Virtual Pet System (struct + method concept)

Description

You are asked to design a simple virtual pet system.
Each pet has the following attributes:

- name: the pet's name (no spaces)
- happiness: an integer between 0 and 100
- energy: an integer between 0 and 100

In addition to these data fields, each pet must also support two actions.
These actions must be implemented using function pointers inside the struct, simulating the idea of "methods" in object-oriented programming.

Action 1: play

- happiness increases by 10
- energy decreases by 5
- values must remain within the range 0–100

Action 2: rest

- energy increases by 10
- happiness decreases by 2
- values must remain within the range 0–100

You must use the following struct (fields may not be removed):

```
struct Pet {
    char name[50];
    int happiness;
    int energy;
    void (*play)(struct Pet* p);
    void (*rest)(struct Pet* p);
};
```

After reading each pet's data, you must assign the play and rest function pointers to your implemented functions.

Input Format

The first line contains an integer N.

Each of the next N lines contains:

```
name happiness energy action
```

where action is either:

- play
- rest

Output Format

After performing the specified action on each pet, print:

```
name happiness energy
```

Ex. Input

```
3
Mimi  60 70  play
Kuro  80 40  rest
Pipi  50 50  play
```

Ex. Output

```
Mimi 70 65
Kuro 78 50
Pipi 60 45
```

虛擬寵物系統（struct + 方法概念）

題目敘述

請你設計一個簡單的虛擬寵物系統。
每一隻寵物（Pet）包含以下屬性：

- name：寵物名字（不含空白）
- happiness：快樂值（範圍 0–100）
- energy：能量值（範圍 0–100）

除了資料欄位外，每隻寵物必須具備兩個行為，並使用 **struct 內的函式指標（function pointer）** 來實作，模擬物件導向語言中的「方法」。

行為 1：play（玩耍）

- happiness 增加 10
- energy 減少 5
- 最終數值需限制在 0–100 之間

行為 2：rest（休息）

- energy 增加 10
- happiness 減少 2
- 最終數值需限制在 0–100 之間

請使用以下 struct（不可刪除欄位）：

```
struct Pet {
    char name[50];
    int happiness;
    int energy;
    void (*play)(struct Pet* p);
    void (*rest)(struct Pet* p);
};
```

每隻寵物在讀入後，必須將 play 與 rest 綁定到你自行實作的函式，並依據輸入的 action 執行對應行為。

輸入格式

第一行為一個整數 N，代表寵物的數量。

接下來 N 行，每行格式為：

```
name happiness energy action
```

其中 action 為：

- play
- rest

輸出格式

每隻寵物執行完指定行為後，輸出：

```
name happiness energy
```

一行代表一隻寵物。

範例一

輸入

```
3
Mimi 60 70 play
Kuro 80 40 rest
Pipi 50 50 play
```

輸出

```
Mimi 70 65
Kuro 78 50
Pipi 60 45
```