

C프로그래밍및실습

# 기온에 따른 옷차림(룩) 추천 프로그램

진척 보고서 #1

제출일자: 2023년 11월 2일

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.

### 2) 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 입력

- 입출력

입력

clothes: 옷차림(룩)을 저장하는 배열

season: 현재 입력되는 계절을 가리키는 문자열

numClothes: 해당 계절의 옷 개수

#### - 설명

사용자로부터 특정 계절의 옷 정보를 입력받는다.

사용자에게 특정 형식(띄어쓰기 없이)에 따라 옷차림(룩)을 입력받는 방법을 안내한다.

#### - 적용된 배운 내용 (반복문, 배열, 조건문, 포인터)

#### - 코드 스크린샷

```
// recommendOutfit 함수를 determineSeason 함수 위에 선언
void recommendOutfit(const char clothes[MAX_CLOTHES][MAX_LENGTH], int numClothes, const char* season);

void getClothes(char clothes[MAX_CLOTHES][MAX_LENGTH], const char* season, int* numClothes) {
    // 사용자에게 옷 입력을 요청하는 함수
    printf("%s 옷차림(룩)을 입력해주세요: (최대 %d가지, 띄어쓰기 사용 X) \n", season, MAX_CLOTHES);
    while (*numClothes < MAX_CLOTHES) {
        printf("옷차림(룩) %d: ", *numClothes + 1);
        scanf_s("%s", clothes[*numClothes], MAX_LENGTH);

        (*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 옷차림(룩)의 리스트

#### - 입출력

입력

clothes: 옷차림(룩)을 저장하는 배열

season: 출력할 계절

numClothes: 해당 계절의 옷 개수

출력

선택된 계절의 옷차림(룩) 목록을 출력함

#### - 설명

clothes[MAX\_CLOTHES][MAX\_LENGTH]는 각 계절에 해당하는 옷차림(룩)을 저장하는 이중 배열

season은 출력될 계절을 알려주는 문자열

numClothes는 해당 계절에 입력된 옷의 개수를 추적합니다.

- 적용된 배운 내용(반복문, 배열, 포인터)
- 코드 스크린샷

```
// 선택된 계절의 옷 목록을 출력하는 함수
void printClothesForSeason(const char clothes[MAX_CLOTHES][MAX_LENGTH], 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]);
    }
}
```

### (3) 구현한 기능 이름 기온을 입력받고 계절을 판별

- 입출력

입력

springClothes: 봄 옷차림(룩) 배열

numSpringClothes: 봄 옷의 개수

summerClothes: 여름 옷차림(룩) 배열

numSummerClothes: 여름 옷의 개수

fallClothes: 가을 옷차림(룩) 배열

numFallClothes: 가을 옷의 개수

winterClothes: 겨울 옷차림(룩) 배열

numWinterClothes: 겨울 옷의 개수

- 출력

해당 계절에 맞는 옷 추천

- 설명

이 함수는 현재 기온을 입력받아 계절을 판별하고 해당 계절에 맞는 옷을 추천하는 함수

emperature 변수는 사용자로부터 입력받은 현재 기온을 저장

각 기온에 따라 봄, 여름, 가을, 겨울을 판별하고, 해당 계절의 옷 추천을 위해

recommendOutfit 함수를 호출

적용된 배운 내용(조건문, 배열, 포인터)

- 코드 스크린샷

