

C프로그래밍및실습

# 기온에 따른 옷차림 (록) 추천 프로그램

최종 보고서

제출일자: 2023년 12월 15일

제출자명: 김형민

제출자학번: 233332

## **1. 프로젝트 목표**

### **1) 배경 및 필요성**

최근의 지구 온난화는 일상 생활의 다양한 측면에 영향을 미치고 있습니다. 이에 따라 온도의 변화는 개인의 의류 선택에 영향을 미치게 되어, 옷차림 선택이 어려워지고 있습니다. 옷차림은 우리의 편안함과 만족도에 직접적인 영향을 미치며, 기온과의 괴리는 이를 더욱 복잡하게 만들고 있습니다. 기후 변화와 옷차림 선택 사이의 연관성은 분명히 존재하며, 이는 일상적인 상황에서도 영향을 미칩니다. 고객이 옷을 선택할 때, 현재의 기후 상황과 그에 맞는 적절한 옷차림을 선택하는 것이 중요합니다. 그러나 기후 변화로 인해 예측하기 어려운 기온 변화로 인해 옷을 선택하는 것은 어려움을 겪고 있습니다. 이러한 문제를 해결하기 위해 기온과 개인의 보유 의류를 고려하여 적합한 옷을 추천하는 프로그램을 개발하고자 합니다.

### **2) 프로젝트 목표**

현재 기온 정보를 기반으로 사용자에게 적합한 옷차림을 추천하고, 사용자의 보유 의류를 고려하여 부족한 계절의 옷차림 개수를 알려주는 프로그램을 개발하는 것을 목표로 한다.

### **3) 차별점**

이 프로그램의 독특한 점은 사용자 맞춤형 옷 추천 및 부족한 옷 파악을 위해 개인 정보와 기후를 고려한 기능을 가지고 있습니다. 먼저 사용자가 보유한 의류 정보를 입력 받고, 현재 기후를 확인하여 그에 맞는 옷차림을 추천합니다. 또한, 사용자가 보유한 옷의 부재를 식별하고 이를 파일에 저장하여 나중에 참고할 수

있도록 합니다. 이러한 기능은 사용자의 옷 선택을 향상시키고 부족한 옷을 쉽게 식별할 수 있도록 돕는 특징을 가지고 있습니다. 두번째로 사용자의 개인 정보와 실시간 기후 데이터를 반영하여 맞춤형 옷차림을 제안하는데 특화되어 있습니다. 사용자가 가진 의류 정보를 기반으로 현재 기후에 적합한 옷을 추천하며, 추가적으로 부족한 계절의 옷 개수를 추적하여 부재를 파악하고 추천합니다. 이뿐만 아니라, 추천된 옷차림과 부족한 옷의 정보를 파일로 저장하여 사용자가 언제든지 확인할 수 있도록 지원합니다. 이러한 기능들은 사용자에게 맞춤형 옷 선택과 필요한 아이템 파악을 돕는 독특한 점으로 프로그램을 특별하게 만들어줍니다.

