

C프로그래밍

# 청춘을 꽃과 함께

최종보고서

제출일자: 12월 24일

제출자명: 김세희

제출자 학번: 233973

## **1. 프로젝트 목표**

### **1) 배경 및 필요성**

살다 보면 누군가에게 꽃을 주는 상황이 생긴다. 소중한 사람에게 그 사람과 잘 어울리는 꽃과 바라는 바람을 담아 꽃을 준다면 좋지 않을까? 라는 생각부터 시작하여 인터넷에 있는 방대한 자료들을 모아 한번에 볼 수 있는 프로그램을 계획하였다.

### **2) 프로젝트 목표**

고객이 바라는 데이터를 수집하여 꽃을 추천해주고, 상황에 맞게 꽃을 추천해주며 더불어 꽃과 관련된 명언을 볼 수 있는 프로그램을 만드는 것이 목표이다.

### **3) 차별 점**

꽃에 마음을 담을 수 있다는 게 중요한 차이점이다. 하나의 꽃에는 여러 꽃말이 있고, 다른 꽃이라도 같은 꽃말을 갖고 있는 경우가 많다. 또한 내가 원하는 꽃말을 가진 꽃을 찾기도 쉽지 않은 일이다. 이런 문제점들을 해결하기 위해 다양한 기능들이 추가되어 다른 프로그램과 차이를 두었다.

## **2. 기능 계획**

### **1) 사용자의 탄생일에 맞는 탄생화를 알려주는 기능**

- 사용자의 태어난 생일을 입력 받고, 맞는 탄생화를 알려준다.

- (1) 세부 기능 1: 탄생화의 뜻을 추가로 알려준다.

### **2) 사용자가 원하는 테마를 골라 선물할 꽃을 추천해주는 기능**

- 다양한 테마 (우정, 사랑, 응원, 격려, 감사) 등을 골라 테마를 입력하면 그 테마에 맞는 꽃을 꽃말과 함께 추천해준다.

### 3) 꽃 명언 보기 기능

- 꽃 점을 통해 랜덤으로 좋은 글귀, 좋은 명언을 뽑을 수 있는 기능이다.

(1) 세부 기능 1: 랜덤으로 꽃과 관련된 명언들을 추천해준다.

### 4) 근처 지역 꽃집 알려주기 기능

- 사용자에게 지역을 입력 받고, 근처에 위치한 SNS나 인스타그램 등 유명한 감성 꽃집을 추천해준다.

## 3. 기능구현

### (1) 사용자의 탄생일에 맞는 탄생화를 알려주는 기능

- 입력: 사용자에게 생일을 입력 받음.

- 출력: [유효한 값인 경우]사용자의 생일의 맞는 탄생화를 출력함.

[유효하지 않는 값인 경우] "올바른 일 또는 월을 입력해주세요"를 출력함.

[잘못된 입력 형식인 경우] "올바른 형식으로 입력해주세요." 를 출력함.

- 설명: 사용자에게 생일을 입력 받은 후 올바른 입력인 경우 'findBirthFlower' 함수를 호출한다. 다양한 탄생화 리스트가 정의 되어있는 2차원 배열 중에서 사용자의 생일에 해당하는 달과 일을 뽑아 탄생화와 꽃말을 출력해준다.

- 적용된 배운 내용

① 2차원 배열: 탄생화와 뜻을 저장해 놓은 2차원 배열을 사용함.

② 포인터 사용: 탄생화를 저장해 놓은 2차원 배열에서 포인터를 사용하였음.

③ 반복 문: while 무한루프를 통해 사용자가 올바르게 생일을 입력할 수 있도록 함.

- ④ 조건문: 이중 조건문을 통해 사용자에게 받은 생일이 유효한 값인지 확인.
- ⑤ scanf\_s : scanf\_s 함수를 통해 사용자에게 생일을 입력 받음.
- ⑥ 함수: 복잡한 프로그램 형식을 findBirthFlower(month, day) 함수를 통해 간결화 시킴.
- ⑦ switch문: switch 문을 통해 달을 case화 시키고 사용자의 생일 달에 맞는 2차원 배열을 불러옴.
- ⑧ File: 사용자의 생일에 맞는 탄생화와 꽃말을 파일에 쓰고 난 후 저장함.
- ⑨ const: 변하지 않을 2차원 배열 리스트를 const로 상수화 시켜서 변하지 않는 값으로 고정시킴.
- ⑩ Do-while문: 사용자에게 무조건 생일을 입력하라는 프린트문을 한 번 출력시킴. 이후 무한루프를 통해 사용자가 올바른 입력을 할 때까지 반복
- ⑪ 구조체: 구조체를 통해 사용자에게 보여지는 출력 값에 사용자 이름을 출력하고, 함수를 호출할 때 사용자가 입력한 파일의 값을 전달함.
- ⑫ 모듈화: main파일과 user\_birth.c 파일을 구분하여 보기 간편하게 코딩함.

