

C프로그래밍및실습

# 기온에 따른 옷차림(룩) 추천 프로그램

진척 보고서 #2

제출일자: 2023년 12월 9일

제출자명: 김형민

제출자학번: 233332

## 1. 프로젝트 목표

### 1) 배경 및 필요성

우리는 기온변화와 옷차림 선택에 관한 문제에 직면하고 있다. 요즈음 지구 온난화가 가속화하면서, 이상기온으로 인해 옷차림(룩)을 정하기 쉽지 않아졌다. 기온변화와 개인의 의류 선택 사이에는 중요한 연결이 있으며, 잘못된 의류 선택은 고객의 편안함과 만족도에 영향을 미칠 수 있다. 따라서 이 문제를 해결하기 위해 기온과 개인이 가진 옷을 고려하여 옷을 추천해주는 프로그램이 필요하다.

### 2) 프로젝트 목표

오늘의 기온 정보와 고객이 가진 옷들을 고려하여 옷차림(룩)을 추천해주고 부족한 계절의 옷차림(룩)의 개수를 알려주어 추천하는 프로그램을 개발

### 3) 차별점

이 프로그램의 독특한 점 중 하나는 사용자가 현재 가진 옷의 수량을 기반으로 어떤 계절의 옷을 더 구매해야 하는지 추천해준다는 점이다. 각 계절마다 최대 5가지의 옷을 입력할 수 있고, 부족한 옷을 사용자에게 알려줌으로써 옷을 적절히 보충할 수 있도록 돕는다. 이렇게하여 사용자는 다가올 계절에 맞춰 옷을 구매하여 옷장을 효율적으로 관리할 수 있게 됩니다.

## 2. 기능 계획

### 1) 기능 1 사용자 옷차림(룩) 입력 기능

- (1) 세부 기능 1 (각 계절의 옷차림(룩)을 입력)
- 최대 5개의 옷차림(룩)을 입력받을 수 있게 한다.

#### (1) 세부 기능 2 (옷차림(룩)의 리스트)

- 사용자가 입력한 옷차림(룩)을 계절별로 나누어 리스트로 보여준다.

### 2) 기능 2 기온에 따른 추천 기능

- (1) 세부 기능 1 (기온을 입력받고 계절을 판별)
- 사용자로부터 기온을 입력받고 계절을 판별한다.
- (1) 세부 기능 2 (판별한 계절에 따라 옷차림(룩) 추천)
- 판별한 계절로부터 사용자가 입력한 옷차림(룩) 중에서 랜덤으로 추천한다.

### 3) 기능 3 부족한 옷 알림 기능

- (1) 세부 기능 1 (부족한 계절별 옷차림(룩)을 알림)
- 사용자가 입력한 옷차림(룩) 중에서 어떤 계절의 옷을 더 사야 할지 사용자에게 알려준다.

## 3. 진척사항

### 1) 기능 구현

#### (1) 구현한 기능 이름 각 계절의 옷차림(룩)을 입력하고 저장하는 기능의 헤더파일

- 입력:

clothes: 옷차림(룩)을 저장하는 배열

season: 현재 입력되는 계절을 가리키는 문자열

numClothes: 해당 계절의 옷 개수를 나타내는 변수

- 설명:

