

프로그래밍기초 실습 과제 #06

소스 파일 상단에 학번, 이름을 주석으로 추가하세요.

- 주석이 없으면, -1점

실행 결과를 하드 코딩하면 0점 처리함

반드시 토끼와 거북이의 이동 및 현재 위치 출력은 함수로 구현하세요.

1. 토끼와 거북이 경주 게임 (20점)

- 제출 파일: hw06_race.c

토끼와 거북이가 50m 경주를 하는데, 1~10 사이의 숫자를 랜덤하게 생성하여 나온 숫자에 따라 아래 표와 같이 이동을 하는 게임을 구현하세요.

■ 이동 거리 테이블

| 거북이(Tortoise) 이동 | | 토끼(Hare) 이동 | |
|------------------|--------------|-------------|--------------|
| 랜덤 숫자 | 이동 거리(meter) | 랜덤 숫자 | 이동 거리(meter) |
| 1~5 | +3 | 1~2 | 0 |
| 6~7 | -6 | 3~4 | +9 |
| 8~10 | +1 | 5 | -12 |
| | | 6~8 | +1 |
| | | 9~10 | -2 |

■ 경기 진행 (4점)

- ✓ 토끼와 거북이는 초기 0m 위치에서 경주를 시작하여 둘 중 이동 거리가 50m 이상인 경우, 경주는 끝나며 이동 거리가 50m 이상이면 화면상 표시는 50m로 출력함 (1점)
- ✓ 토끼와 거북이는 1 ~ 10 사이의 값을 랜덤하게 생성 후 이동
 - `srand(time(NULL))` 및 `rand()`함수 사용
 - 이동 후 300msec 시간 지연을 설정함: `usleep(300 * 1000);`
- ✓ 거북이나 토끼의 현재 위치가 0보다 작은 경우, 0부터 시작 (1점)
- ✓ 경주 종료 후 승패 출력: 무승부인 경우 “Even Game!” 출력 (2점)
 - ex: “Tortoise Win!(Tortoise: 50, Hare: 20)” 형태로 이동거리도 같이 출력
 - 무승부의 경우도 이동 거리 출력함: “Even Game!(Tortoise:50, Hare: 50)” 형태

■ 거북이 이동 및 현재 위치 출력 함수 구현 (8점, 각 4점)

- ✓ 랜덤 값(1~10) 생성 후 거북이 이동 거리 테이블에 정의된 거리만큼 현재 위치 이동 및 50m 이상이면 50m에서 멈춤
- ✓ 이동 거리 출력: 누적된 거리(`pos`:현재 위치) 및 랜덤 숫자가 의미하는 이동 거리(`step`: 이동 거리)를 현재 위치(빈 공백 출력)에 출력

- ex: T(pos:10, step:3)
- ✓ 함수로 구현하지 않으면 해당 점수 없음

■ 토끼 이동 및 현재 위치 출력 함수 구현 (8점, 각 4점)

- ✓ 랜덤 값(1~10) 생성 후 토끼 이동 거리 테이블에 정의된 거리만큼 현재 위치 이동 및 50m 이상이면 50m에서 멈춤
- ✓ 이동 거리 출력: 누적된 거리(pos:현재 위치) 및 랜덤 숫자가 의미하는 이동 거리(step: 이동 거리)를 현재 위치(빈 공백 출력)에 출력
 - ex: H(pos:1, step:1)
- ✓ 함수로 구현하지 않으면 해당 점수 없음

■ usleep 함수 사용

GCC 사용(Windows, Mac 동일)

```
#include <stdio.h>
#include <time.h>
#include <stdlib.h>
#include <unistd.h>

. . .

usleep(300 * 1000); //300 msec 지연
. . .
```

■ “Type Enter to Start Race !!” 구현 코드

```
printf("T(%d)\n", tortoise_pos);
printf("H(%d)\n", hare_pos);
printf("Type Enter to Start Race !!");
scanf("%c", &ch); // 단순히 Enter키를 입력할 때까지 대기하기 위해 사용
```

- 실행 결과: `srand(time(NULL))` 호출로 실행 결과는 매번 다름

```
T(0)
H(0)
Type Enter to Start Race !!

-----
T(pos:3, step:3)
H(pos:9, step:9)

-----
T(pos:6, step:3)
H(pos:9, step:0)

-----
T(pos:7, step:1)
H(pos:18, step:9)

-----
T(pos:8, step:1)
H(pos:16, step:-2)

-----
T(pos:11, step:3)
H(pos:14, step:-2)

-----
T(pos:14, step:3)
H(pos:23, step:9)

-----
T(pos:17, step:3)
H(pos:21, step:-2)

-----
T(pos:20, step:3)
H(pos:30, step:9)

-----
T(pos:21, step:1)
H(pos:18, step:-12)

-----
T(pos:24, step:3)
H(pos:27, step:9)

-----
T(pos:27, step:3)
H(pos:28, step:1)

-----
T(pos:21, step:-6)
H(pos:16, step:-12)

-----
T(pos:24, step:3)
H(pos:4, step:-12)

-----
T(pos:25, step:1)
H(pos:5, step:1)
```

T(pos:28, step:3)
H(pos:14, step:9)

T(pos:22, step:-6)
H(pos:23, step:9)

T(pos:23, step:1)
H(pos:32, step:9)

T(pos:26, step:3)
H(pos:30, step:-2)

T(pos:27, step:1)
H(pos:31, step:1)

T(pos:30, step:3)
H(pos:32, step:1)

T(pos:31, step:1)
H(pos:33, step:1)

T(pos:25, step:-6)
H(pos:42, step:9)

T(pos:28, step:3)
H(pos:50, step:9)

Hare Win!(Tortoise: 28, Hare: 50)
\$