## - 코드 스크린샷

### ① 탄생화, 뜻을 저장해 놓은 2차원 배열 (\*긴 관계로 1월만 표현)

```
//탄생화 리스트 정의
const char* januaryFlowers[][2] = { {"스노드롭", "희망"},
{"노랑수선화", "사랑에 당하여"},
{"사프란", "후회 없는 결혼"},
{"하이아신스", "하이아신스"},
{"노루귀", "민내"},
{"흰제비꽃", "순진무구한 사랑"},
{"튤립", "실연"},
{"모란꽃", "사랑"},
{"노랑제비꽃", "수줍은 사랑"},
{"화양무", "참고 견여됨"},
{"속백나무", "건고한 우정"},
{"할기 알리섬", "배어난 미로"},
{"수선화", "신비"},
{"시클라멘", "내성적 성격"},
{"가시", "엄격"},
{"노랑 하이아신스", "승부"},
{"수련", "진단확정"},
{"아리까", "영속"},
{"소나무", "물류정수"},
{"미나리아재비", "진진난만"},
{"달창이영글", "우정"},
{"아까", "모성애"},
{"부들", "순종"},
{"사프란", "꽃도의 미"},
{"윌로우", "순진"},
{"미모사", "여미와 마음"},
{"아카시", "계모를 흐르는 마음"},
{"검은 보들라", "통기"},
{"아까", "모성애"},
{"매어 메리골드", "반드시 오고야 할 행복"},
{"노랑 사프란", "정조의 행복"} };

const char* februaryFlowers[][2] = {
const char* marchFlowers[][2] = {
const char* aprilFlowers[][2] = {
const char* mayFlowers[][2] = {
const char* juneFlowers[][2] = {
const char* julyFlowers[][2] = {
const char* augustFlowers[][2] = {
const char* septemberFlowers[][2] = {
const char* octoberFlowers[][2] = {
```

## ② 메인 파일 함수 호출 코드

```
if (choice == 1) {
    int month, day;
    do {
        printf("생일을 입력하세요 (월 일): ");
        scanf_s("%d월 %d일", &month, &day);

        if (month >= 1 && month <= 12) {
            if (day >= 1 && day <= 31) {
                findBirthFlower(month, day, user1.file_name, user1.name);
                break; // 올바른 입력일 경우 반복문 탈출
            }
            else {
                printf("올바른 일을 입력해주세요.\n");
            }
        }
        else {
            printf("올바른 월을 입력해주세요.\n");
        }
    } while (1); // 무한 루프를 돌면서 사용자가 올바른 입력을 할 때까지 계속 반복
}
```

## ③ findBirthFower 함수 및 switch문( Case문은 12월까지 있음)

```
int arrayLength;
const char* flower = NULL;
const char* meaning = NULL;

FILE* file;
fopen_s(&file, filename, "a");
switch (month) {
case 1:
    arrayLength = sizeof(januaryFlowers) / sizeof(januaryFlowers[0]);
    if (day >= 1 && day <= arrayLength) {
        flower = januaryFlowers[day - 1][0];
        meaning = januaryFlowers[day - 1][1];
        printf("%d월 %d일 탄생화는 %s이며 꽃말은 %s 입니다.!", month, day,
            flower, meaning);
        fputs("<생일의 탄생화>\n", file);
        fprintf(file, "%d월 %d일 탄생화는 %s이며 꽃말은 %s 입니다.!", month, day,
            flower, meaning);
        fclose(file);
    }
    break;
case 2:
    arrayLength = sizeof(februaryFlowers) / sizeof(februaryFlowers[0]);
    if (day >= 1 && day <= arrayLength) { ... }
    break;
case 3:
    arrayLength = sizeof(marchFlowers) / sizeof(marchFlowers[0]);
    if (day >= 1 && day <= arrayLength) { ... }
    break;
case 4:
    arrayLength = sizeof(aprilFlowers) / sizeof(aprilFlowers[0]);
    if (day >= 1 && day <= arrayLength) { ... }
    break;
case 5:
    arrayLength = sizeof(mayFlowers) / sizeof(mayFlowers[0]);
    if (day >= 1 && day <= arrayLength) { ... }
    break;
case 6:
    arrayLength = sizeof(juneFlowers) / sizeof(juneFlowers[0]);
    if (day >= 1 && day <= arrayLength) { ... }
    break;
case 7:
    arrayLength = sizeof(julyFlowers) / sizeof(julyFlowers[0]);
    if (day >= 1 && day <= arrayLength) { ... }
    break;
}
```

## ④ user\_birth.h 파일

```
void findBirthFlower(int month, int day, char filename, char user_name);
```

## (2) 사용자가 원하는 테마를 골라 선물할 꽃을 추천해주는 기능

- 입력: 사용자에게 [우정, 사랑, 응원, 감사, 격려] 중 원하는 테마를 입력 받음.
- 출력: 사용자가 원하는 테마에 맞게 알맞은 꽃과 꽃말을 출력함. 단 잘못입력했을 경우 다시 입력하라는 명령문 출력.
- 설명: 사용자에게 [우정, 사랑, 응원, 감사, 격려] 중 한 테마를 "user\_theme" 변수에 저장 후 조건문을 통해 유효한 값인 경우 "comeuser\_theme()" 함수를 불러와 배열에서 랜덤함수를 통해 무작위로 꽃과 꽃말을 추천해줌.
- 적용한 배운 내용
  - ① 2차원 배열: 테마 별 꽃 리스트를 총 5개 생성 및 꽃말과 꽃이 함께 담겨있는 2차원 배열을 완성하였음.
  - ② scanf\_s 함수: scanf\_s 함수를 통해 사용자가 원하는 테마 중 하나를 골라 입력 받을 수 있음
  - ③ if 조건문: 조건문을 통해 사용자의 값이 유효한 값인지 확인함.
  - ④ const: 변하지 않은 꽃과 꽃말이 담겨있는 리스트를 상수 값으로 고정.
  - ⑤ 모듈화: main 파일과 theme.c 파일을 분리하여 코드를 보기 좋게 제작.
  - ⑥ 함수: 'comeuser\_theme' 함수를 만듦으로써 의도가 분명하고 구별하기 좋게 만듦.
  - ⑦ Random 함수: import로 무작위 숫자를 생성 가능한 모듈을 가져와 랜덤으로 꽃과 꽃말을 추천해줄 수 있는 기능 사용.
  - ⑧ Do-while문: 처음에 무조건 사용자에게 테마 5가지를 출력한 뒤 사용자가 5가  
지 테마 중 올바른 테마를 입력할 때까지 무한 루프를 돌며 사용자의 입력을 받