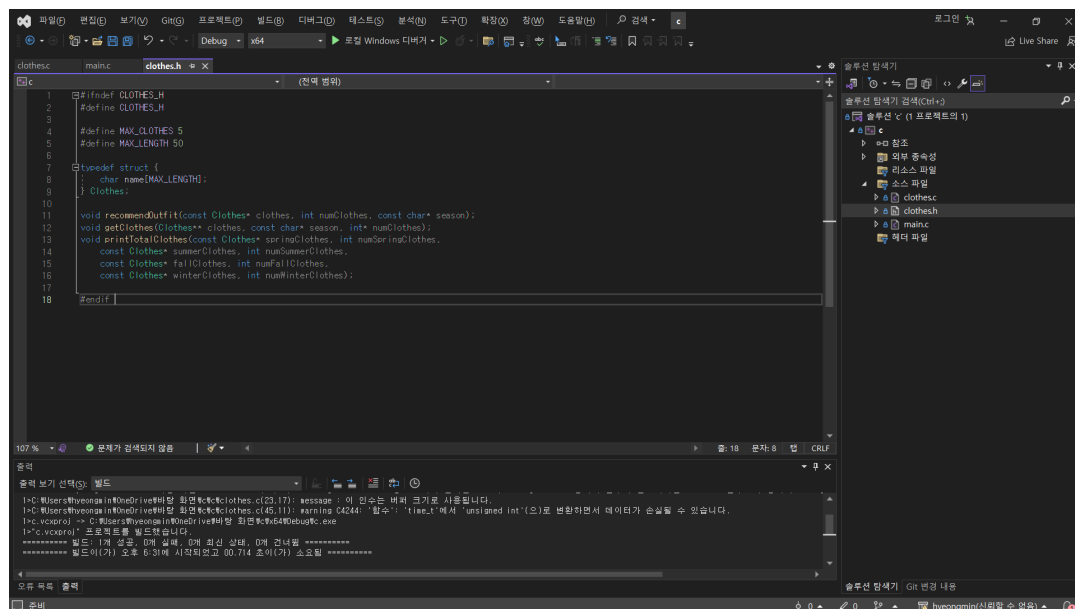
사용자로부터 특정 계절의 옷 정보를 입력받음

사용자에게 특정 형식(띄어쓰기 없이)에 따라 옷차림(룩)을 입력받는 방법을 안내  
입력된 정보를 배열에 저장하여 관리  
선택된 계절에 따라 해당 옷차림을 추천하는 기능을 제공

- 적용된 배운 내용:

배열, 조건문, 포인터, 구조체, 헤더파일

## - 코드 스크린샷



```
1 #ifndef CLOTHES_H
2 #define CLOTHES_H
3
4 #define MAX_CLOTHES 5
5 #define MAX_LENGTH 50
6
7 typedef struct {
8     char name[MAX_LENGTH];
9 } Clothes;
10
11 void recommendOutfit(const Clothes* clothes, int numClothes, const char* season);
12 void getClothes(Clothes* clothes, const char* season, int* numClothes);
13 void printTotalClothes(const Clothes* springClothes, int numSpringClothes,
14     const Clothes* summerClothes, int numSummerClothes,
15     const Clothes* fallClothes, int numFallClothes,
16     const Clothes* winterClothes, int numWinterClothes);
17
18 #endif
```

## (2) 구현한 기능 이름 각 계절의 옷차림(룩)을 입력하고 저장하는 기능의 메인파일

- 메인 함수(main()):

springClothes, summerClothes, fallClothes, winterClothes와 같은 포인터 변수를 선언하여 각 계절의 옷 정보를 저장할 준비

numSpringClothes, numSummerClothes, numFallClothes, numWinterClothes와 같은 변수를 선언하여 각 계절별 옷의 개수를 추적

- 옷 정보 입력 및 저장:

getClothes()함수를 호출하여 각 계절에 해당하는 옷 정보를 입력. 해당 계절에 따른 옷 정보를 사용자로부터 입력받고, 입력된 정보를 Clothes구조체 배열에 저장

- 옷 정보 출력:

printTotalClothes()함수를 호출하여 각 계절별로 입력된 옷 정보를 출력. 입력된 각 계절의 옷 정보를 화면에 출력

### - 메모리 해제:

free()함수를 사용하여 동적으로 할당된 메모리를 해제. 각 계절별 옷 정보를 저장하기 위해 동적으로 메모리를 할당하고 있기 때문에 프로그램이 종료될 때 이들을 해제해야함

