C언어 교육 5주차 과제들

#1. 반환하지만, 반환하지 않는 함수

정수 x, y를 차례대로 입력받은 후, x와 y의 합과 차를 차례로 출력하는 프로그램을 만들자.

덧셈과 뺄셈을 수행하는 함수를 만들어야 한다. 함수는 main함수 외에 단 한 개만 만들 수 있으며, 반환형은 void이어야 한다. 또한 해당 함수에는 printf 등의 출력 함수가 포함되어서는 안 된다.

#2. 라이엇 팀장 준서

평소 게임을 좋아하는 준서는 자신만의 게임 컨트롤러를 개발하려한다.

각 구조체의 멤버변수는 변수 = <초기값>의 형태로 주어져 있다. 이 조건은 반드시 지켜야 한다.

게임 컨트롤러는 다음의 정보를 가지고 있다.

현재 배터리 퍼센트 = <사용자 입력>(0 ~ 100까지의 정수),

UP 버튼, DOWN 버튼, LEFT 버튼, RIGHT 버튼,

A 버튼, B 버튼

연결된 플레이어

위의 정보를 담은 구조체를 선언하라. 플레이어와 버튼 또한 구조체로 표현한다.

버튼은 다음의 정보를 가지고 있다.

눌린 상태 = <0>(0~1 의 정수),

총 눌린 횟수 = <0>(0부터 시작하는 양수)

플레이어는 다음의 정보를 가지고 있다. 플레이어 구조체의 각 멤버변수 이름은 무조건 괄호에 명시된 변수 이름으로 정한다.

ex) 플레이어의 x좌표를 나타내는 변수의 이름은 반드시 x로 한다.

플레이어의 이름(name) = <사용자 입력> (100byte의 문자열)

플레이어의 x 좌표(x) = <0>, y좌표 = <0> (각각 정수)

플레이어의 HP(hp) = <100>, MP(mp) = <100> (각각 정수)

각 구조체 멤버 변수의 초기값은 위 조건의 명시된대로 init~ 함수들을 구현하며, main 함수에서 init~ 함수들을 실행하여 각각의 구조체를 초기화 해준다.

다만, 몇 가지 정보들은 입력을 받는데,

플레이어의 이름, 배터리의 용량을 입력받는다.

그 후 U, D, L, R, A, B, X의 문자들이 연속되어 입력될 것이다.

U를 입력받으면 UP버튼을, D를 입력받으면 DOWN을, L을 입력받으면 LEFT버튼을 R을 입력받으면 RIGHT 버튼을, A를 입력받으면 A버튼을, B를 입력받으면 B 버튼을 누르는 기능을 구현하라. X를 입력받으면 모든 입력이 종료되었다는 의미이다.

