## const int *pa = &a;

### 含义：不能换水果

- pa 是一个“指向 int 的指针”，但这个 int 是 const 的。
- 你可以让 pa 指向别的 int（换房间），但不能通过 pa 改变指向的内容（换水果）。

```
const int a = 10;
const int *pa = &a;
*pa = 20; // ❌报错，不能通过 pa 修改 a
pa = &b;  // ✅可以让 pa 指向另一个地址
```

### 🍍比喻解释：

你有一个指针 pa，它是一个门牌号（地址），指向 "艾米丽大道01号房间"。房间里有一个苹果。 - 你可以把 pa 换成别的地址，比如指向 "皮埃尔大道02号房间"，但你不能走进房间换掉里面的苹果。

---

## int *const pa = &a;

### 含义：不能换房间

- pa 是一个“不能换地址的指针”。
- 你可以通过 pa 修改指向的值（换水果），但不能让 pa 指向别的地址。

```
int a = 10;
int *const pa = &a;
*pa = 20; // ✅可以修改 a 的值
pa = &b;  // ❌报错，不能改 pa 的指向
```

### 🍍比喻解释：

你有一个指针 pa，它固定指向 "艾米丽大道01号房间"，但你可以进去把苹果换成菠萝。 - 房间地址不能换（不能改 pa） - 房间里的水果可以换（可以改 *pa）

---

## const int *const pa = &a;

### 含义：不能换水果，也不能换房间

- pa 是一个指向 const int 的 const 指针
- 地址不能变，内容也不能改

```
const int a = 10;
const int *const pa = &a;
```

```
*pa = 20; // ❌报错，不能改内容
pa = &b;  // ❌报错，不能改地址
```

🍍**比喻解释：**

你有一个门牌号 pa，指向一个房间： - 房间的位置不能换（不能改地址） - 房间里的水果也不能换（不能改值）

---

## 我的比喻代表的意义

- pa：房间的名字，是一个变量，里面存着地址（也就是另一个房间的门牌号）
- &a / &b：某条大道上的具体地址，比如 "艾米丽大道01号房间"，是变量 a 或 b 的地址
- *pa：通过地址取到的房间里水果的内容
- const 修饰 *pa：表示房间里的水果不能换（不能修改指向的值）
- const 修饰 pa：表示房间不能换位置（不能修改地址）
- 换水果：改 *pa 的值（改地址里的值）
- 换房间：改 pa（换地址）

---

## 额外说明：`*` 是什么意思？

在 `int *const pa = &a;` 或 `const int *pa = &a;` 这样的语句中，`*` 的作用是：

> `*` 代表“这个变量是个指针”，pa 存的内容是某个 int 变量的地址。

☸️也就是说： - `pa` 是放地址的房间名 - `*pa` 是通过地址取到的值（水果） - `const` 可以修饰 `pa`（不能换地址），也可以修饰 `*pa`（不能换值）

| 声明方式 | 是否是指针（需要 `*`） | const 修饰谁 | 含义说明 |
| --- | --- | --- | --- |
| `int pa = 5;` | ❌否 | 无 | pa 是个普通 int 变量 |
| `int *pa = &a;` | ✅是 | 无 | pa 是一个可以换房间、换水果的指针 |
| `int *const pa = &a;` | ✅是 | 修饰 `pa` | pa 是不能换地址的指针（房间不能换） |
| `const int *pa = &a;` | ✅是 | 修饰 `*pa` | pa 是不能换水果的指针（内容不能换） |

---

可以随时拿这个对照理解不同组合，掌握指针和 const 的核心语法！