### - 적용된 배운 내용:

포인터, 동적메모리, 구조체, 메인파일

### - 코드 스크린샷

```
1 #include "clothes.h"
2 #include <stdio.h>
3 #include <stdlib.h>
4
5 int main() {
6     Clothes* springClothes = &summerClothes, *fallClothes, *winterClothes;
7     int numSpringClothes = 0, numSummerClothes = 0, numFallClothes = 0, numWinterClothes = 0;
8
9     getClothes(&springClothes, "봄", &numSpringClothes);
10    getClothes(&summerClothes, "여름", &numSummerClothes);
11    getClothes(&fallClothes, "가을", &numFallClothes);
12    getClothes(&winterClothes, "겨울", &numWinterClothes);
13
14    printf("Clothes: %p, %p, %p, %p\n", springClothes, summerClothes, fallClothes, winterClothes);
15    printf("numClothes: %d, %d, %d, %d\n", numSpringClothes, numSummerClothes, numFallClothes, numWinterClothes);
16
17    free(springClothes);
18    free(summerClothes);
19    free(fallClothes);
20    free(winterClothes);
21
22    return 0;
23 }
```

## 3) 구현한 기능 이름 사용자로부터 각 계절의 옷차림 정보를 입력받아 저장하는 함수

### - 입력:

clothes: 옷차림(룩)을 저장하는 배열을 가리키는 포인터

season: 현재 입력되는 계절을 가리키는 문자열

numClothes: 해당 계절의 옷 개수를 나타내는 변수를 가리키는 포인터

### - 설명:

사용자에게 특정 계절의 옷 정보를 입력하라는 안내 메시지를 출력

해당 계절의 옷 정보를 저장하기 위해 동적 메모리를 할당합니다. 이때, 최대 옷 개수(MAX\_CLOTHES)에 따라 메모리를 할당

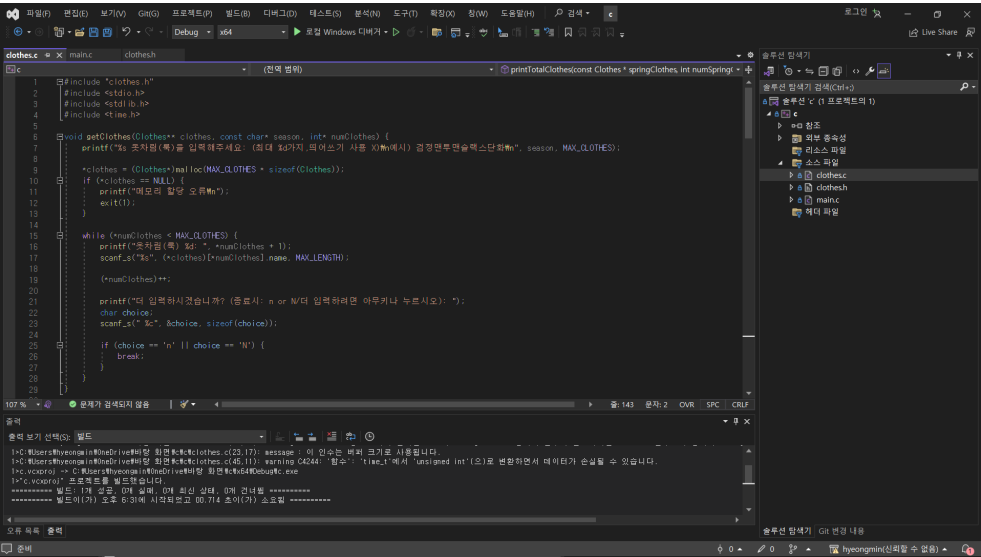
사용자에게 입력을 받고, 입력된 정보를 배열에 저장합니다. 이때, 사용자는 예시 형식에 맞춰 옷차림 정보를 입력

사용자가 더 이상 입력을 원하지 않을 때까지 계속해서 옷차림 정보를 입력받고 저장

### - 적용된 배운 내용:

동적 메모리 할당, 포인터, 입출력, 함수, 반복문, 조건문

- 코드 스크린샷



4) 구현한 기능 이름 특정 계절의 옷차림 목록을 출력하는 기능

- 입력:

clothes: 옷차림(룩) 정보가 담긴 배열을 가리키는 포인터  
season: 현재 출력하고자 하는 계절을 가리키는 문자열  
numClothes: 해당 계절의 옷 개수를 나타내는 정수 값