음.

- 코드 스크린샷

### ① main 파일 함수 호출

```
else if (choice == 2) {
    int user_theme;
    do {
        printf("\n\n ----- \n\n");
        printf("1.우정\n2.사랑\n3.응원\n4.감사\n6.격려\n\n 총 5가지 테마중 선물할 꽃의 분위기에 맞는 테마를\n");
        scanf("%d", &user_theme);

        if (user_theme <= 5 && user_theme >= 1) {
            comeuser_theme(user_theme);
            break; // 올바른 입력일 경우 반복문 탈출
        }
        else {
            printf("잘못입력하셨습니다. 다시 입력해주세요: ");
        }
    } while (1); // 무한 루프를 돌면서 사용자가 올바른 입력을 할 때까지 계속 반복
}
```

### ② 테마 리스트 정의(theme.c파일)

```
const char* theme_friend[][2] = {
    {"라일락(보라색)", "친구의 사랑, 젊은 날의 추억"},
    {"촉백", "변함없는 우정"},
    {"장미(노란색)", "우정"},
    {"바이올렛", "영원한 우정"}
};

const char* theme_love[][2] = {
    {"리시안셔스", "변치않는 사랑"},
    {"튤립(빨간색)", "사랑을 고백하다"},
    {"소국(보라색)", "내 모든것을 그대에게"},
    {"장미(빨간색)", "열렬한 사랑"}
};

const char* theme_cheer[][2] = {
    {"프리지아(노란색)", "당신의 시작을 응원합니다"},
    {"장미(노란색)", "응원"},
    {"메리골드(노란색)", "반드시 오고야 말 행복"},
    {"솔리다고", "성공을 기원합니다"}
};

const char* theme_thank[][2] = {
    {"장미(진핑크)", "깊은 감사드립니다!"},
    {"캄파눌라", "감사합니다"},
    {"장미(핑크)", "당신을 존경합니다"},
    {"초롱꽃", "은혜를 보답하려는 마음"}
};

const char* theme_sad[][2] = {
    {"양귀비", "위로, 위안"},
    {"크렌베리", "마음의 고통을 위로하다"},
    {"용담", "위로가 되어주는"},
    {"안개꽃", "슬픔"}
};
```

### ③ theme.h 파일

```
#include <stdio.h>

void comeuser_theme(int user_theme);
```

#### ④ theme.c 파일

```
const char* theme_flower = NULL;
const char* theme_meaning = NULL;

//랜덤 숫자
int random_num = rand() % 4;

// 테마 별 확인하는 조건문
if (user_theme == 1) {
    theme_flower = theme_friend[random_num][0];
    theme_meaning = theme_friend[random_num][1];
    printf("스은(는) 어떠세요? 꽃말은 %s 이랍니다!", theme_flower, theme_meaning);
}
else if (user_theme == 2) {
    theme_flower = *theme_love[random_num][0];
    theme_meaning = *theme_love[random_num][1];
    printf("스은(는) 어떠세요? 꽃말은 %s 이랍니다!", theme_flower, theme_meaning);
}
else if (user_theme == 3) {
    theme_flower = *theme_cheer[random_num][0];
    theme_meaning = *theme_cheer[random_num][1];
    printf("스은(는) 어떠세요? 꽃말은 %s 이랍니다!", theme_flower, theme_meaning);
}
else if (user_theme == 4) {
    theme_flower = *theme_thank[random_num][0];
    theme_meaning = *theme_thank[random_num][1];
    printf("스은(는) 어떠세요? 꽃말은 %s 이랍니다!", theme_flower, theme_meaning);
}
else if (user_theme == 5) {
    theme_flower = *theme_sad[random_num][0];
    theme_meaning = *theme_sad[random_num][1];
    printf("스은(는) 어떠세요? 꽃말은 %s 이랍니다!", theme_flower, theme_meaning);
}

return 1;
```

### (3) 꽃 명언 보기 기능

- 입력: 사용자에게 명언을 볼 지 말지에 대한 의사를 입력 받음.
- 출력: [Yes인경우] 꽃 명언을 출력 [No인 경우] "괜찮아요! 그런 날도 있는 걸  
요!" 출력 [잘못 입력했을 경우] 다시 입력하는 문구 출력
- 설명: 사용자에게 명언을 볼 여부를 입력 받고 보는 경우(Y) LessonList() 함수를  
호출한다. (다른 경우는 위에 출력과 같이 똑같이 출력됨.) 함수 호출 후
- 적용한 배운 내용

① while 반복 문: while 무한 반복문을 통해 사용자에게 유효한 의사결정



을 받을 때까지 무한루프를 돌림.