## 2. 기능 계획

### 1) 기능 1 옷 정보 입력

- 사용자로부터 옷 정보를 입력받는 기능

(1) 세부 기능 1: 옷 정보 동적 메모리 할당

- 사용자로부터 입력된 옷 정보를 저장하기 위해 메모리를 동적으로 할당. 이를 통해 입력된 옷 정보들을 보관할 메모리 공간을 마련

(2) 세부 기능 2: 최대 옷 개수까지 정보 입력

- 사용자에게 최대 옷 개수만큼 입력을 요청. 사용자가 원하는 만큼 옷 정보를 입력할 수 있도록 함

(3) 세부 기능 3: 옷 정보 구조체 배열에 저장

- 입력된 옷 정보들을 구조체 배열에 순서대로 저장

### 2) 기능 2 계절별 옷 리스트 출력

- 특정 계절의 옷 리스트를 출력하는 기능

(1) 세부 기능 1: 특정 계절의 옷 리스트를 받아 출력

해당 계절의 옷 리스트를 받아와 사용자에게 출력

(2) 세부 기능 2: 입력된 옷들을 순서대로 번호와 함께 출력

- 입력된 옷들을 순서대로 번호와 함께 화면에 출력하여 사용자가 확인할 수 있도록 함

### **3) 기능 3 날씨에 따른 옷 추천**

- 현재 날씨에 맞게 옷을 추천하는 기능

(1) 세부 기능 1: 사용자가 입력한 옷 중에서 추천할 옷 선택

- 사용자가 입력한 옷 정보 중에서 난수를 활용하여 추천할 옷을 선택

(2) 세부 기능 2: 선택된 옷과 현재 계절 정보 출력 및 파일 저장

- 선택된 옷과 해당하는 계절 정보를 출력하고 이를 파일에 저장하여 추천된 옷 정보를 기록

### **4) 기능 4 추가 옷 구매 권장**

- 사용자가 부족한 계절의 옷을 확인하고, 부족한 만큼 구매를 권장하는 기능

(1) 세부 기능 1: 각 계절별 보유한 옷 개수 확인

- 봄, 여름, 가을, 겨울 각 계절별 보유한 옷 개수를 확인

(2) 세부 기능 2: 부족한 옷 개수에 따른 권장 사항 출력 및 파일 저장

- 부족한 옷 개수를 확인하고, 부족한 만큼의 옷을 권장하여 화면에 출력하고 동시에 파일에 저장

### **5) 기능 5 현재 계절에 따른 옷 추천**

- 입력된 기온에 따라 현재 계절을 판단하고, 해당 계절에 맞는 옷을 추천하는 기능

(1) 세부 기능 1: 사용자로부터 현재 기온 입력

- 사용자로부터 현재 기온을 입력받아 계절을 판단하는 기준으로 활용

(2) 세부 기능 2: 계절 판단

- 입력된 기온에 따라 계절을 판단

(3) 세부 기능 3: 해당 계절의 옷 추천

- 판단된 계절에 맞는 옷을 추천하고, 이를 사용자에게 출력하고 파일에 저장

## **6) 기능 6 전체 옷 리스트 출력 및 부족한 옷 확인**

- 사용자가 보유한 전체 옷 리스트를 출력하고, 부족한 옷을 확인하는 기능

### **(1) 세부 기능 1: 각 계절별 보유한 옷 리스트**

- 봄, 여름, 가을, 겨울 각 계절별로 보유한 옷 리스트를 화면에 출력

### **(2) 세부 기능 2: 각 계절별 부족한 옷 개수 확인 및 파일 저장**

- 각 계절별 보유한 옷 개수를 확인하고, 부족한 옷의 개수를 계산하여 화면에 출력하고 동시에 파일에 저장

### **(3) 세부 기능 3: 부족한 옷을 추천하는 함수 호출**

- 부족한 옷을 추천하는 함수를 호출하여 사용자에게 구매를 권장

## **7) 기능 7 옷의 개수와 옷 이름 길이 정의 및 기능1,2,3,4,5,6을 가지는 헤더 파일**

- 옷의 개수와 옷 이름을 담을 수 있는 길이를 정의하고 기능 1,2,3,4,5,6을 가지는 헤더파일

## **8) 기능 8 옷 정보를 입력받고, 각 계절에 따른 옷 리스트를 출력하며 부족한 옷을 확인하는 프로그램의 메인 파일**

- 사용자로부터 봄, 여름, 가을, 겨울에 해당하는 옷 정보를 입력받고, 이후 각 계절별로 보유한 옷 리스트를 출력하고 부족한 옷을 확인. 마지막으로, 할당된 메모리를 해제하여 메모리 누수를 방지

### 3. 기능 구현

#### (1) 구현한 기능 이름 옷 정보 입력

- 입력: 사용자로부터 옷 정보를 입력받음 (getClothes)
- 출력: 사용자에게 정보를 안내하고 입력된 정보를 확인할 수 있도록 함.
- 출력내용: 옷차림(룩)을 입력해주세요: (최대 5가지, 띄어쓰기 사용 X)예시) 검정맨투맨슬랙스단화
- 이 기능은 사용자로부터 여러 계절의 옷 정보를 입력받아 해당하는 계절에 맞는 옷 리스트를 만들어줌. 사용자에게 옷 정보를 입력하는 방법을 안내하고, 사용자가 입력한 정보를 배열에 저장하여 계절별로 관리
- 적용된 배운 내용: 동적 메모리 할당(malloc), 반복문(while), 조건문(if), 포인터
- 코드 스크린샷

```
// 사용자로부터 옷 정보를 입력받는 함수
void getClothes(Clothes** clothes, const char* season, int* numClothes) {
    printf("%s 옷차림(룩)을 입력해주세요: (최대 %d가지, 띄어쓰기 사용 X)\n예시) 검정맨투맨슬랙스단화\n", season, MAX_CLOTHES);

    *clothes = (Clothes*)malloc(MAX_CLOTHES * sizeof(Clothes)); // 메모리 할당

    if (*clothes == NULL) {
        printf("메모리 할당 오류\n");
        exit(1);
    }

    while (*numClothes < MAX_CLOTHES) {
        printf("옷차림(룩) %d: ", *numClothes + 1);
        scanf_s("%s", (*clothes)[*numClothes].name, MAX_LENGTH); // 옷 정보 입력

        (*numClothes)++; // 옷 개수 증가

        printf("더 입력하시겠습니까? (종료시: n or N/더 입력하려면 아무키나 누르시오): ");
        char choice;
        scanf_s("%c", &choice, sizeof(choice));

        if (choice == 'n' || choice == 'N') {
            break;
        }
    }
}
```

#### (2) 구현한 기능 이름 계절별 옷 리스트 출력

- 출력: 해당 계절의 옷 리스트를 화면에 출력 각 옷에 대해 인덱스와 옷 이름 출력
- 출력내용: "[ ] 옷차림(룩) 목록; 계절 정보 ( [ ]은 계절)

- 이 기능은 특정 계절에 속하는 옷 리스트를 사용자에게 보여줌. 계절에 해당하는 옷 정보가 저장된 배열과 해당 계절의 이름이 함수에 전달. 함수는 해당 계절의 옷 리스트를 번호와 함께 화면에 출력하는 역할을 수행
- 적용된 배운 내용: 반복문(for), 포인터
- 코드 스크린샷

```
// 해당 계절의 옷 리스트를 출력하는 함수
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);
    }
}
```

### (3) 구현한 기능 이름 날씨에 따른 옷 추천

- 출력: 추천된 옷 리스트를 화면에 출력하고, 해당 내용을 파일에 저장
- 해당 기능은 먼저 입력된 옷 정보 배열에서 날씨에 따른 옷을 랜덤으로 선택하여 화면에 출력. 선택된 옷 정보는 해당 계절과 함께 출력되고, 또한 "recommended\_outfit.txt"라는 파일에도 해당 내용이 저장. 파일 열기를 시도하고 실패한 경우에는 파일을 열 수 없다는 메시지가 출력.
- 적용된 배운 내용: 파일 입출력(txt파일), 조건문(if)
- 코드 스크린샷

```
// 현재 날씨에 따라 옷을 추천하는 함수
void recommendOutfit(const Clothes* clothes, int numClothes, const char* season) {
    srand(time(NULL));
    int randomIndex = rand() % numClothes;

    printf("\n현재 날씨에 따른 옷차림(룩)을 추천합니다: %s - %s\n", season, clothes[randomIndex].name);
    // 파일 이름 설정
    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(룩)을 추천합니다: %s - %s\n", season, clothes[randomIndex].name);

    // 파일 닫기
    fclose(file);
}
```

#### (4) 구현한 기능 이름 추가 옷 구매 권장

- 출력: 추가로 구매 권장하는 옷의 개수 및 계절
- 출력 내용: 추가적인 옷을 구매하시길 권장하는 계절: [ ], 봄 옷을 n개 더 구매하십시오. [ ] 옷을 n개 더 구매하십시오. ([ ]는 계절 n은 숫자)
- 이 기능은 각 계절별로 현재 보유한 옷의 개수와 최대 옷 개수(MAX\_CLOTHES)를 비교, 부족한 만큼 어떤 계절의 옷을 더 구매하는 것이 좋을지를 사용자에게 안내하고, "recommended\_clothes.txt" 파일에도 동일한 내용을 저장
- 적용된 배운 내용: 파일 입출력(txt파일), 조건문(if)
- 코드 스크린샷

```
// 추가적인 옷을 권장하는 함수
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
}
```



## (5) 구현한 기능 이름 현재 계절에 따른 옷 추천

- 입력: 현재 기온 입력
- 출력: 추천된 현재 계절의 옷 정보를 화면에 출력
- 출력 내용: 현재 기온을 입력해 주세요
- 함수는 사용자로부터 현재 기온을 입력받은 후, 기온에 따라 계절을 판단. 판단된 계절에 맞게 옷 추천 함수(recommendOutfit)를 호출하여 해당 계절에 어울리는 옷을 화면에 출력.
- 적용된 배운 내용: 조건문(if else), 포인터
- 코드 스크린샷

```
// 현재 계절을 판단하고 옷을 추천하는 함수
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);
    }
}
```