- 설명:

함수는 특정 계절의 옷차림 목록을 출력하는 역할  
printf를 사용하여 해당 계절의 옷차림 목록을 출력합니다. 출력할 때는 해당 계절의 이름을 먼저 출력한 후, 그 계절에 해당하는 옷 정보를 리스트로 나열  
for반복문을 사용하여 해당 계절의 옷 정보를 배열에서 하나씩 가져와서 순차적으로 출력합니다. 각 옷 정보는 순서와 함께 출력

- 적용된 배운 내용:

반복문, 포인터,입출력 함수

- 코드스크린샷:

```
void printClothesForSeason(const Clothes* clothes, const char* season, int numClothes) {
    printf("\n%s 옷차림(룩) 목록:\n", season);
    for (int i = 0; i < numClothes; ++i) {
        printf("%d. %s\n", i + 1, clothes[i].name);
    }
}
```

**5) 구현한 기능 이름** 특정 계절의 옷 정보를 출력하고, 무작위로 선택된 옷차림을 추천하여 화면에 표시하고 동시에 파일에 저장

- 입력:

clothes: 옷차림 정보가 담긴 배열을 가리키는 포인터

season: 출력하고자 하는 계절을 가리키는 문자열

numClothes: 해당 계절의 옷 개수를 나타내는 정수 값

- 설명:

특정 계절의 옷차림 목록을 출력. 만약 해당 계절에 대한 옷 정보가 없을 경우, 이를 알리는 메시지를 출력하고 함수를 종료

rand() 함수를 이용하여 난수를 생성하여 해당 계절의 옷차림 배열에서 무작위로 하나의 옷차림을 선택

선택된 옷차림 정보를 화면에 출력하여 사용자에게 추천

"recommended\_outfit.txt"라는 파일에도 추천된 옷차림 정보를 추가로 기록

- 적용된 배운 내용:

조건문, 파일 입출력, 포인터, 배열

- 코드 스크린샷

```
void recommendOutfit(const Clothes* clothes, int numClothes, const char* season) {
    if (numClothes == 0) {
        printf("추천할 수 없습니다. %s에 대한 옷차림(복)이 없습니다.\n", season);
        return;
    }

    srand(time(NULL));
    int randomIndex = rand() % numClothes;

    printf("\n현재 날씨에 따른 옷차림(복)을 추천합니다: %s - %s\n", season, clothes[randomIndex].name);
    const char* fileName = "recommended_outfit.txt";

    FILE* file;
    errno_t err = fopen_s(&file, fileName, "a"); // 파일 열기 시도

    if (err != 0) {
        printf("파일을 열 수 없습니다.\n");
        return;
    }

    // 결과를 파일에 쓰기
    fprintf(file, "\n현재 날씨에 따른 옷차림(복)을 추천합니다: %s - %s\n", season, clothes[randomIndex].name);

    // 파일 닫기
    fclose(file);
}
```

**6) 구현한 기능 이름** 각 계절에 대해 필요로 하는 추가적인 옷 구매를 권장하는 기능과 파일에 저장

- 입력:

numSpringClothes: 봄 계절의 현재 옷 개수를 나타내는 정수

numSummerClothes: 여름 계절의 현재 옷 개수를 나타내는 정수

numFallClothes: 가을 계절의 현재 옷 개수를 나타내는 정수

numWinterClothes: 겨울 계절의 현재 옷 개수를 나타내는 정수

- 설명:

함수는 각 계절별로 현재 보유한 옷 개수를 확인하고, 추가로 필요한 옷의 개수를 계산하여 사용자에게 권장

만약 어떤 계절의 옷 개수가 최대 옷 개수(MAX\_CLOTHES)보다 적다면, 해당 계절의 옷을 추가로 구매하라는 메시지를 출력하고, 파일에도 동일한 내용을 추가 기록

파일 이름은 "recommended\_clothes.txt"로 설정되며, 파일에는 추가적인 옷 구매 권장 사항이 기록

- 적용된 배운 내용:

조건문, 파일 입출력

-코드 스크린샷:

```
void recommendAdditionalClothes(int numSpringClothes, int numSummerClothes,
int numFallClothes, int numWinterClothes) {
    printf("\n추가적인 옷을 구매하시길 권장하는 계절:\n");

    // 파일 이름 설정
    const char* fileName = "recommended_clothes.txt";

    FILE* file;
    errno_t err = fopen_s(&file, fileName, "a"); // 파일 열기 시도

    if (err != 0) {
        printf("파일을 열 수 없습니다.\n");
        return;
    }

    if (numSpringClothes < MAX_CLOTHES) {
        printf("- 봄 옷을 %d개 더 구매하십시오.\n", MAX_CLOTHES - numSpringClothes);
        fprintf(file, "- 봄 옷을 %d개 더 구매하십시오.\n", MAX_CLOTHES - numSpringClothes);
    }

    if (numSummerClothes < MAX_CLOTHES) {
        printf("- 여름 옷을 %d개 더 구매하십시오.\n", MAX_CLOTHES - numSummerClothes);
        fprintf(file, "- 여름 옷을 %d개 더 구매하십시오.\n", MAX_CLOTHES - numSummerClothes);
    }

    if (numFallClothes < MAX_CLOTHES) {
        printf("- 가을 옷을 %d개 더 구매하십시오.\n", MAX_CLOTHES - numFallClothes);
        fprintf(file, "- 가을 옷을 %d개 더 구매하십시오.\n", MAX_CLOTHES - numFallClothes);
    }

    if (numWinterClothes < MAX_CLOTHES) {
        printf("- 겨울 옷을 %d개 더 구매하십시오.\n", MAX_CLOTHES - numWinterClothes);
        fprintf(file, "- 겨울 옷을 %d개 더 구매하십시오.\n", MAX_CLOTHES - numWinterClothes);
    }

    fclose(file); // 파일 닫기
}
```

7) 구현한 기능 이름 현재 기온을 기준으로 계절을 판단하고, 해당 계절에 맞는 옷차림을 추천하는 기능

- 입력:

springClothes: 봄 계절의 옷 정보가 담긴 배열을 가리키는 포인터

numSpringClothes: 봄 계절의 옷 개수를 나타내는 정수 값

summerClothes: 여름 계절의 옷 정보가 담긴 배열을 가리키는 포인터

numSummerClothes: 여름 계절의 옷 개수를 나타내는 정수 값

fallClothes: 가을 계절의 옷 정보가 담긴 배열을 가리키는 포인터

numFallClothes: 가을 계절의 옷 개수를 나타내는 정수 값

winterClothes: 겨울 계절의 옷 정보가 담긴 배열을 가리키는 포인터



numWinterClothes: 겨울 계절의 옷 개수를 나타내는 정수 값

- 설명:

사용자로부터 현재 기온을 입력받음

입력된 기온을 기준으로 if-else if-else구문을 사용하여 계절을 판단

각 계절에 해당하는 기온 범위에 따라 해당 계절의 문자열을 설정하고,

recommendOutfit함수를 호출하여 해당 계절에 맞는 옷차림을 추천

- 적용된 배운 내용:

조건문, 포인터, 배열

-코드 스크린샷:

```
void determineSeason(const Clothes* springClothes, int numSpringClothes,
    const Clothes* summerClothes, int numSummerClothes,
    const Clothes* fallClothes, int numFallClothes,
    const Clothes* winterClothes, int numWinterClothes) {
    int temperature;
    printf("\n현재 기온을 입력해 주세요: ");
    scanf_s("%d", &temperature);

    const char* season;
    if (temperature >= 23) {
        season = "여름";
        recommendOutfit(summerClothes, numSummerClothes, season);
    }
    else if (temperature >= 12 && temperature < 23) {
        season = "봄";
        recommendOutfit(springClothes, numSpringClothes, season);
    }
    else if (temperature >= 5 && temperature < 12) {
        season = "가을";
        recommendOutfit(fallClothes, numFallClothes, season);
    }
    else {
        season = "겨울";
        recommendOutfit(winterClothes, numWinterClothes, season);
    }
}
```