② if 조건문: if문과 if\_else 조건문을 통해 사용자의 의사결정에 따라 프로그램 실행.

③ scanf\_s: scanf 함수를 통해 사용자의 의사결정을 입력 받음.

④ List: 꽃과 관련된 명언을 리스트화 시켜서 저장해 놓음.

⑤ 함수: "LessonList()" 함수를 통해 복잡한 프로그래밍을 보기 좋게 간결화 시킴.

⑥ const: 명언 리스트는 변화하지 않는 값이기 때문에 상수 값으로 고정

시

켜 놓음.

⑦ 포인터: 랜덤으로 뽑은 명언을 포인터로 주소로 가리켜서 사용자에게 출력할 때와 파일로 출력할 때 주소로 가리킴. 또한 함수를 호출할 때 주소로 호출

⑧ 구조체: 사용자의 이름과 사용자의 이름을 구조체를 통해 화면에 출력되도록 함.

⑨ Random 함수: import로 무작위 숫자를 생성 가능한 모듈을 가져와 랜덤으로 명언을 추천해 줄 수 있는 기능 사용.

⑩ File: 사용자에게 입력 받은 구조체 값을 통해 파일을 열고 꽃과 관련된 명언을 출력함.

## - 코드 스크린샷

### ① Do - while, 조건문을 통한 main 함수 내 함수들

```
else if (choice == 3) {
    char user_lesson;
    do {
        printf("\n\n-----\n\n");
        printf("오늘의 꽃 명언을 보시겠어요? (중요면 'Y', 싫으면 'N'): ");
        scanf_s(" %c", &user_lesson, 1);

        if (user_lesson == 'Y') {
            LessonList(user1.name, user1.file_name);
            break; // 올바른 입력일 경우 반복문 탈출
        }
        else if (user_lesson == 'N') {
            printf("괜찮아요 ! 그런 날도 있는 걸요!");
            break; // 올바른 입력일 경우 반복문 탈출
        }
        else {
            printf("잘못입력하셨습니다. 다시 입력해주세요.\n");
        }
    } while (1); // 무한 루프를 돌면서 사용자가 올바른 입력을 할 때까지 계속 반복
}
```

### ② 꽃 관련 명언 리스트

```
//꽃 관련 명언 리스트
const char *Lessons[] = {
    "꽃을 손에 들고 정말 바라보면 잠시나마 너의 세상이다. - 조지아 오키프",
    "꽃을 보고자 하는 사람에겐 어디에나 꽃이 피어있다. - 앙리 마티스",
    "인생은 꽃과 같고 사랑은 그 속의 꿀과 같다. -빅토르 휴고"
    "틀림은 누구에게도 깊은 인상을 주려고 애쓰지 않습니다. 장미와 달라지기 위해 애쓰지 않습니다.",
    "꽃은 꽃이 되라고 하지 않았고 나도 내가 되라고 하지 않았다. -커트 보네거트",
    "모든 꽃을 자를 수는 있지만 봄이 오는 것을 막을 순 없습니다. - 파블로 네루다"};
```

### ③ LessonList 함수

```
void LessonList(const char* user_name, const char* filename) {
    const char* Lessons[] = { ... }

    // 랜덤 수를 뽑는 함수 아직 안배움 --> 미리 4로 배정
    srand((unsigned int)time(NULL));
    int randomnum = rand() % 6;
    const char* Lesson = Lessons[randomnum];
    printf("\n%s\n", Lesson);

    FILE* file;

    if (fopen_s(&file, filename, "a") == 0) {
        int result = fprintf(file, "\n<%s님의 꽃과 관련된 명언>\n- %s\n", user_name, Lesson);

        if (result < 0) {
            printf("파일에 쓰는 중 오류가 발생했습니다.\n");
            fclose(file);
            return;
        }
    }
}
```

#### ④ Lesson.h 파일

```
void LessonList(const char* filename, const char* user_name);
```

#### (4) 근처 지역 꽃집 알려주기 기능

- 입력 값: 사용자에게 사는 지역을 입력 받음 (동구, 서구, 광산구, 남구, 북구)
- 출력 값: 사용자의 지역에 맞는 가까운 꽃집 3개를 추천하고 파일에 저장함. 파일을 열 수 없는 경우 "파일을 열 수 없습니다."를 출력함.
- 설명: 사용자에게 지역(광주)을 입력 받는다. 그리고 'getNearbyFlowerShops' 함수를 통해 동적메모리를 할당하고 꽃집 데이터를 복사한 뒤 사용자에게 가까운 꽃집 리스트를 출력한다 또한 구조체 포인터로 받은 사용자 파일 이름을 통해 가까운 꽃집 리스트를 파일로 출력함.
- 적용한 배운 내용
  - ① 함수: 'provideFlowerShop', 'freeFlowershops', 'getNearbyFlowerShops' 함수를 통해 코드를 보기 좋게 간결화 시킴.
  - ② 구조체 및 구조체 배열: 구조체를 통해 사용자의 이름과 사용자 파일의 정보를 활용하고 Flowershop 을 통해 구조체 배열을 생성하여 꽃집이름과 지역을 정리함
  - ③ 상수: define 함수를 통해 변하지 않는 값(MAX\_REGION\_LENGTH 등)을 정리하여 오류가 안 나도록 값을 고정시킴.
  - ④ 동적메모리: 가까운 꽃집 정보를 저장하기 위해 동적 메모리 할당을 활용하고, 각 꽃집의 이름과 위치를 \_strdup 함수를 사용하여 복제. 사용자가

입력한 지역에 따라 가까운 꽃집을 찾아 화면에 출력하고, 해당 정보를 파일에 추가할 때 동적 메모리를 활용함.