```
void determineSeason(const char springClothes[MAX_CLOTHES][MAX_LENGTH], int numSpringClothes,
const char summerClothes[MAX_CLOTHES][MAX_LENGTH], int numSummerClothes,
const char fallClothes[MAX_CLOTHES][MAX_LENGTH], int numFallClothes,
const char winterClothes[MAX_CLOTHES][MAX_LENGTH], int numWinterClothes) {
// 현재 기온을 입력받아 계절을 판별하고 해당 계절에 맞는 옷을 추천해주는 함수
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);
}
}
```

#### (4) 구현한 기능 이름 판별한 계절에 따라 옷차림(룩) 추천

- 입력

clothes: 해당 계절의 옷차림(룩)을 담은 배열

numClothes: 해당 계절의 옷 개수

season: 현재 계절을 나타내는 문자열

- 출력

선택된 계절의 랜덤 옷차림(룩)을 출력

- 설명

해당 함수는 사용자에게 추천할 옷차림(룩)을 랜덤으로 선택하여 출력

만약 해당 계절의 옷 개수가 0이라면 "추천할 수 없습니다." 메시지를 출력하고 함수를

종료

그렇지 않을 경우, rand() 함수를 사용하여 랜덤한 인덱스를 생성하고, 해당 인덱스에 해당하는 옷차림(룩)을 출력

출력되는 메시지는 "현재 날씨에 따른 옷차림(룩)을 추천합니다: [계절] - [옷차림(룩)]" 형태로 출력

적용된 배운 내용(배열)

- 코드 스크린샷

```
void recommendOutfit(const char clothes[MAX_CLOTHES][MAX_LENGTH], int numClothes, const char* season) {  
    // 해당 계절의 옷 중에서 랜덤으로 옷을 추천해주는 함수  
    if (numClothes == 0) {  
        printf("추천할 수 없습니다. %s에 대한 옷차림(룩)이 없습니다.\n", season);  
        return;  
    }  
  
    srand(time(NULL));  
    int randomIndex = rand() % numClothes;  
  
    printf("\n현재 날씨에 따른 옷차림(룩)을 추천합니다: %s - %s\n", season, clothes[randomIndex]);  
}
```

(5) 구현한 기능 이름 부족한 계절별 옷차림(룩)을 알림

- 입력

numSpringClothes: 봄 옷의 개수

numSummerClothes: 여름 옷의 개수

numFallClothes: 가을 옷의 개수

numWinterClothes: 겨울 옷의 개수

- 출력

해당 계절에 부족한 옷 개수를 권장

- 설명

해당 함수는 각 계절별로 현재 보유 중인 옷의 개수를 확인하고, 최대 옷 개수 (MAX\_CLOTHES)와 비교하여 부족한 옷 개수를 출력

개별 계절의 옷 개수가 최대 개수(MAX\_CLOTHES)보다 적을 때, 부족한 옷 개수만큼 옷을 추가로 구매하는 것을 사용자에게 권장

각 계절에 대해 필요한 옷의 부족함을 알려줌으로써 옷을 추가로 구매할 수 있는 도움을 줌

- 적용된 배운 내용(조건문, 배열)

- 코드 스크린샷

```
void recommendAdditionalClothes(const int numSpringClothes, const int numSummerClothes,
const int numFallClothes, const int numWinterClothes) {
    // 부족한 계절의 옷 개수를 알려주는 함수
    printf("\n추가적인 옷을 구매하시길 권장하는 계절:\n");

    if (numSpringClothes < MAX_CLOTHES) {
        printf("- 봄 옷을 %d개 더 구매하십시오.\n", MAX_CLOTHES - numSpringClothes);
    }
    if (numSummerClothes < MAX_CLOTHES) {
        printf("- 여름 옷을 %d개 더 구매하십시오.\n", MAX_CLOTHES - numSummerClothes);
    }
    if (numFallClothes < MAX_CLOTHES) {
        printf("- 가을 옷을 %d개 더 구매하십시오.\n", MAX_CLOTHES - numFallClothes);
    }
    if (numWinterClothes < MAX_CLOTHES) {
        printf("- 겨울 옷을 %d개 더 구매하십시오.\n", MAX_CLOTHES - numWinterClothes);
    }
}
```

(6) 구현한 기능 이름 부족한 계절별 옷차림(룩)을 알림& 현재 날씨에 맞는 옷을 추천

- 출력

각 계절에 대한 옷의 부족한 개수를 사용자에게 출력하여 권장하는 내용을 알림.

- 설명

함수가 시작되면, 모든 계절에 대한 옷 목록을 출력

각 계절별로 현재 옷의 개수와 최대 개수(MAX\_CLOTHES)를 비교하여, 부족한 개수를 출력하여 사용자에게 추가 구매를 권장

이후, recommendAdditionalClothes함수를 호출하여 각 계절에 대한 부족한 옷의 개수를 알림

determineSeason함수를 호출하여 사용자에게 현재 날씨에 맞는 옷을 추천

- 적용된 배운 내용(배열)

- 코드 스크린샷