**8) 구현한 기능 이름** 가진 옷 리스트를 출력하고, 각 계절별 옷 정보를 출력한 후  
추가적으로 필요한 옷을 권장하고 현재 계절을 결정하는 기능(여러 함수 통합)

- 입력:

springClothes: 봄 계절의 옷 정보가 담긴 배열을 가리키는 포인터

numSpringClothes: 봄 계절의 옷 개수를 나타내는 정수 값

summerClothes: 여름 계절의 옷 정보가 담긴 배열을 가리키는 포인터

numSummerClothes: 여름 계절의 옷 개수를 나타내는 정수 값

fallClothes: 가을 계절의 옷 정보가 담긴 배열을 가리키는 포인터

numFallClothes: 가을 계절의 옷 개수를 나타내는 정수 값

winterClothes: 겨울 계절의 옷 정보가 담긴 배열을 가리키는 포인터

numWinterClothes: 겨울 계절의 옷 개수를 나타내는 정수 값

- 설명:

함수는 각 계절별로 가진 옷 리스트를 출력하기 위해 printClothesForSeason 함수를 호출

이후, recommendAdditionalClothes 함수를 호출하여 각 계절별로 추가적으로 필요한 옷을 권장

마지막으로, determineSeason 함수를 호출하여 현재 기온을 기준으로 현재 계절을 결정하고 해당 계절에 맞는 옷차림을 추천

- 적용된 배운 내용:

포인터, 배열

-코드 스크린샷:

```
void printTotalClothes(const Clothes* springClothes, int numSpringClothes,
    const Clothes* summerClothes, int numSummerClothes,
    const Clothes* fallClothes, int numFallClothes,
    const Clothes* winterClothes, int numWinterClothes) {
    printf("\n가진 옷 리스트:\n");
    printClothesForSeason(springClothes, "봄", numSpringClothes);
    printClothesForSeason(summerClothes, "여름", numSummerClothes);
    printClothesForSeason(fallClothes, "가을", numFallClothes);
    printClothesForSeason(winterClothes, "겨울", numWinterClothes);

    recommendAdditionalClothes(numSpringClothes, numSummerClothes, numFallClothes, numWinterClothes);

    determineSeason(springClothes, numSpringClothes, summerClothes, numSummerClothes,
        fallClothes, numFallClothes, winterClothes, numWinterClothes);
}
```

## 2) 테스트 결과

1) 테스트한 기능 이름 특정 계절의 옷 정보를 출력하고, 무작위로 선택된 옷차림을 추천하여 화면에 표시하고 동시에 파일에 저장

const char\* fileName = "recommended\_outfit.txt"; 파일 이름을 "recommended\_outfit.txt"로 설정

FILE\* file; errno\_t err = fopen\_s(&file, fileName, "a"); 파일을 열기 위한 포인터 file을 생성하고, fopen\_s 함수를 사용하여 파일을 "a" (추가 모드)로 엽니다. 파일 열기에 대한 오류 코드는 err에 저장

if (err != 0) { printf("파일을 열 수 없습니다.\n"); return; }; 파일 열기에 실패한 경우, 오류 메시지를 출력하고 함수를 종료