## (6) 구현한 기능 이름 전체 옷 리스트 출력 및 부족한 옷 확인

- 출력: 현재 보유하고 있는 옷 리스트를 화면에 출력하고, 부족한 옷을 각 계절에 대해 화면에 출력

- 출력 내용: 가진 옷 리스트

봄

여름

가을

겨울

- "가진 옷 리스트"라는 메시지를 출력한 후, 각 계절별 옷 리스트를 출력하기 위해 printClothesForSeason함수를 호출합니다. 이 함수는 각 계절의 옷을 화면에 출력. 부족한 옷을 추천하기 위해 recommendAdditionalClothes함수를 호출. 이 함수는 각 계절별로 부족한 옷의 개수를 확인하고, 부족한 만큼 옷을 더 구매할 것을 사용자에게 권장. determineSeason함수를 호출하여 현재 계절에 따른 옷을 추천. 이 함수는 현재 기온을 입력받고, 그에 맞는 계절에 해당하는 옷을 사용자에게 추천

- 적용된 배운 내용: 구조체

- 코드 스크린샷

```
// 전체 옷 리스트를 출력하고 부족한 옷을 추천하는 함수
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);
}
```

(7) 구현한 기능 이름 옷의 개수와 옷 이름 길이 정의 및 기능1,2,3,4,5,6을 가지는 헤더 파일

- 해당 헤더파일은 MAX\_CLOTHES와 MAX\_LENGTH는 각각 옷의 최대 개수와 옷 이름의 최대 길이를 정의하는 매크로 상수

Clothes구조체는 옷의 이름을 담는 배열로 구성

getClothes: 사용자로부터 옷 정보를 입력받는 함수 (기능1)

printClothesForSeason: 특정 계절의 옷 리스트를 출력하는 함수 (기능2)

recommendOutfit: 현재 계절에 따라 옷을 추천하는 함수 (기능3)

recommendAdditionalClothes: 부족한 옷을 추천하여 추가 구매를 권장하는 함수 (기능4)

determineSeason: 현재 계절을 판단하고 해당 계절에 맞는 옷을 추천하는 함수 (기능5)

printTotalClothes: 전체 옷 리스트를 출력하고 부족한 옷을 확인하는 함수 (기능6)

- 적용된 배운 내용: 구조체, 배열, 헤더파일

- 코드 스크린샷

```
#ifndef CLOTHES_H
#define CLOTHES_H

#define MAX_CLOTHES 5
#define MAX_LENGTH 50

typedef struct {
    char name[MAX_LENGTH];
} Clothes;

// 사용자로부터 옷 정보를 입력받는 함수
void getClothes(Clothes** clothes, const char* season, int* numClothes);

// 해당 계절의 옷 리스트를 출력하는 함수
void printClothesForSeason(const Clothes* clothes, const char* season, int numClothes);

// 현재 날씨에 따라 옷을 추천하는 함수
void recommendOutfit(const Clothes* clothes, int numClothes, const char* season);

// 추가 옷 구매를 권장하는 함수
void recommendAdditionalClothes(int numSpringClothes, int numSummerClothes,
    int numFallClothes, int numWinterClothes);

// 현재 계절을 판단하고 옷을 추천하는 함수
void determineSeason(const Clothes* springClothes, int numSpringClothes,
    const Clothes* summerClothes, int numSummerClothes,
    const Clothes* fallClothes, int numFallClothes,
    const Clothes* winterClothes, int numWinterClothes);

// 전체 옷 리스트를 출력하고 부족한 옷을 확인하는 함수
void printTotalClothes(const Clothes* springClothes, int numSpringClothes,
    const Clothes* summerClothes, int numSummerClothes,
    const Clothes* fallClothes, int numFallClothes,
    const Clothes* winterClothes, int numWinterClothes);

#endif
```

헤더파일&전처리기

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

**(8) 구현한 기능 이름** 옷 정보를 입력받고, 각 계절에 따른 옷 리스트를 출력하며 부족한 옷을 확인하는 프로그램의 메인 파일

- 네 가지 계절(봄, 여름, 가을, 겨울)에 대한 옷 정보를 저장할 포인터와 해당 계절 옷의 개수를 추적하기 위한 변수를 선언

getClothes()함수를 호출하여 각 계절에 대한 옷 정보를 입력

printTotalClothes()함수를 호출하여 각 계절별 옷 리스트를 출력하고 부족한 옷을 확인

마지막으로, 동적으로 할당된 메모리를 해제하고 프로그램을 종료

- 적용된 배운 내용: 동적메모리 해체, 구조체, 헤더파일

- 코드 스크린샷

```
#include "clothes.h"
#include <stdio.h>
#include <stdlib.h>

int main() {
    Clothes* springClothes, * summerClothes, * fallClothes, * winterClothes;
    int numSpringClothes = 0, numSummerClothes = 0, numFallClothes = 0, numWinterClothes = 0;

    // 봄 옷 정보 입력
    getClothes(&springClothes, "봄", &numSpringClothes);

    // 여름 옷 정보 입력
    getClothes(&summerClothes, "여름", &numSummerClothes);

    // 가을 옷 정보 입력
    getClothes(&fallClothes, "가을", &numFallClothes);

    // 겨울 옷 정보 입력
    getClothes(&winterClothes, "겨울", &numWinterClothes);

    // 각 계절별 옷 리스트 출력 및 부족한 옷 확인
    printTotalClothes(springClothes, numSpringClothes, summerClothes, numSummerClothes,
        fallClothes, numFallClothes, winterClothes, numWinterClothes);

    // 할당된 메모리 해제
    free(springClothes);
    free(summerClothes);
    free(fallClothes);
    free(winterClothes);

    return 0;
}
```

## 4. 테스트 결과

### (1) 테스트한 기능 이름 옷 정보 입력

- void getClothes(Clothes\*\* clothes, const char\* season, int\* numClothes) {:  
getClothes함수는 세 가지 파라미터를 받음