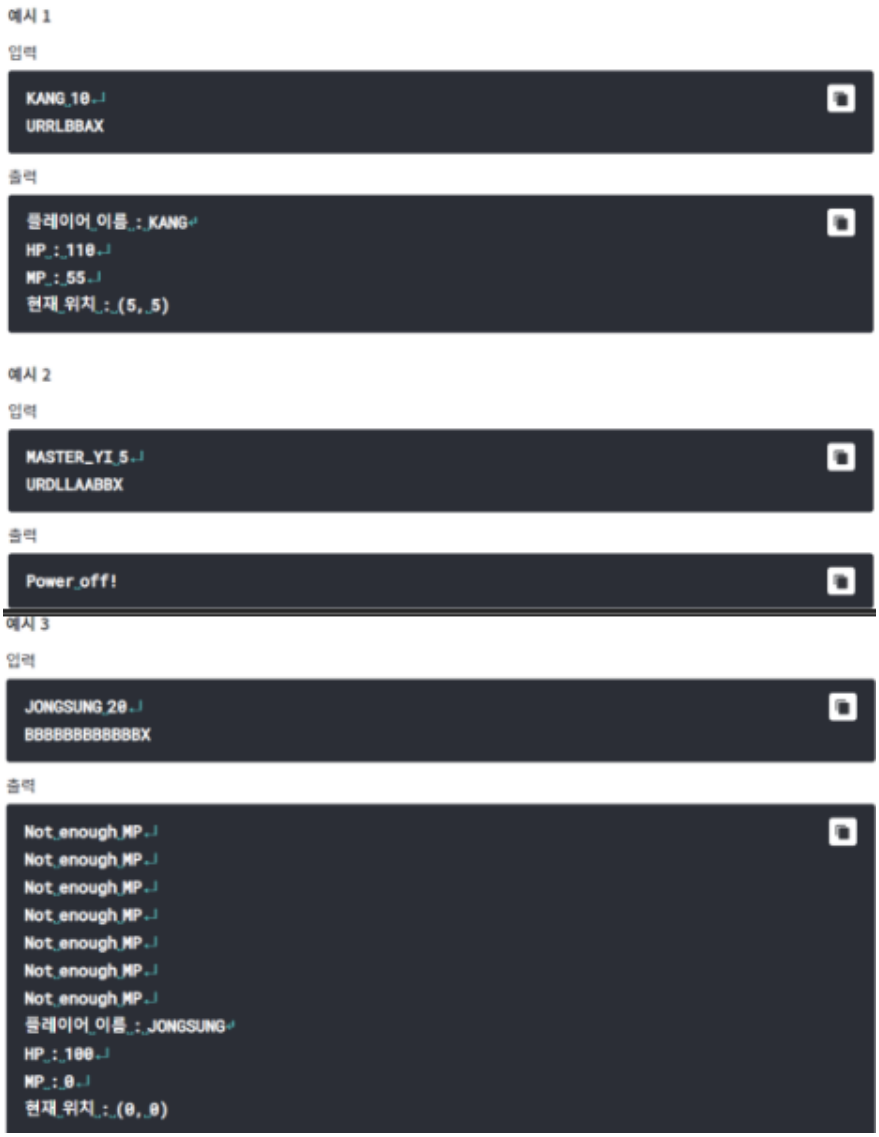
입력을 전부 받아서 모든 명령을 수행하면 plaerStatux 함수를 호출하고, 프로그램을 종료한다. 만약 도중에 배터리가 모자라서 수행하지 못하는 경우에는 “Power off!”라는 문자열을 출력하며 프로그램을 종료한다.