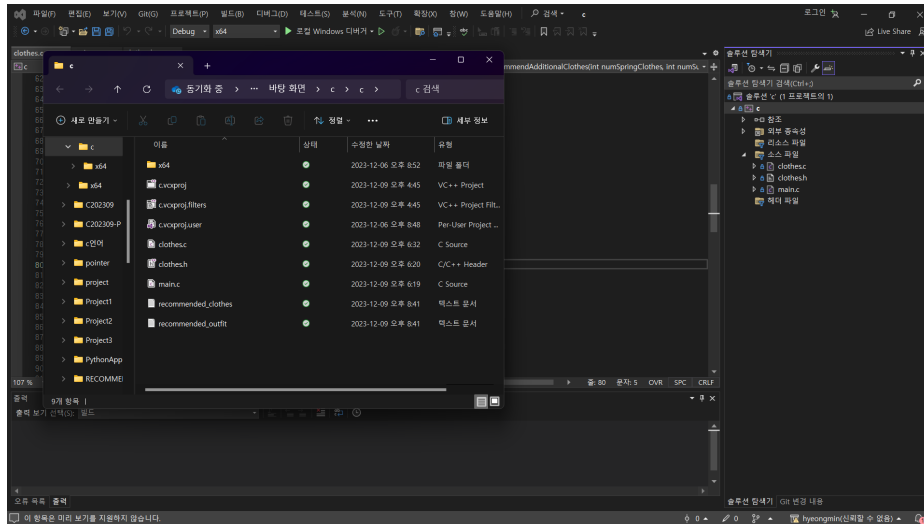
fprintf(file, "\n현재 날씨에 따른 옷차림(룩)을 추천합니다: %s - %s\n", season, clothes[randomIndex].name); 파일에 내용을 기록 fprintf 함수를 사용하여 파일에 현재 날씨에

다른 옷 추천 정보를 형식에 맞게 기록

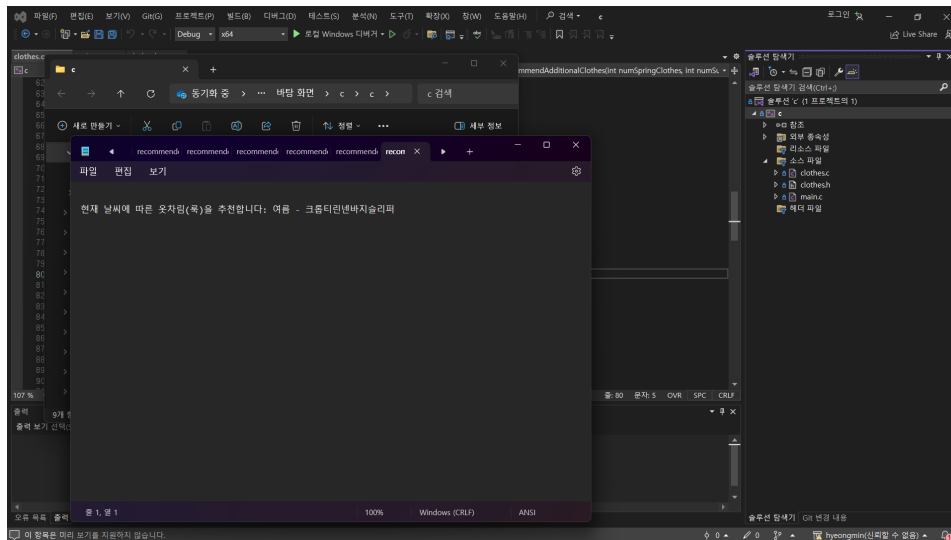
fclose(file);: 파일을 닫기

-코드 스크린샷

-recommended\_outfit 파일 생성



-recommended\_outfit 파일에 내용 저장



2) 테스트한 기능 이름 각 계절에 대해 필요로 하는 추가적인 옷 구매를 권장하는 기능과 파일에 저장

const char\* fileName = "recommended\_clothes.txt";: 파일 이름을 "recommended\_clothes.txt"로 설정

FILE\* file; errno\_t err = fopen\_s(&file, fileName, "a"); 파일을 열기 위한 포인터 file을 생성하고, fopen\_s함수를 사용하여 파일을 "a" (추가 모드)로 엽니다. 파일 열기에 대한 오류 코드는 err에 저장