Clothes\*\* clothes: 옷 정보를 저장할 포인터 배열을 가리키는 이중 포인터

const char\* season: 계절 정보를 나타내는 문자열

int\* numClothes: 옷의 개수를 저장하는 변수를 가리키는 포인터

printf("%s 옷차림(룩)을 입력해주세요: (최대 %d가지, 띄어쓰기 사용 X)\n", season, MAX\_CLOTHES);  
검정맨투 맨슬랙스단화\n", season, MAX\_CLOTHES);: 사용자에게 옷차림(룩) 정보를 입력하라는 메시지를 출력

season은 입력된 계절 정보를 나타내며, MAX\_CLOTHES는 최대 옷의 가짓수를 나타냄

\*clothes = (Clothes\*)malloc(MAX\_CLOTHES \* sizeof(Clothes));  
malloc함수를 사용하여 MAX\_CLOTHES만큼의 Clothes구조체 배열을 할당

if (\*clothes == NULL) { printf("메모리 할당 오류\n"); exit(1); }  
할당된 메모리가 제대로 할당되었는지 확인하고, 만약 메모리 할당에 실패한 경우 오류 메시지를 출력하고 프로그램을 종료

while (\*numClothes < MAX\_CLOTHES) {:  
입력된 옷의 개수가 최대 옷 가짓수보다 적은 동안 반복

printf("옷차림(룩) %d: ", \*numClothes + 1);  
사용자에게 옷차림(룩) 정보를 입력하라는 메시지를 출력

scanf\_s("%s", (\*clothes)[\*numClothes].name, MAX\_LENGTH);  
사용자로부터 옷의 정보를

입력받아 name멤버 변수에 저장

(\*numClothes)++; 옷의 개수를 하나 증가시킴

printf("더 입력하시겠습니까? (종료시: n or N/더 입력하려면 아무키나 누르시오): "); 사용자에게 추가 입력 여부를 묻는 메시지를 출력

char choice; scanf\_s(" %c", &choice, sizeof(choice)); 사용자로부터 추가 입력 여부를 입력 받음

if (choice == 'n' || choice == 'N') { break; }; 만약 사용자가 'n' 또는 'N'을 입력하면 추가 입력을 멈춤

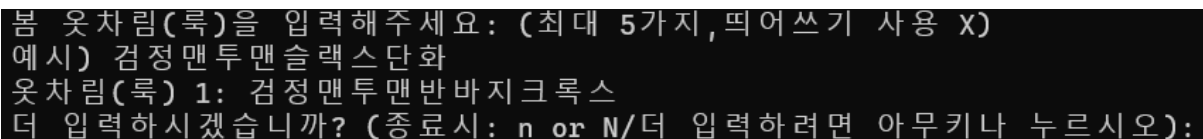
- 테스트 결과 스크린샷

사용자에게 옷차림(룩) 정보를 입력하라는 메시지를 출력



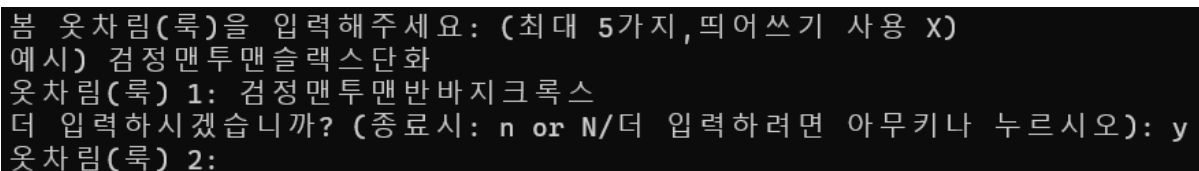
```
봄 옷차림(룩)을 입력해주세요: (최대 5가지, 띄어쓰기 사용 X)
예시) 검정맨투맨슬랙스단화
옷차림(룩) 1:
```

사용자에게 추가 입력 여부를 묻는 메시지를 출력



```
봄 옷차림(룩)을 입력해주세요: (최대 5가지, 띄어쓰기 사용 X)
예시) 검정맨투맨슬랙스단화
옷차림(룩) 1: 검정맨투맨반바지크룩스
더 입력하시겠습니까? (종료시: n or N/더 입력하려면 아무키나 누르시오):
```

사용자가 'n' 또는 'N'을 입력하지 않은 경우 -> 사용자에게 옷차림(룩) 정보를 입력하라는 메시지를 출력



```
봄 옷차림(룩)을 입력해주세요: (최대 5가지, 띄어쓰기 사용 X)
예시) 검정맨투맨슬랙스단화
옷차림(룩) 1: 검정맨투맨반바지크룩스
더 입력하시겠습니까? (종료시: n or N/더 입력하려면 아무키나 누르시오): y
옷차림(룩) 2:
```

사용자가 'n' 또는 'N'을 입력한 경우 -> 다음 계절로 넘어감

```
봄 옷차림(룩)을 입력해주세요: (최대 5가지, 띄어쓰기 사용 X)
예시) 검정맨투맨슬랙스단화
옷차림(룩) 1: 검정맨투맨반바지크룩스
더 입력하시겠습니까? (종료시: n or N/더 입력하려면 아무키나 누르시오): y
옷차림(룩) 2: 하얀반팔슬랙스단화
더 입력하시겠습니까? (종료시: n or N/더 입력하려면 아무키나 누르시오): n
여름 옷차림(룩)을 입력해주세요: (최대 5가지, 띄어쓰기 사용 X)
예시) 검정맨투맨슬랙스단화
옷차림(룩) 1:
```

각 계절에 대한 사용자의 옷을 모두 입력한 스크린샷

```
봄 옷차림(룩)을 입력해주세요: (최대 5가지, 띄어쓰기 사용 X)
예시) 검정맨투맨슬랙스단화
옷차림(룩) 1: 검정맨투맨반바지크룩스
더 입력하시겠습니까? (종료시: n or N/더 입력하려면 아무키나 누르시오): y
옷차림(룩) 2: 하얀반팔슬랙스단화
더 입력하시겠습니까? (종료시: n or N/더 입력하려면 아무키나 누르시오): n
여름 옷차림(룩)을 입력해주세요: (최대 5가지, 띄어쓰기 사용 X)
예시) 검정맨투맨슬랙스단화
옷차림(룩) 1: 파란색브이넥반팔린넨반바지슬리퍼
더 입력하시겠습니까? (종료시: n or N/더 입력하려면 아무키나 누르시오): o
옷차림(룩) 2: 나시형광바지운동화
더 입력하시겠습니까? (종료시: n or N/더 입력하려면 아무키나 누르시오): p
옷차림(룩) 3: 노란색라운드티셔츠치마쪼리
더 입력하시겠습니까? (종료시: n or N/더 입력하려면 아무키나 누르시오): N
가을 옷차림(룩)을 입력해주세요: (최대 5가지, 띄어쓰기 사용 X)
예시) 검정맨투맨슬랙스단화
옷차림(룩) 1: 코트청바지단화
더 입력하시겠습니까? (종료시: n or N/더 입력하려면 아무키나 누르시오): n
겨울 옷차림(룩)을 입력해주세요: (최대 5가지, 띄어쓰기 사용 X)
예시) 검정맨투맨슬랙스단화
옷차림(룩) 1: 과잠진청바지농구화
더 입력하시겠습니까? (종료시: n or N/더 입력하려면 아무키나 누르시오): m
옷차림(룩) 2: 패딩런닝바지러닝화
더 입력하시겠습니까? (종료시: n or N/더 입력하려면 아무키나 누르시오): n
```

