한 개에 함수랑 모든 것이 담겨있었다.
- 이후: Hwa\_main.c, hwa\_project.c, hwa\_project.h 파일 3개를 만들어서 함수를 따로 관리하도록 만들었다.
- 사유: 함수를 더 쉽게 관리하기 위해 헤더파일을 만들어서 총 3개의 파일로 만들었다.

## 9) 기능2(화장품 사용 이력 저장소)에 삭제하기 기능 추가

- 이전: X
- 이후: 사용자가 입력했던 화장품의 이름, 별점, 메모를 삭제한다. 저장된 화장품이 없다면 없다는 문구를 출력한다. 삭제한 후에는 배열을 당겨오는 식으로 해서 배열(화장품 목록)

록)의 중간에 빈 칸이 없도록 만든다. 삭제한 화장품은 할당된 동적 메모리를 해제시킨다.

- 사유: 최대 30개까지밖에 저장을 못하게 제한했는데 그 이상 저장하려면 필요없는 화장품은 빼버려야하기 때문에 사용자 편의를 위해 삭제기능도 추가하였다.

## 10) 기능4(자신의 피부에 대해 기록하고 돌아보기)에 약 계산 기능 추가

- 이전: 오늘이 먹는 날이지만 알려준다고 했다.
- 이후: 다음 먹을 날을 계산해서 알려준다.
- 사유: 약 복용 주기가 이틀인 경우는 다음 먹을 날을 아는 것이 중요하기 때문에 계산해서 알려주는 것을 추가했다.

## 11) 중복 코드 함수화 시키기 (ErrorDetect, Freemalloc, MedCalculator 등등)

- 이전: 중복되는 부분이 있어도 전부 코드를 똑같이 복붙했다.
- 이후: 함수화 시켰다.
- 사유: 특히 여러 번 쓰이는 코드들을 변경되었을 때 관리하기 더 쉽게 만들었다.

## 5. 프로젝트 일정

업무		11/3	11/10	11/17	11/28	12/3	12/9	12/16	12/23
제안서 작성		완료							
기능1	세부기능1		완료						
기능2	세부기능1							완료	
	세부기능2			진행중					

기능3	세부기능1						완료		
기능4	세부기능1						완료		
	세부기능2					완료			
기능5				완료					
기능6			완료						
기능7		완료							
기능8		완료							

## 6. 느낀점

시험기간에는 프로젝트에 거의 손대지 못했지만... 그래도 끝나고 열심히 했다고 생각한다. 한학기동안 그래도 C언어 문법에 대해서는 다 알게 되어서 좋은 것 같고 어려웠지만 많은 걸 배웠던 것 같다. 사실 프로젝트도 처음이라 아직 미숙하고 기능들도 미약하지만 사용에는 전혀 문제없도록 프로그램을 구현했다고 생각한다. 방학에 다시 c언어 복습하면서 자바랑 그런것도 공부 해봐야겠다.