⑤ for 반복 문: 반복문을 통해 사용자에게 가까운 지역에 꽃집 리스트를 출력함.

⑥ 모듈화: main 파일과 'flower\_shop.c' 파일을 분리함으로써 코드를 더욱 보기 좋고 해석하기 쉽게 정리함.

⑦ FILE: 파일을 통해 사용자는 지역에 가까운 꽃집 리스트를 파일로 출력

⑧ scanf\_s: scanf\_s 함수를 통해 사용자가 사는 지역을 입력 받음.

⑨ if문: 조건문을 통해 파일이 정상적으로 열렸을 경우 프로그램을 정상적으로 실행하고 아닌 경우 else함수를 통해 파일이 열리지 않았음을 출력

- 코드 스크린샷

① main 파일 함수 호출

```
else if (choice == 4) {  
    provideFlowerShop(user1.name, user1.file_name);  
    getchar();  
}
```

② flower\_shop.h 파일

```
#include <stdio.h>  
#include <stdlib.h>  
  
void provideFlowerShop(const char* user_name, const char* filename);
```

③ flower\_shop.c 파일 상수 정의 및 구조체 정의

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

#define MAX_REGION_LENGTH 20
#define MAX_FLOWER_SHOPS_PER_REGION 3
#define MAX_REGIONS 5
#define MAX_FLOWER_SHOPS (MAX_FLOWER_SHOPS_PER_REGION * MAX_REGIONS)

typedef struct {
    char* name;
    char* location;
} FlowerShop;
```

#### ④ flowerShop 구조체 배열

```
FlowerShop flowerShopsData[] = {
    {"아뜰리에온화 [장동로 1 장동로 1번길 1층]", "동구"},
    {"플로렌 [동명로 26번길 15-1 1층]", "동구"},
    {"꽃디 [동명로 35 1층]", "동구"},
    {"메이비플라워 [상일로 24번길 17 1층]", "서구"},
    {"Bloomsbury [상무화원로 3번길 6]", "서구"},
    {"오월꽃집 [시청로 27 1층 101호]", "서구"},
    {"민플라워 [임방울대로 330 애플타워 105호]", "광산구"},
    {"도이브 [수완로 9번길 18]", "광산구"},
    {"라레브 [장신로 20번안길 2]", "광산구"},
    {"들로 [봉선1로 112 들로]", "남구"},
    {"플라워오브유 [백서로 71 1층]", "남구"},
    {"딤가든[희재로 1290 - 1 2층]", "남구"},
    {"르로제 [설죽로 258 1층]", "북구"},
    {"꽃이있는집 [첨단연신로 133번길 4호반 2단지]", "북구"},
    {"꽃피어나 [우치로 88 - 1 1층]", "북구"}
};
```

#### ⑤ flower\_shop.c 파일

```

//동적메모리
void getNearbyFlowerShops(const char* region, FlowerShop nearbyShops[MAX_FLOWER_SHOPS_PER_REGION]) {
    int count = 0;
    for (int i = 0; i < MAX_FLOWER_SHOPS; ++i) {
        if (strcmp(region, flowerShopsData[i].location) == 0) {
            // 동적으로 메모리 할당하고 복사
            nearbyShops[count].name = _strdup(flowerShopsData[i].name);
            nearbyShops[count].location = _strdup(flowerShopsData[i].location);
            count++;
            if (count == MAX_FLOWER_SHOPS_PER_REGION) {
                break;
            }
        }
    }
}

void freeFlowerShops(FlowerShop nearbyShops[MAX_FLOWER_SHOPS_PER_REGION]) {
    for (int i = 0; i < MAX_FLOWER_SHOPS_PER_REGION; ++i) {
        free(nearbyShops[i].name);
        free(nearbyShops[i].location);
    }
}

void provideFlowerShop(const char* user_name, const char* filename) {
    FILE* file;
    if (fopen_s(&file, filename, "a") == 0) {
        char userRegion[MAX_REGION_LENGTH];
        FlowerShop nearbyShops[MAX_FLOWER_SHOPS_PER_REGION];

        printf("시는 지역을 입력하세요 (동구, 서구, 광산구, 남구, 북구): ");
        scanf_s("%s", userRegion, (unsigned)_countof(userRegion));

        getNearbyFlowerShops(userRegion, nearbyShops);

        printf("가까운 꽃집 추천 리스트:\n");
        fprintf(file, "%s<님의 %s 가까운 꽃집 List>\n", user_name, userRegion);
    }
}

```