```

void printTotalClothes(const char springClothes[MAX_CLOTHES][MAX_LENGTH], int numSpringClothes,
const char summerClothes[MAX_CLOTHES][MAX_LENGTH], int numSummerClothes,
const char fallClothes[MAX_CLOTHES][MAX_LENGTH], int numFallClothes,
const char winterClothes[MAX_CLOTHES][MAX_LENGTH], int numWinterClothes) {
    // 모든 계절의 옷 목록을 출력하고, 부족한 옷 개수를 알려주며, 해당 계절의 추천 옷을 보여주는 함수
    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 각 계절의 옷을 입력받고, 총 옷 목록을 출력하며, 추천 옷을 결정하는 메인 함수

- 코드 스크린샷

```

int main() {
    // 각 계절의 옷을 입력받고, 총 옷 목록을 출력하며, 추천 옷을 결정하는 메인 함수
    char springClothes[MAX_CLOTHES][MAX_LENGTH];
    char summerClothes[MAX_CLOTHES][MAX_LENGTH];
    char fallClothes[MAX_CLOTHES][MAX_LENGTH];
    char winterClothes[MAX_CLOTHES][MAX_LENGTH];

    int numSpringClothes = 0, numSummerClothes = 0, numFallClothes = 0, numWinterClothes = 0;

    getClothes(springClothes, "봄", &numSpringClothes);
    getClothes(summerClothes, "여름", &numSummerClothes);
    getClothes(fallClothes, "가을", &numFallClothes);
    getClothes(winterClothes, "겨울", &numWinterClothes);

    printTotalClothes(springClothes, numSpringClothes, summerClothes, numSummerClothes,
        fallClothes, numFallClothes, winterClothes, numWinterClothes);

    return 0;
}

```

## 2) 테스트 결과

(1) 테스트한 기능 이름 각 계절의 옷차림(룩)을 입력

설명

- printf 함수를 사용하여 해당 계절과 최대 입력 가능한 옷 개수에 대한 안내 메시지를 출력 (사용자에게는 띄어쓰기를 사용하지 말라는 지시를 전달)

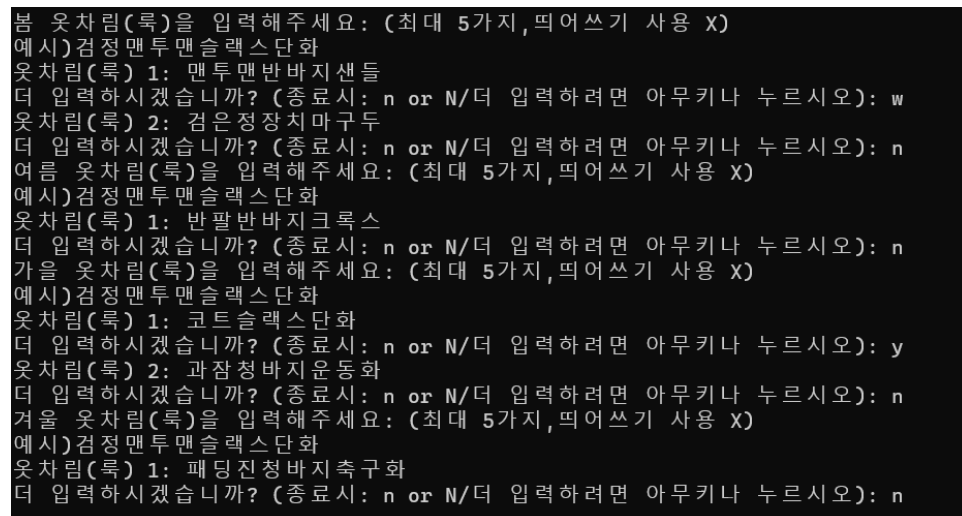
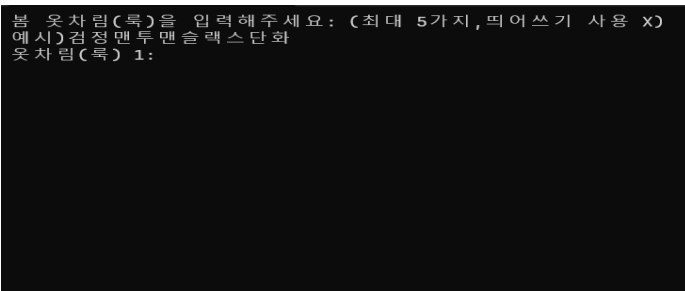
그 후 while 반복문을 이용하여 사용자가 필요한 만큼 옷차림(룩)을 입력

