

## 04

## 프로그래밍 프로젝트

\* 게임에서의 아이템(item)을 관리하는 프로그램을 작성하여 보자. 아이템들은 리스트를 이용하여 관리하여 보자. 각각의 경기자는 현재 가지고 있는 아이템들의 리스트를 가지게 된다. 간단하게 하기 위하여 아이템 리스트에서 각각의 항목들은 현재 아이템들의 무게만을 가지게 된다.



• 아이템 리스트 = ((2kg), (5kg), (3kg))

**01** 아이템을 추가하는 함수 `add_item`를 리스트 ADT가 제공하는 연산만을 이용하여 작성하여 보자. 매개변수로 아이템 종류와 아이템의 무게가 주어진다. `add_item`는 현재 가지고 있는 아이템의 총 무게에다 현재 추가하려고 하는 아이템의 무게를 합쳐서 최대 무게가 초과되지 않는지를 검사한다. 초과되지 않으면 새로운 아이템을 아이템 리스트에 추가한다.

```
#define MAX_WEIGHT 100

// 아이템 추가 함수
void add_item(LinkedListType *list, int new_item)
{
    // 리스트 ADT가 제공하는 연산만을 이용하여 작성한다.
    int i;
    int total_weight=0;
    int length = _____ ;           // 연결 리스트의 길이 연산

    for(i=0;i<length;i++){
        total_weight += _____ ;    //각 항목 추출 연산
    }
    if( (total_weight + new_item) < MAX_WEIGHT )
        _____ ;                   // 삽입 연산
}
```

**02** 같은 방식으로 아이템 삭제 연산을 만들어 보자. 삭제할 때는 단순히 주어진 무게에 해당하는 아이템을 제거한다. 리스트 ADT에서 제공하는 연산만을 사용하라.

```
// 아이템 삭제 함수
void delete_item(LinkedListType *list, int removed_item)
```

```
{
// 리스트 ADT가 제공하는 연산만을 이용하여 작성한다.
int i;
int item;
int length = _____ ;    // 연결 리스트의 길이 연산

for(i=0;i<length;i++){
    item = _____ ;    // 각 항목 추출 연산
    if( item == removed_item ) {
        _____ ;    // 삭제연산
        return;
    }
}
}
```

**03** 위의 add\_item 함수와 delete\_item 함수를 테스트 하기 위하여 main 함수를 작성하여 보자.

```
int main()
{
    LinkedListType list;
    init(&list);
    add_item(&list, 10);
    add_item(&list, 20);
    display(&list);
    delete_item(&list, 10);
    delete_item(&list, 20);
    display(&list);
}
```

**04** 롤플레이팅 게임에서 경기자는 아이템을 가지고 다니게 되는데 이 아이템의 개수를 제한시키면 안되기 때문에 아이템을 저장하는데 배열을 사용하는 것보다는 연결 리스트를 사용하는 것이 자연스럽다. 따라서 리스트 ADT의 구현 방법으로 단순 연결 리스트를 채택하여 위의 프로그램이 수행되도록 하라. display 함수를 작성하여 맞게 구현하였는지를 검사하라. 본문의 4.4절을 참고하라.

**05** 아이템 리스트에 아이템의 무게만 저장하지 말고 아이템의 타입도 같이 저장하게끔 프로그램을 변경하라. 즉 element 타입을 다음과 같은 구조체로 정의하여 동일한 프로그램을 작성하여 보라.

```
typedef struct {
    int type;    // 아이템 타입
    int weight; // 아이템 무게
} element;
```

아이템 리스트 = ( (type=3, weight=2kg), (type=1, weight=5kg) )

**06** 아이템 리스트를 리스트 ADT에 구애받지 말고 단순 연결 리스트를 이용하여 최대한 효율적으로 작성하여 보라. 즉 아이템 리스트를 단순 연결 리스트로 구현하고 항목들을 탐색할 때 포인터를 이용하여 탐색하여 보라.

- 07 아이템 리스트를 단순 연결 리스트 대신에 이중 연결 리스트를 이용하여 구현하여 보라.
- 08 아이템 리스트에서 아이템이 무게별로 오름차순으로 정렬되어 있도록 구현하여 보라.