## (2) 테스트한 기능 이름 계절별 옷 리스트 출력

```
- void printClothesForSeason(const Clothes* clothes, const char* season, int numClothes)
{
```

printClothesForSeason함수는 세 개의 파라미터를 받음

const Clothes\* clothes: 옷 정보를 담고 있는 구조체 배열을 가리키는 포인터

const char\* season: 출력할 옷차림의 계절 정보를 나타내는 문자열

int numClothes: 옷의 총 개수를 나타내는 변수

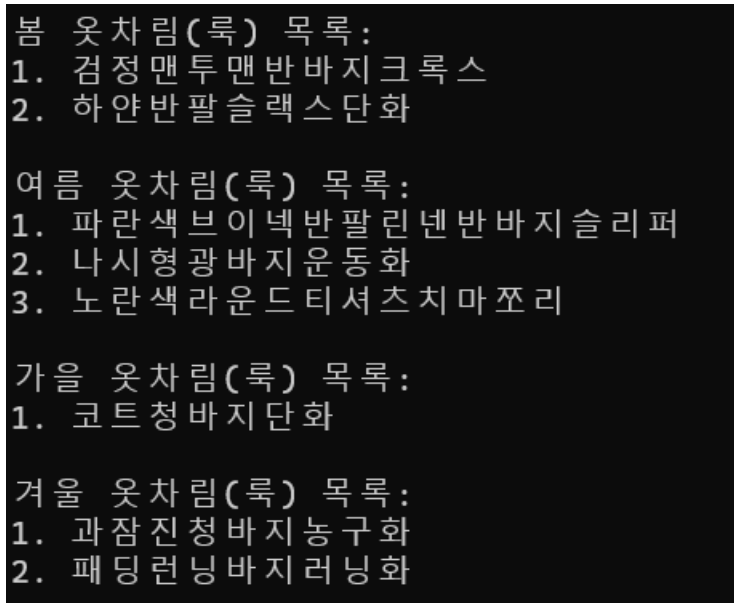
printf("%n%s 옷차림(룩) 목록:%n", season);: 해당 계절의 옷차림 목록을 출력하기 전에, 해당 계절의 이름을 출력

for (int i = 0; i < numClothes; ++i) {: 옷의 개수만큼 반복하는 루프

printf("%d. %s\n", i + 1, clothes[i].name);: 순서와 함께 해당 옷차림의 이름을 출력합니다. i + 1은 옷차림 리스트를 1부터 번호를 매기기 위함이고, clothes[i].name은 각 옷차림의 이름을 나타냄

- 테스트 결과 스크린샷

해당 계절의 옷차림 목록을 출력하기 전에, 해당 계절의 이름을 출력 & 순서와 함께 해당 옷차림의 이름을 출력



```
봄 옷차림(룩) 목록:
1. 검정맨투맨반바지크룩스
2. 하얀반팔슬랙스단화

여름 옷차림(룩) 목록:
1. 파란색브이넥반팔린넨반바지슬리퍼
2. 나시형광바지운동화
3. 노란색라운드티셔츠치마쪼리

가을 옷차림(룩) 목록:
1. 코트청바지단화

겨울 옷차림(룩) 목록:
1. 과잠진청바지농구화
2. 패딩런닝바지러닝화
```

### (3) 테스트한 기능 이름 날씨에 따른 옷 추천

- srand(time(NULL)); int randomIndex = rand() % numClothes;; 난수를 생성하여 clothes 배열에서 랜덤한 옷차림을 선택하기 위해 rand()함수를 사용

srand(time(NULL))는 매번 실행할 때마다 난수 시드를 바꾸어 다양한 난수가 생성되도록



함

rand() % numClothes는 배열의 인덱스 범위 내에서 랜덤한 값을 얻기 위해 사용

```
printf("\n현재 날씨에 따른 옷차림(룩)을 추천합니다: %s - %s\n", season,
clothes[randomIndex].name); 선택된 랜덤한 옷차림을 현재 날씨에 따라 추천하는 메시지를 출력
```

```
const char* fileName = "recommended_outfit.txt"; 결과를 저장할 파일의 이름을 지정
```

```
FILE* file; errno_t err = fopen_s(&file, fileName, "a"); 파일을 열기 위해 fopen_s 함수를 사용, "a" 모드는 파일을 추가 모드로 열어 기존 내용 끝에 새로운 내용을 추가
```

```
if (err != 0) { printf("파일을 열 수 없습니다.\n"); return; }; 파일을 열지 못한 경우 오류 메시지를 출력하고 함수를 종료
```

```
fprintf(file, "\n현재 날씨에 따른 옷차림(룩)을 추천합니다: %s - %s\n", season,
clothes[randomIndex].name); 선택된 추천 옷차림을 파일에 씀
```

```
fclose(file); 파일을 닫음
```

- 테스트 결과 스크린샷

선택된 랜덤한 옷차림을 현재 날씨에 따라 추천하는 메시지를 출력

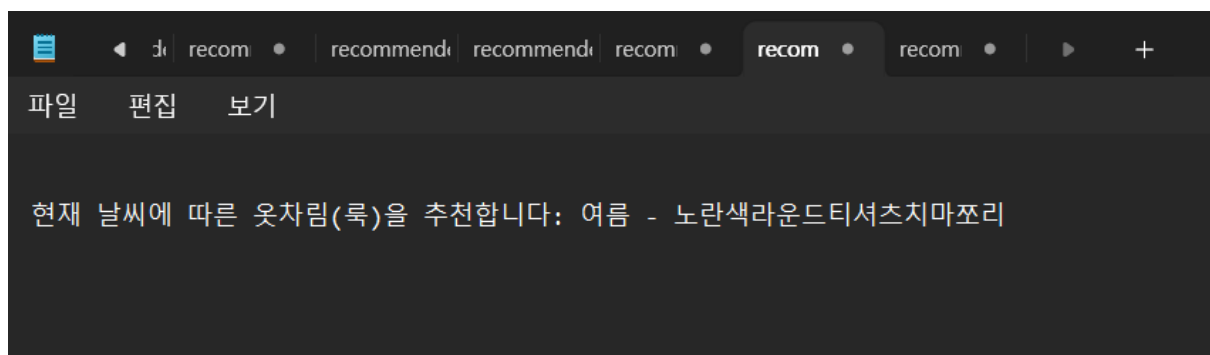
현재 기온을 입력해 주세요: 23

현재 날씨에 따른 옷차림(룩)을 추천합니다: 여름 - 노란색라운드티셔츠치마쪼리

결과를 저장할 파일의 이름을 지정&생성 (recommended\_outfit.txt)