아래의 코드는 전체 코드의 맥락이다. 아래 코드 위에서 문제를 해결하면 된다.

```
#include <stdio.h>
#include <stdlib.h>
//헤더는 자유롭게 include 가능.
struct button{
};//위의 구조체를 문제 조건에 따라 구현하라.
struct player{
};//위의 구조체를 문제 조건에 따라 구현하라.
struct controller{
};//위의 구조체를 문제 조건에 따라 구현하라.
void playerStatus(struct player * p){
printf("플레이어 이름 : %s\n", p->name);
printf("HP : %d\n", p->hp);
printf("MP : %d\n", p->mp);
printf("현재 위치 : (%d, %d)\n", p->x, p->y);
}//위의 함수는 건드리지 않는다.
void pressButton(struct button * btn){
// button 을 누른다.
}
void moveX(struct player * p, int x){
// 플레이어의 x 좌표를 인자 x 만큼 더한다.
}
void moveY(struct player * p, int y){
// 플레이어의 y 좌표를 인자 y 만큼 더한다.
}
void heal(struct player * p){
// 플레이어의 MP 를 5 만큼 줄이고, HP 를 10 늘린다. MP 가 5 보다 작으면 시전 불가.
// 시전 불가시 "Not enough MP" 라는 문자열을 출력한다.
}
void fireball(struct player * p){
// 플레이어의 MP 를 20 만큼 줄인다. MP 가 20 보다 작으면 시전 불가.
// 시전 불가시 "Not enough MP" 라는 문자열을 출력한다.
}
```

```
void pressUp(struct controller * con){
// con 에 등록된 player 의 y 좌표를 5 만큼 증가시킨다.
// con 의 배터리 퍼센트가 1 감소한다.
// con 의 배터리가 0 일 경우, "Power off!"라는 문자열을 출력하며 프로그램을 종료한다.(exit 사용)
}
void pressDown(struct controller * con){
// con 에 등록된 player 의 y 좌표를 5 만큼 감소시킨다.
// con 의 배터리 퍼센트가 1 감소한다.
// con 의 배터리가 0 일 경우, "Power off!"라는 문자열을 출력하며 프로그램을 종료한다.(exit 사용)
}
void pressLeft(struct controller * con){
// con 에 등록된 player 의 x 좌표를 5 만큼 감소시킨다.
// con 의 배터리 퍼센트가 1 감소한다.
// con 의 배터리가 0 일 경우, "Power off!"라는 문자열을 출력하며 프로그램을 종료한다.(exit 사용)
}
void pressRight(struct controller * con){
// con 에 등록된 player 의 x 좌표를 5 만큼 증가시킨다.
// con 의 배터리 퍼센트가 1 감소한다.
// con 의 배터리가 0 일 경우, "Power off!"라는 문자열을 출력하며 프로그램을 종료한다.(exit 사용)
}
void pressA(struct controller * con){
// con 에 등록된 player 가 heal 을 시전하도록 한다.
// con 의 배터리 퍼센트가 1 감소한다.
// con 의 배터리가 0 일 경우, "Power off!"라는 문자열을 출력하며 프로그램을 종료한다.(exit 사용)
}
void pressB(struct controller * con){
// con 에 등록된 player 가 fireball 을 시전하도록 한다.
// con 의 배터리 퍼센트가 1 감소한다.
// con 의 배터리가 0 일 경우, "Power off!"라는 문자열을 출력하며 프로그램을 종료한다.(exit 사용)
}
//위 함수들을 모두 구현하여, 문제의 지시대로 main 함수를 구현하라.

void initButton(struct button * btn){
//button 의 멤버변수를 문제의 조건대로 초기화한다.
}
void initPlayer(struct player * p){
//Player 의 멤버변수를 문제의 조건대로 초기화한다.
}
void initCon(struct controller * con){
//controller 의 멤버변수를 문제의 조건대로 초기화한다.
}
int main() {
return 0;
}
```



#3. 혜진이는 생명과학이 너무 좋아

혜진이는 프로그래밍과 생명과학을 모두 좋아한다.

그래서 정말 쓸모없는 DNA to mRNA 자동화 프로그램을 만들기로 했다. 또한, 혜진이는 자기 프로그래밍 실력을 믿기 때문에 result 함수에서는 인덱스 연산자를 사용하지 않기로 했다.

숫자를 배열에 받아 size가 10인 이차원 DNA 염기서열 5개를 생성한다.

숫자와 염기쌍의 관계는 다음과 같다. (0 = ‘A’, 1 = ‘T’, 2 = ‘G’, 3 = ‘C’)

해당 DNA 염기서열은 함수에서 DNA → mRNA 과정을 거치게 된다.

총 과정은 다음과 같다.

stage #0. 숫자 배열 → 염기서열

stage #1. ‘A’ → ‘T’, ‘T’ → ‘A’, ‘G’ → ‘C’, ‘C’ → ‘G’

stage #2. ‘A’ → ‘A’, ‘T’ → ‘U’, ‘G’ → ‘G’, ‘C’ → ‘C’

함수에서 배열을 리턴하여 main의 rna 이차원 배열에 넣고 stage #0 ~ #2 까지의 과정을 총 5번 반복해서 출력한다.

(단, 인덱스 연산자 사용 시 0점 처리한다.)

```
#include <stdio.h>
???? result(???? sequence, ???? nucleic) { //void 사용 불가, 함수 레퍼런스 수정 불가, [](인덱스)사용 불가
}
int main() {
int nucleic[10]; //숫자 받을 배열
char sequence[5][11]; //숫자 to 염기서열
char rna[5][11]; //DNA to mRNA
return 0;
}
```