```

printf("가까운 꽃집 추천 리스트:\n");
fprintf(file, "%s<님의 %s 가까운 꽃집 List>\n", user_name, userRegion);

for (int i = 0; i < MAX_FLOWER_SHOPS_PER_REGION; ++i) {
    printf("%d. %s (%s)\n", i + 1, nearbyShops[i].name, nearbyShops[i].location);
    fprintf(file, "%d. %s (%s)\n", i + 1, nearbyShops[i].name, nearbyShops[i].location);
}

fclose(file);
// 메모리 해제를 provideFlowerShop 함수가 종료되기 전에 수행
freeFlowerShops(nearbyShops);
}
else {
    printf("파일을 열 수 없습니다.\n");
    return 0;
}
}

```

## (5) 기타 인사말 파일 및 기능

- 입력 값: 사용자에게 이름과 파일이름을 입력 받음
- 출력 값: 프로그램 내에서 사용자의 이름을 출력하고, 파일 이름을 통해 파일 출력
- 설명: 사용자에게 구조체를 통해 깔끔하게 이름과 파일이름을 입력 받기 및 인사말을 출력한다.
- 적용한 배운 내용

- ① scanf\_s: 사용자에게 이름과 파일이름을 입력 받음
- ② 구조체: 구조체를 통해 사용자의 이름과 파일이름을 정리함
- ③ 모듈화: 모듈화를 통해 main파일과 분리함으로써 보기 좋게 간결화시킴.
- ④ 함수: greeting 함수를 분리함으로써 코드를 보기 좋게 간결화 시킴.

- 코드 스크린샷

- ① user\_Info.c 파일

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include "userInfo.h"

struct userInfo greeting(void) {
    struct userInfo user1;

    //인삿말
    printf("-----\n");
    printf("'청춘을 꽃과 함께' 프로그램에 오신것을 환영합니다.\n");
    printf("프로그램의 결과는 사용자님의 파일에 uesr.txt파일로 저장됩니다.\n");
    printf("-----\n\n");

    printf("사용자의 이름을 입력해주세요: ");
    scanf_s("%s", user1.name, (int)sizeof(user1.name));

    printf("%s님의 파일 이름을 입력해주세요: ", user1.name);
    scanf_s("%s", user1.file_name, (int)sizeof(user1.file_name));

    printf("\n%s파일로 결과를 출력해드릴게요!\n", user1.file_name);

    printf("-----\n");
    printf("★청춘을 꽃처럼★");

    return user1;
}
```

## ② user\_Info.h 파일

```
#include <stdio.h>
#include <stdlib.h>
#include <time.h>

struct userInfo {
    char name[10]; //이름
    char file_name[30]; //파일
};

struct userInfo greeting(void);
```

## ③ main 파일 함수 호출

```
int main() {
    struct userInfo user1 = greeting();
}
```

## ④ 결과 스크린샷

```
-----
'청춘을 꽃과 함께' 프로그램에 오신것을 환영합니다.
프로그램의 결과는 사용자님의 파일에 uesr.txt파일로 저장됩니다.
-----

사용자의 이름을 입력해주세요: 김세희
김세희님의 파일 이름을 입력해주세요: user.txt

user.txt파일로 결과를 출력해드릴게요!
-----
★청춘을 꽃처럼★
-----
```

# 4. 테스트 결과



## (1) 사용자의 탄생일에 맞는 탄생화를 알려주는 기능

- 설명: 사용자에게 생일을 입력 받고 그에 맞는 탄생화와 꽃말을 출력한다.

### ① 생일을 옳게 입력했을 경우

```
-----  
원하는 기능을 선택해주세요: 1  
생일을 입력하세요 (월 일): 10월 26일  
10월 26일 탄생화는 수영이며 꽃말은 애정 입니다.!  
-----
```

### ② 월을 옳바르게 입력하지 않은 경우

```
생일을 입력하세요 (월 일): 13월 30일  
올바른 월을 입력해주세요.  
생일을 입력하세요 (월 일):
```

### ③ 일을 옳바르게 입력하지 않은 경우

```
생일을 입력하세요 (월 일): 12월 42일  
올바른 일을 입력해주세요.
```

### ④ user.txt 파일

```
파일  편집  보기  
  
<김세희님의 탄생화>  
10월 26일 탄생화는 수영이며 꽃말은 애정 입니다.!
```

## (2) 사용자가 원하는 테마를 골라 선물할 꽃을 추천해주는 기능

- 설명: 사용자에게 원하는 상황에 맞는 테마를 골라 해당 테마에 맞는 꽃을 랜덤으로 출력해주는 기능

① 우정을 고를 경우

```
1.우정
2.사랑
3.응원
4.감사
5.격려
총 5가지 테마중 선물할 꽃의 분위기에 맞는 테마를 골라 주세요!: 1
측백은(는) 어떠세요? 꽃말은 변함없는 우정 이랍니다!
-----
```

② 사랑을 고를 경우

```
1.우정
2.사랑
3.응원
4.감사
5.격려
총 5가지 테마중 선물할 꽃의 분위기에 맞는 테마를 골라 주세요!: 2
장미(빨간색)은(는) 어떠세요? 꽃말은 열렬한 사랑 이랍니다!
-----
```

③ 응원을 고를 경우

```
1.우정
2.사랑
3.응원
4.감사
5.격려
총 5가지 테마중 선물할 꽃의 분위기에 맞는 테마를 골라 주세요!: 3
메리골드(노란색)은(는) 어떠세요? 꽃말은 반드시 오고야 말 행복 이랍니다!
-----
```

④ 감사를 고를 경우

```
1.우정
2.사랑
3.응원
4.감사
5.격려
총 5가지 테마중 선물할 꽃의 분위기에 맞는 테마를 골라 주세요!: 4
장미(진핑크)은(는) 어떠세요? 꽃말은 깊은 감사드립니다! 이랍니다!
-----
```

⑤ 격려를 고를 경우

1. 우정
2. 사랑
3. 응원
4. 감사
5. 격려

총 5가지 테마중 선물할 꽃의 분위기에 맞는 테마를 골라 주세요!: 5  
크랜베리은(는) 어떠세요? 꽃말은 마음의 고통을 위로하다 이랍니다!