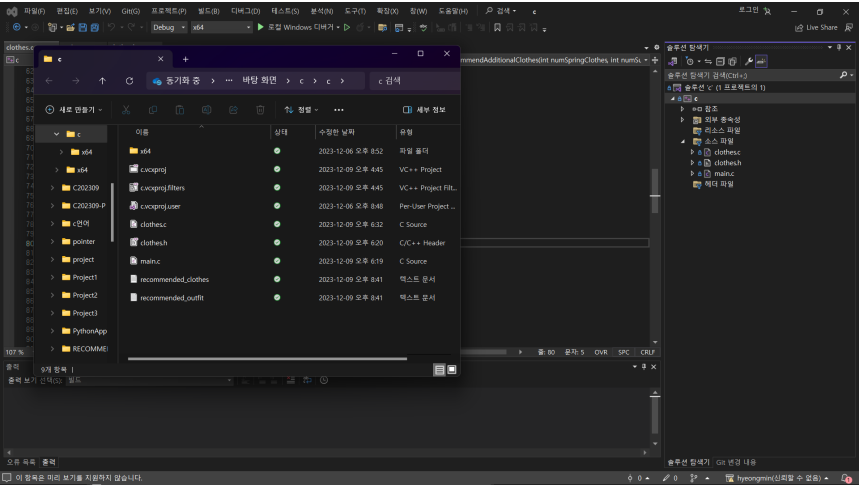
if (err != 0) { printf("파일을 열 수 없습니다.\n"); return; }; 파일 열기에 실패한 경우, 오류 메시지를 출력하고 함수를 종료

각 계절별로 필요한 추가 옷 개수를 확인하고, 필요한 만큼의 옷을 권장하는 메시지를 출력하고 파일에 기록 각 계절의 추가 옷 필요 개수가 MAX\_CLOTHES보다 적을 경우, 필요한 옷의 개수와 메시지를 출력하고 파일에도 같은 내용을 기록

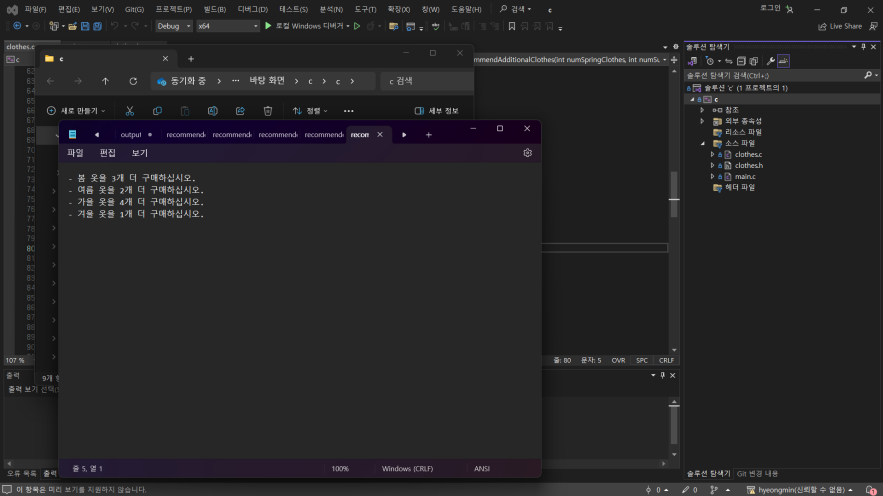
fclose(file); 파일을 닫기

-코드 스크린샷

- recommended\_clothes 파일 생성



- recommended\_clothes 파일에 내용 저장



## **4. 계획 대비 변경 사항**

### **1) 변경 내역 제목**

- 전체적인 코드를 이중 배열을 이용한 것에서 구조체와 동적 메모리를 이용하게끔 수정

### **2) 변경 내역 제목추천된 옷&구매 옷 저장 기능 (추가)**

- (recommendOutfit, recommendAdditionalClothes)의 결과를 txt파일에 저장
- 사용자가 입력한 정보를 바탕으로 기온에 따른 계절을 판단하여 추천한 옷과 부족한 계절별 옷을 알려주는 출력물을 각 txt파일을 만들어 저장

## **5. 프로젝트 일정**

(진행한 작업과 진행 중인 작업 등을 표기)

업무		11-26	12-3	12-10	.....
기능1,2,3		완료			
새로운 기능 1	헤더파일, 소스파일, 메인 파일 나누기 (커밋 잊어버림)		-->		
새로운 기능 2	구조체& 동적메모리 이용			--->	