이름	상태	수정한 날짜	유형
x64	✓	2023-12-06 오후 8:52	파일 폴더
c.vcxproj	✓	2023-12-09 오후 4:45	VC++ Project
c.vcxproj.filters	✓	2023-12-09 오후 4:45	VC++ Project Filt...
c.vcxproj.user	✓	2023-12-06 오후 8:48	Per-User Project ...
clothes.c	✓	2023-12-15 오후 11:30	C Source
clothes.h	✓	2023-12-15 오후 9:03	C/C++ Header
main.c	✓	2023-12-15 오후 9:05	C Source
recommended_clothes	✓	2023-12-15 오후 11:30	텍스트 문서
recommended_outfit	✓	2023-12-15 오후 11:30	텍스트 문서

선택된 추천 옷차림을 파일에 씀



#### (4) 테스트한 기능 이름 추가 옷 구매 권장

- printf("%n추가적인 옷을 구매하시길 권장하는 계절:%n");: 함수가 실행될 때, 추가적인 옷을 구매하는 것을 권장하는 계절을 출력

const char\* fileName = "recommended\_clothes.txt"; 결과를 저장할 파일의 이름을 설정

FILE\* file; errno\_t err = fopen\_s(&file, fileName, "a");: 파일을 열기 위해 fopen\_s 함수를 사용

if (err != 0) { printf("파일을 열 수 없습니다.%n"); return; }; 파일을 열지 못한 경우 오류

메시지를 출력하고 함수를 종료

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

각 계절에 대해 최대 옷의 가짓수 (MAX\_CLOTHES)보다 부족한 경우, 몇 개의 옷을 추가로 구매해야 하는지를 화면에 출력  
또한, 이 내용을 파일에도 기록

fclose(file); 파일을 닫음

- 테스트 결과 스크린샷

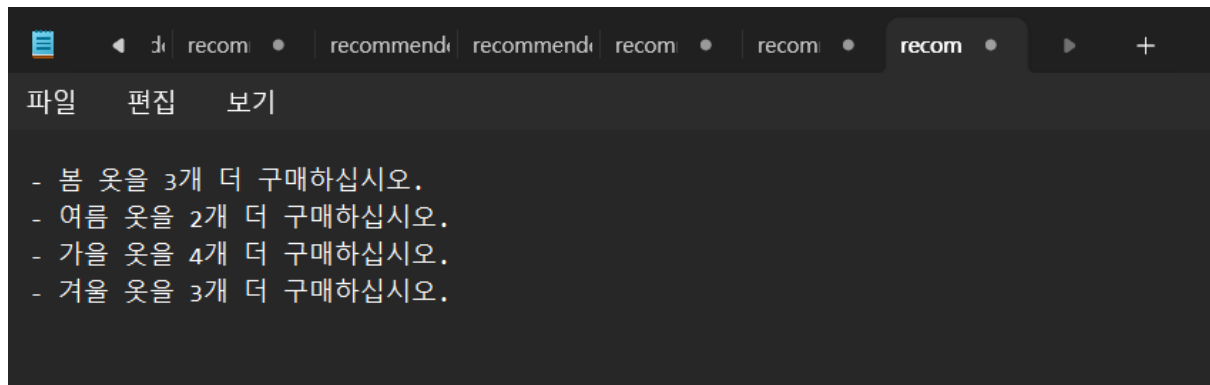
옷을 구매하는 것을 권장하는 계절을 출력&몇 개의 옷을 추가로 구매해야 하는지를 화면에 출력

```
추가적인 옷을 구매하시길 권장하는 계절 :  
- 봄 옷을 3개 더 구매하십시오.  
- 여름 옷을 2개 더 구매하십시오.  
- 가을 옷을 4개 더 구매하십시오.  
- 겨울 옷을 3개 더 구매하십시오.
```

결과를 저장할 파일의 이름을 지정&생성 (recommended\_clothes.txt)

이름	상태	수정한 날짜	유형
x64	✓	2023-12-06 오후 8:52	파일 폴더
c.vcxproj	✓	2023-12-09 오후 4:45	VC++ Project
c.vcxproj.filters	✓	2023-12-09 오후 4:45	VC++ Project Filt...
c.vcxproj.user	✓	2023-12-06 오후 8:48	Per-User Project ...
clothes.c	✓	2023-12-15 오후 11:30	C Source
clothes.h	✓	2023-12-15 오후 9:03	C/C++ Header
main.c	✓	2023-12-15 오후 9:05	C Source
recommended_clothes	✓	2023-12-15 오후 11:30	텍스트 문서
recommended_outfit	✓	2023-12-15 오후 11:30	텍스트 문서

계절별 추천 구매 옷 개수를 파일에 씀



#### (5) 테스트한 기능 이름 현재 계절에 따른 옷 추천

- int temperature; printf("\n현재 기온을 입력해 주세요: "); scanf\_s("%d", &temperature);  
사용자로부터 현재 기온을 입력받음

const char\* season;: 계절을 저장할 문자열 포인터를 선언

if (temperature >= 23) { season = "여름"; recommendOutfit(summerClothes, numSummerClothes, season); } : 현재 기온이 23도 이상인 경우, 여름으로 판단하고 recommendOutfit함수를 호출하여 여름 옷차림을 추천함

else if (temperature >= 12 && temperature < 23) { season = "봄"; recommendOutfit(springClothes, numSpringClothes, season); } : 현재 기온이 12도 이상 23도 미만인 경우, 봄으로 판단하고 봄 옷차림을 추천

else if (temperature >= 5 && temperature < 12) { season = "가을"; recommendOutfit(fallClothes, numFallClothes, season); } : 현재 기온이 5도 이상 12도 미만인 경우, 가을으로 판단하고 가을 옷차림을 추천

else { season = "겨울"; recommendOutfit(winterClothes, numWinterClothes, season); } : 위