#### ⑥ 잘못 입력했을 경우

1. 우정
2. 사랑
3. 응원
4. 감사
5. 격려

총 5가지 테마중 선물할 꽃의 분위기에 맞는 테마를 골라 주세요!: 6  
잘못 입력하셨습니다. 다시 입력해주세요:

#### ⑦ user.txt 파일

<김세희님의 테마: 우정>

추천 꽃: 측백

측백의 꽃말: 변함없는 우정

<김세희님의 테마: 우정>

추천 꽃: 장미(빨간색)

장미(빨간색)의 꽃말: 열렬한 사랑

<김세희님의 테마: 우정>

추천 꽃: 메리골드(노란색)

메리골드(노란색)의 꽃말: 반드시 오고야 말 행복

<김세희님의 테마: 우정>

추천 꽃: 장미(진핑크)

장미(진핑크)의 꽃말: 깊은 감사드립니다!

<김세희님의 테마: 우정>

추천 꽃: 크랜베리

크랜베리의 꽃말: 마음의 고통을 위로하다

### (3) 꽃 명언 보기 기능

- 설명: 사용자에게 여부에 따라 꽃과 관련된 명언을 추천해주는 기능

#### ① 'Y'인 경우

```
-----  
오늘의 꽃 명언을 보시겠어요? (좋으면 "Y", 싫으면 "N"): Y  
꽃을 보고자 하는 사람에겐 어디에나 꽃이 피어있다. - 앙리 마티스  
-----
```

#### ② 'N'인 경우

```
-----  
오늘의 꽃 명언을 보시겠어요? (좋으면 "Y", 싫으면 "N"): N  
괜찮아요 ! 그런 날도 있는 걸요!  
-----
```

#### ③ 잘못 입력했을 경우

```
-----  
오늘의 꽃 명언을 보시겠어요? (좋으면 "Y", 싫으면 "N"): 0  
잘못입력하셨습니다. 다시 입력해주세요.  
  
-----  
오늘의 꽃 명언을 보시겠어요? (좋으면 "Y", 싫으면 "N"):
```

#### ④ user.txt 파일

```
<김세희님의 꽃과 관련된 명언>  
- 인생은 꽃과 같고 사랑은 그 속의 꿀과 같다. - 빅토르 휴고
```

### (4) 근처 지역 꽃집 알려주기 기능

- 설명: 사용자가 사는 지역을 입력 받고 그 지역과 가까운 꽃집을 3개 추천해주는 기능

#### ① 서구를 입력한 경우

```
원하는 기능을 선택해주세요: 4
사는 지역을 입력하세요 (동구, 서구, 광산구, 남구, 북구): 서구
가까운 꽃집 추천 리스트:
1. 메이비플라워 [ 상일로 24번길 17 1층 ] (서구)
2. Bloomsbury [ 상무화원로 3번길 6 ] (서구)
3. 오월꽃집 [ 시청로 27 1층 101호 ] (서구)
```

#### ② 광산구를 입력한 경우

```
-----
원하는 기능을 선택해주세요: 4
사는 지역을 입력하세요 (동구, 서구, 광산구, 남구, 북구): 광산구
가까운 꽃집 추천 리스트:
1. 민플라워 [ 임방울대로 330 애플타워 105호 ] (광산구)
2. 도이브 [ 수완로 9번길 18 ] (광산구)
3. 라레브 [ 장신로 20번안길 2 ] (광산구)
-----
```

#### ③ 동구를 입력한 경우

```
원하는 기능을 선택해주세요: 4
사는 지역을 입력하세요 (동구, 서구, 광산구, 남구, 북구): 동구
가까운 꽃집 추천 리스트:
1. 아뜰리에온화 [장동로 1 장동로 1번길 1층] (동구)
2. 플로렌 [ 동명로 26번길 15-1 1층] (동구)
3. 꽃디 [동명로 35 1층] (동구)
```

#### ④ 남구를 입력한 경우

```
원하는 기능을 선택해주세요: 4
사는 지역을 입력하세요 (동구, 서구, 광산구, 남구, 북구): 남구
가까운 꽃집 추천 리스트:
1. 들로 [ 봉선1로 112 들로 ] (남구)
2. 플라워오브유 [ 백서로 71 1층 ] (남구)
3. 딥가든[회재로 1290 - 1 2층] (남구)
```

#### ⑤ 북구를 입력한 경우

```
원하는 기능을 선택해주세요: 4
사는 지역을 입력하세요 (동구, 서구, 광산구, 남구, 북구): 북구
가까운 꽃집 추천 리스트:
1. 르로제 [ 설죽로 258 1층 ] (북구)
2. 꽃이있는집 [ 첨단연신로 133번길 4호반 2단지 ] (북구)
3. 꽃피어나 [ 우치로 88 - 1 1층 ] (북구)
```

## ⑥ 잘못 입력한 경우

```
원하는 기능을 선택해주세요: 4
사는 지역을 입력하세요 (동구, 서구, 광산구, 남구, 북구): 구구
올바른 지역을 입력해주세요.
사는 지역을 입력하세요 (동구, 서구, 광산구, 남구, 북구):
```