사용자가 입력한 옷차림(룩)은 scanf\_s 함수를 이용하여 입력받으며, 추가 입력 여부를 확인하기 위해 사용자로부터 조건문을 이용해 아무키 또는 'N'과 같은 선택지를 입력받음

'n' or 'N' 을 입력하면 반복문이 종료되고 입력을 마칩니다.

입력된 옷차림(룩)은 배열에 저장됩니다. 각 계절에 따라 해당하는 배열에 순차적으로 저장

- 스크린샷



## (2) 테스트한 기능 이름 옷차림(룩)의 리스트

설명

printf("%n%s 옷차림(룩) 목록:%n", season);: 해당 계절의 옷차림 목록을 출력하기 위해 해당 계절을 표시

for반복문: 해당 계절에 저장된 옷차림을 배열에서 하나씩 가져와 출력

for (int i = 0; i < numClothes; ++i): 배열의 크기만큼 반복하며 옷차림을 출력

printf("%d. %s\n", i + 1, clothes[i]);: 각 옷차림을 순차적으로 번호를 붙여서 출력

- 스크린샷

가진 옷 리스트:

봄 옷차림(룩) 목록:

1. 맨투맨반바지샌들
2. 검은정장치마구두

여름 옷차림(룩) 목록:

1. 반팔반바지크룩스

가을 옷차림(룩) 목록:

1. 코트슬랙스단화
2. 과잠청바지운동화

겨울 옷차림(룩) 목록:

1. 패딩진청바지축구화

## (2) 테스트한 기능 이름 기온을 입력받고 계절을 판별

설명

먼저 scanf\_s함수를 사용하여 사용자로부터 기온을 입력받음

이 입력된 기온을 기준으로 조건문을 통해 계절을 판별

입력된 기온에 따라 if-else조건문이 사용되는데, 각 조건에 맞게 계절을 설정하고 recommendOutfit함수를 호출

이때 해당 계절에 맞는 옷차림(룩) 배열과 배열의 길이를 매개변수로 전달

예를 들어, 현재 기온이 23도 이상이면 '여름'으로 설정되고, recommendOutfit함수에 여름에 해당하는 옷차림 배열과 배열의 길이가 전달

기온별로 계절이 판별되고, 해당 계절에 맞게 옷차림을 출력하는 함수인 recommendOutfit이 호출

- 스크린샷

현재 기온을 입력해 주세요: 29

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

#### (4) 테스트한 기능 이름 판별한 계절에 따라 옷차림(룩) 추천

##### - 설명

temperature변수를 이용해 현재 기온을 입력 받음

if-else if-else문을 사용하여 입력된 기온에 따라 봄, 여름, 가을, 겨울 중 어떤 계절인지를 판별

각 계절에 따라 recommendOutfit함수를 호출하여 해당 계절의 옷차림(룩) 배열과 개수를 전달

recommendOutfit함수는 해당 계절의 옷 중에서 랜덤으로 하나를 선택하여 현재 날씨에 맞는 옷을 출력

##### -스크린샷

현재 기온을 입력해 주세요: 29

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

#### (5) 테스트한 기능 이름 부족한 계절별 옷차림(룩)을 알림

##### - 설명

temperature변수를 이용하여 현재 기온을 판별하지 않고, 각 계절별로 저장된 옷의 개수를 확인하여 최대 개수인 MAX\_CLOTHES와 비교