의 조건들에 해당하지 않는 경우, 즉 남은 경우는 겨울로 판단하고 겨울 옷차림을 추천

- 테스트 결과 스크린샷

사용자로부터 현재 기온을 입력받음

현재 기온을 입력해 주세요:

봄으로 판단 (12도 이상 23도 미만)

현재 기온을 입력해 주세요: 17

현재 날씨에 따른 옷차림(룩)을 추천합니다: 봄 - 노란색라운드티셔츠조리센들

여름으로 판단 (23도 이상)

현재 기온을 입력해 주세요: 31

현재 날씨에 따른 옷차림(룩)을 추천합니다: 여름 - 나시반바지운동화

가을으로 판단 (5도 이상 12도 미만)

현재 기온을 입력해 주세요: 10

현재 날씨에 따른 옷차림(룩)을 추천합니다: 가을 - 롱코트슬랙스단화

위의 조건들에 해당하지 않는 경우 겨울로 판단

현재 기온을 입력해 주세요: 3

현재 날씨에 따른 옷차림(룩)을 추천합니다: 겨울 - 과잠청바지크룩스

**(6) 테스트한 기능 이름 전체 옷 리스트 출력 및 부족한 옷 확인**

- printf("%n가진 옷 리스트:%n");: 사용자가 보유한 옷 리스트를 출력하기 위해 "가진 옷 리스트"라는 메시지를 출력

printClothesForSeason(springClothes, "봄", numSpringClothes);: 봄 옷 리스트를 출력하는 함수를 호출, springClothes배열에 저장된 봄 옷의 정보를 출력

printClothesForSeason(summerClothes, "여름", numSummerClothes);: 여름 옷 리스트를 출력하는 함수를 호출, summerClothes배열에 저장된 여름 옷의 정보를 출력

printClothesForSeason(fallClothes, "가을", numFallClothes);: 가을 옷 리스트를 출력하는 함수를 호출, fallClothes배열에 저장된 가을 옷의 정보를 출력

printClothesForSeason(winterClothes, "겨울", numWinterClothes);: 겨울 옷 리스트를 출력하는 함수를 호출, winterClothes배열에 저장된 겨울 옷의 정보를 출력

recommendAdditionalClothes(numSpringClothes, numSummerClothes, numFallClothes, numWinterClothes);: 부족한 옷을 권장하는 함수를 호출. 각 계절별로 필요한 옷 개수를 파라미터로 전달

determineSeason(springClothes, numSpringClothes, summerClothes, numSummerClothes, fallClothes, numFallClothes, winterClothes, numWinterClothes);: 현재 계절을 판단하여 그에 맞는 옷차림을 추천하는 함수를 호출

- 테스트 결과 스크린샷

"가진 옷 리스트"라는 메시지를 출력

**가진 옷 리스트:**

각 계절별 필요한 옷 개수 함수 호출

```
추가적인 옷을 구매하시길 권장하는 계절 :  
- 봄 옷을 3개 더 구매하십시오.  
- 여름 옷을 2개 더 구매하십시오.  
- 가을 옷을 4개 더 구매하십시오.  
- 겨울 옷을 3개 더 구매하십시오.
```

재 계절을 판단하여 그에 맞는 옷차림을 추천하는 함수를 호출

```
현재 기온을 입력해 주세요: 23
```

```
현재 날씨에 따른 옷차림(룩)을 추천합니다: 여름 - 노란색라운드티셔츠치마쑤리
```

**(7) 테스트한 기능 이름** 옷의 개수와 옷 이름 길이 정의 및 기능1,2,3,4,5,6을 가지는 헤더 파일

- #ifndef CLOTHES\_H와 #define CLOTHES\_H는 헤더 파일의 중복 포함을 방지하기 위한 전처리기 지시자

MAX\_CLOTHES와 MAX\_LENGTH는 각각 옷의 최대 개수와 옷 이름의 최대 길이를 정의하는 상수들

typedef struct { char name[MAX\_LENGTH]; } Clothes;는 옷 정보를 담는 구조체를 정의. 이 구조체는 옷의 이름을 저장합니다.

**(7) 테스트한 기능 이름** 옷 정보를 입력받고, 각 계절에 따른 옷 리스트를 출력하며 부족한 옷을 확인하는 프로그램의 메인 파일

- Clothes\* springClothes, \* summerClothes, \* fallClothes, \* winterClothes;; 봄, 여름, 가을, 겨울 계절별 옷 정보를 저장할 포인터들을 선언

int numSpringClothes = 0, numSummerClothes = 0, numFallClothes = 0, numWinterClothes = 0;; 각 계절별 옷의 개수를 나타내는 변수들을 초기화

getClothes(&springClothes, "봄", &numSpringClothes);: 사용자로부터 봄 계절의 옷 정보를 입력받음

getClothes(&summerClothes, "여름", &numSummerClothes);, getClothes(&fallClothes, "가을", &numFallClothes);, getClothes(&winterClothes, "겨울", &numWinterClothes);: 여름, 가을, 겨울 계절의 옷 정보를 입력받음

printTotalClothes(...);: 각 계절별 옷 리스트를 출력하고 부족한 옷을 확인하는 함수를 호출

free(springClothes); free(summerClothes); free(fallClothes); free(winterClothes);: 동적으로 할당된 메모리를 해제