## ⑦ user.txt 파일

```
<김세희님의 동구 가까운 꽃집 List>
1. 아뜰리에온화 [ 장동로 1 장동로 1번길 1층 ] (동구)
2. 플로렌 [ 동명로 26번길 15-1 1층 ] (동구)
3. 꽃디 [ 동명로 35 1층 ] (동구)

<김세희님의 서구 가까운 꽃집 List>
1. 메이비플라워 [ 상일로 24번길 17 1층 ] (서구)
2. Bloomsbury [ 상무화원로 3번길 6 ] (서구)
3. 오월꽃집 [ 시청로 27 1층 101호 ] (서구)

<김세희님의 광산구 가까운 꽃집 List>
1. 민플라워 [ 임방울대로 330 애플타워 105호 ] (광산구)
2. 도이브 [ 수완로 9번길 18 ] (광산구)
3. 라레브 [ 장신로 20번안길 2 ] (광산구)

<김세희님의 남구 가까운 꽃집 List>
1. 들로 [ 봉선1로 112 들로 ] (남구)
2. 플라워오브유 [ 백서로 71 1층 ] (남구)
3. 답가든[ 회재로 1290 - 1 2층 ] (남구)

<김세희님의 북구 가까운 꽃집 List>
1. 르로제 [ 설죽로 258 1층 ] (북구)
2. 꽃이있는집 [ 첨단연신로 133번길 4호반 2단지 ] (북구)
3. 꽃피어나 [ 우치로 88 - 1 1층 ] (북구)
```

## 5. 제안서 계획 대비 변경 사항

## 1) 사용자가 태어난 날에 탄생화를 알려주는 기능

- 이전: 교훈과 탄생화의 배경을 같이 알려줌
- 이후: 탄생화와 꽃말을 알려줌.
- 사유: 교훈과 탄생화의 꽃말이 비슷한 점이 있어 교훈은 이미 비슷한 명언을 알려주는 기능이 있기에 꽃말로 변경, 배경은 각 꽃마다 추상적이고 확실치 않는 정보이기에 제외.

## 2) 사용자가 원하는 테마를 골라 선물할 꽃을 추천해주는 기능

- 이전: 사용자가 선물해 줄 사람의 달까지도 고려하여 테마 꽃을 추천
- 이후: 테마 5개중 선택하여 달과 상관없이 꽃을 추천
- 사유: 사용자의 달마다 꽃을 추천해줄 경우 전체적으로 예쁘거나 의미 있는 꽃을 추천을 해주지 못하는 상황이 발생함. 그리하여 달과 상관없이 상황에 상황에 맞는 꽃을 추천해주는 것으로 변경.

## 3) 꽃 명언 보기 기능

- 이전: 명언과 더불어 꽃 점을 통해 랜덤으로 좋은 글귀, 좋은 명언을 뽑을 수 있는 기능
- 이후: 꽃 점을 제외하고 명언을 랜덤으로 출력해주는 기능으로 변경
- 사유: 꽃 점을 통해 사용자의 기분과 상황 등을 고려할 수 없음에. 위험요소를 발견 그리하여 꽃 점을 없애고 명언을 볼 수 있는 기능으로 변경.

## 4) 기능 추가

- 이전: 기능 3개
- 이후: '근처 꽃집을 알려주는 기능', '파일로 결과를 정리하여 출력해주는 기능' 추가
- 사유: 꽃은 바로바로 필요한 경우가 더 많은 것으로 판별 그리하여 온라인 사이트가 아닌 오프라인 매장으로 알려주는 것으로 변경 및 추가함. 또한 사용자의 편리함을 위해 결과를 한번에 볼 수 있도록 파일로 결과를 출력해주는 기능을 추가함.

## 6. 느낀 점

파이썬에 비해 오히려 C언어가 쉽다 라는 생각이 들었다. 하지만 파이썬 보다는 C언어가 개인적으로 프로그램을 하기 편했던 것 같다. 수업에서 아쉬운 점은 과제에 비해 보고서와 시험기간에 준비할 것들이 많아 힘들었지만 하지만 그만큼 배워가는 것도 많았다.

개인적으로는 다른 프로그래밍 수업보다 유익한 시간이었던 것 같다. 시험기간에 준비하는 기말고사 대체 과제라 시간과 여유가 비교적 부족하여 프로그램을 더욱 완성도 있게 마무리하지 못해 아쉬웠지만 방학에 한 번 더 고퀄리티에 기능을 완성해 보고싶다 라는 생각이 들 정도로 내가 만든 프로그램에 애정이 생긴 것 같다.

수업 후반으로 갈수록 내용이 어려워져서 사실 포기하고 싶다 라는 생각도 들었다. 하지만 교수님의 빠른 피드백과 질문에 답변 그리고 중간고사 때 교수님이 우리를 위해 PPT 정리와 수업을 아픈신 데도 불구하고 열정적으로 하시는 걸 보면서

포기하지 않고 끝까지 프로그램을 완성시킬 수 있었다.

교수님께 한마디: 교수님 덕분에 타과임에 불안하고, 따라가기 힘든 적도 있었지만 끝까지 마무리할 수 있었던 것 같습니다. 개인적으로 굉장히 유익했던 강의고 처음이신 데도 불구하고 재밌게 수업 들을 수 있었습니다 github를 사용한 수업이었던 점은 굉장히 좋았습니다! 한학기 동안 감사했습니다.