각 계절의 옷의 부족한 개수를 확인하고, 해당하는 계절에 대해 사용자에게 얼마나 더 옷을 구매해야 하는지 알림

봄, 여름, 가을, 겨울 각각의 계절에 대해 옷의 개수가 최대값에 미치지 못하는 경우, 부족한 옷의 개수를 출력하여 추가 구매를 권장

##### - 스크린샷

추가적인 옷을 구매하시길 권장하는 계절:

- 봄 옷을 3개 더 구매하십시오.
- 여름 옷을 4개 더 구매하십시오.
- 가을 옷을 3개 더 구매하십시오.
- 겨울 옷을 4개 더 구매하십시오.

#### (6) 테스트한 기능 이름 부족한 계절별 옷차림(룩)을 알림& 현재 날씨에 맞는 옷을 추천

temperature변수 대신에, 각 계절에 저장된 옷의 개수를 확인하여 최대 개수인 MAX\_CLOTHES와 비교

각 계절별로 옷의 부족한 개수를 확인하고, 해당 계절에 대해 사용자에게 추가적인 옷을 구매해야 하는 정보를 알림

봄, 여름, 가을, 겨울 각 계절에 대해 옷의 개수가 최대값에 미치지 못하는 경우, 부족한 옷의 개수를 출력하여 추가 구매를 권장

- 스크린샷

추가적인 옷을 구매하시길 권장하는 계절:

- 봄 옷을 3개 더 구매하십시오.
- 여름 옷을 4개 더 구매하십시오.
- 가을 옷을 3개 더 구매하십시오.
- 겨울 옷을 4개 더 구매하십시오.

현재 기온을 입력해 주세요: 29

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

## 4. 계획 대비 변경 사항

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

옷장에 내가 소유한 옷(상의, 하의, 신발) 정보를 입력하여 저장, 최소 3일 전에 이미 입은 옷차림(룩)을 입력받는기능

- 상의 하의 신발 로 입력 받아서 기온에 따른 랜덤으로 추천하는게 아닌 계절별로 옷을 입력 받기

- 상의 하의 신발 다 따로받아서 랜덤으로 추천하는것 보다는 기온을 명확하게 나눌수 있게 하여 랜덤하게 추천하는것이 이 기능을 구현하기에 더 적합하다고 판단

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

최소 3일전에 입었던 옷들 추천 데이터에서 제외하는 기능

- 없앴
- 아직 나의 지식과 경험으로서는 구현하기 힘들다고 판단했기 때문이다. 추후에 다른 기능으로 추가할 예정

## **3) 변경 내역 제목** 기온정보 수집 및 옷차림 추천

- 기온에 따른 오늘의 옷차림(룩)을 랜덤하게 추천
- 기온에 따라서 추천받는것을 맞지만 기온에 따라 계절을 판별하여 그에 맞는 옷차림을 추천하는 것으로 변경
- 기온에 따라서 옷을 추천하기에 옷을 입력받을때 이 옷이 몇도의 기온에서 입어야하는 옷인지 까지 판별하기 위해선 직접 하나하나 사용자가 입력하는 방법밖에 없었다. 이 방법은 좋지 못한 프로그램이라 생각하여 포괄적으로 계절에 따른 옷을 받아서 추천한다면 몇도의 기온에 입어야하는지 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
- 부족한 카테고리의 옷은 옷의 종류에 따라 사용자가 직접 입력해야하기 때문에 포괄적으로 판단할수 있는 계절별 옷으로 판단하는게 적합하다고 생각하여 변경

## **5. 프로젝트 일정**

(진행한 작업과 진행 중인 작업 등을 표기)

업무		11-26	12-3	12-10	.....
기능1,2,3		완료			
새로운 기능 1	세부기능1		-->		
새로운 기능 2	세부기능1			--->	

## 추가 하고 싶은 내용

- 사용자가 입력한 옷을 추가할수 있도록 하거나 번호를 입력받아 예) 1. 수정, 2. 삭제 하는 기능

-