- 테스트 결과 스크린샷

프로그램의 전체 스크린샷

```
봄 옷차림(목록)을 입력해주세요: (최대 5가지, 띄어쓰기 사용 X)
예시) 검정맨투맨슬랙스단화
옷차림(목록) 1: 검정맨투맨반바지크룩스
더 입력하시겠습니까? (종료시: n or N/더 입력하려면 아무키나 누르시오): y
옷차림(목록) 2: 하얀반팔슬랙스단화
더 입력하시겠습니까? (종료시: n or N/더 입력하려면 아무키나 누르시오): n
여름 옷차림(목록)을 입력해주세요: (최대 5가지, 띄어쓰기 사용 X)
예시) 검정맨투맨슬랙스단화
옷차림(목록) 1: 파란색브이넥반팔린넨반바지슬리퍼
더 입력하시겠습니까? (종료시: n or N/더 입력하려면 아무키나 누르시오): o
옷차림(목록) 2: 나시형광바지운동화
더 입력하시겠습니까? (종료시: n or N/더 입력하려면 아무키나 누르시오): p
옷차림(목록) 3: 노란색라운드티셔츠치마쪼리
더 입력하시겠습니까? (종료시: n or N/더 입력하려면 아무키나 누르시오): N
가을 옷차림(목록)을 입력해주세요: (최대 5가지, 띄어쓰기 사용 X)
예시) 검정맨투맨슬랙스단화
옷차림(목록) 1: 코트청바지단화
더 입력하시겠습니까? (종료시: n or N/더 입력하려면 아무키나 누르시오): n
겨울 옷차림(목록)을 입력해주세요: (최대 5가지, 띄어쓰기 사용 X)
예시) 검정맨투맨슬랙스단화
옷차림(목록) 1: 과장진청바지농구화
더 입력하시겠습니까? (종료시: n or N/더 입력하려면 아무키나 누르시오): m
옷차림(목록) 2: 패딩런닝바지러닝화
더 입력하시겠습니까? (종료시: n or N/더 입력하려면 아무키나 누르시오): n

가진 옷 리스트:

봄 옷차림(목록) 목록:
1. 검정맨투맨반바지크룩스
2. 하얀반팔슬랙스단화

여름 옷차림(목록) 목록:
1. 파란색브이넥반팔린넨반바지슬리퍼
2. 나시형광바지운동화
3. 노란색라운드티셔츠치마쪼리

가을 옷차림(목록) 목록:
1. 코트청바지단화

겨울 옷차림(목록) 목록:
1. 과장진청바지농구화
2. 패딩런닝바지러닝화

추가적인 옷을 구매하시길 권장하는 계절:
- 봄 옷을 3개 더 구매하십시오.
- 여름 옷을 2개 더 구매하십시오.
- 가을 옷을 4개 더 구매하십시오.
- 겨울 옷을 3개 더 구매하십시오.

현재 기온을 입력해 주세요: 23

현재 날씨에 따른 옷차림(목록)을 추천합니다: 여름 - 노란색라운드티셔츠치마쪼리

C:\Users\hyeongmin\OneDrive\바탕 화면\c\x64\Debug\c.exe(프로세스 9680개)이(가) 종료되었습니다(코드: 0개).
이 창을 닫으려면 아무 키나 누르세요...
```



## 5. 계획 대비 변경 사항

### 1) 변경 내역 제목 내가 소유한 옷 수집 및 입었던 옷차림(룩)수집

- 옷장에 내가 소유한 옷(상의 하의 신발) 정보를 입력하여 저장, 최소 3일 전에 이미 입은 옷차림(룩)을 입력 받는 기능
- (상의 하의 신발) 의 형태로 입력 받는 기능이 아닌 계절별로 옷을 입력받을수 있도록 수정. 현재 기능1의 형태로 수정
- (상의 하의 신발) 다 따로 입력 받게 된다면 카테고리별로 하니까 사용자에게는 입력하기 쉽고 시간이 적게 들어 편하겠지만 내 코딩 실력으로는 카테고리별 각자 입력받은 내용들을 날씨에 맞게 랜덤으로 지정할수 없다고 판단하여 계절별 (상의하의신발)로 입력을 받아 더욱 계절을 특정하기 쉽게 수정

### 2) 변경 내역 제목 최소 3일전에 입었던 옷들 추천 데이터에서 제외

- 최소 3일전에 입었던 옷들 추천 데이터에서 제외하는 기능
- 없음
- 아직 나의 지식과 경험으로서는 구현하기 힘들다고 판단했기 때문

### 3) 변경 내역 제목 기온정보 수집 및 옷차림 추천

- 기온에 따른 오늘의 옷차림(룩)을 랜덤하게 추천
- 기온에 따라서 추천 받는 것은 같지만 기온에 따라 계절을 판별하여 그에 맞는 옷차림을 추천하는 것으로 변경. 현재 기능5의 형태로 수정
- 기온에 따라서 옷을 추천하기에 옷을 입력 받을때 이 옷이 몇 도의 기온에서 입어야 하는 옷인지까지 판별하기 위해선 직접 하나 하나 사용자가 입력하는 방법 밖에 없다고 판단. 이 방법은 좋지 못한 프로그램이라 생각하여 포괄적으로 계절을 기온에 따라 나누고 계절에 따른 옷을 받아서 추천 한다면 몇도의 기온에서 입어야 하는지 사용자가 직접 입력하지 않아도 범위가 정해져 있기 때문에 더 적합한 방법이라고 생각하게 되어 변경

#### 4) 변경 내역 제목 옷 구매 추천

- 사용자의 옷장에 부족한 카테고리의 옷을 추천하여 쇼핑 목록에 저장하는 기능
- 부족한 카테고리의 옷을 부족한 계절별 옷으로 변경. 현재 기능4의 형태로 수정
- 부족한 카테고리의 옷은 옷의 종류에 따라 사용자가 직접 입력해야하기 때문에 포괄적으로 판단할 수 있는 계절별 옷으로 판단하는게 적합하다고 생각하여 변경

### 6. 계획 대비 추가 사항

#### 1) 추가 내역 제목 추천된 옷&구매 옷 저장 기능

- (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

## 6. 느낀점

한 학기동안 이 프로젝트를 진행하면서 나를 돌아보고 나에 대한 객관화가 되는 경험이 되었다. 프로젝트를 시작하기 전에는 나에게 너무 과대평가했던거 같다. 내가 생각한대로 코드를 쉽게 짜고 계획한 기능은 충분히 만들어 낼 수 있을거라고 생각했지만 막상 시작해보니 무엇부터 해야될지 난감했고 수업에서 배운 것을 나의 창의력을 더해 새로운 기능을 만드는 것은 어려운 일이었다. 그래서 초반에 되게 많이 수정하고 계획했던 것의 스케일을 줄여보기도 하고 심지어 삭제까지 하게 되었다. 모든 것을 비우고 차근차근 하나하나 계획하고 이어나가는 것이 중요하다고 느끼게 되었고 그 후부터는 계획하는 것에 많은 시간을 쏟게 되었다. 물론 코딩 실력은 뒷받쳐주진 못했지만 그래도 이 수업을 하면서 얻은게 있다면 계획을 할때 스케일과 나에 대한 객관화가 진행된 상태에서 하는 것이 중요하다 생각하게 되었다. 물론 그를 따라주는 코딩 실력까지 있다면 정말 좋은 프로그래머가 될수 있을 거라고 생각한다. 수업을 마치며 상당히 얻은 것이 많은 수업이었다고 생각한다. 교수님도 처음 가르치셔서 학생들을 고려하지 못한것도 있지만 이해한다. 나였어도 그랬을거같다. 그만큼 항상 열정적이신것이 본 받을 점이라고 생각한다. 한 학기동안 수고 많으셨습니다. 짱