



# 指针基础概念练习题

## 一、选择题（每题5分，共20分）

1. 以下关于指针的描述，正确的是？

- A. 指针是存储变量地址的变量
- B. 指针可以直接存储整数值
- C. 指针的大小与它所指向的数据类型无关
- D. 指针变量名前面需要加 \* 声明

答案：A

解析：指针存储内存地址，声明时用 \*，但变量名本身不加 \*。指针大小通常固定（如4/8字节），与类型无关，但选项C表述不严谨（应为"sizeof(指针)"与类型无关）。

---

2. 若有 `int a = 10; int *p = &a;`，则 \*p 的值是？

- A. a 的地址
- B. 10
- C. p 的地址
- D. 随机值

答案：B

解析：\*p 表示解引用，获取 p 指向的值（即 a 的值 10）。

---

3. 以下代码的输出是？

```
int x = 5, y = 10;
int *ptr = &x;
*ptr = y;
printf("%d", x);
```

- A. 5
- B. 10
- C. x 的地址
- D. 编译错误

答案：B

解析: ptr 指向 x , \*ptr = y 修改了 x 的值为 10 。

---

4. 关于 NULL 指针, 错误的是?

- A. NULL 指针不指向任何有效内存
- B. 解引用 NULL 指针会导致运行时错误
- C. NULL 与 0 等价
- D. malloc() 失败时返回 NULL

答案: C

解析: NULL 在C中通常是 (void\*)0 , 与整数 0 不等价 (但比较时行为相同) 。

---

## 二、填空题 (每空5分, 共30分)

1. 补全代码, 通过指针交换两个变量的值:

```
void swap(int *a, int *b) {  
    int temp = _____; // 填空1  
    _____ = *b;      // 填空2  
    *b = temp;  
}
```

答案:

- 填空1: \*a
  - 填空2: \*a
- 

2. 补全指针声明:

```
int x = 100;  
_____ ptr = &x; // 声明指向x的指针  
printf("%d", _____); // 输出x的值
```

答案:

- 第一空: int \*

- 第二空: `*ptr`

---

3. 以下代码的输出是:

```
char str[] = "Hello";  
char *p = str;  
printf("%c", *(p + 2));
```

答案: `l`

解析: `p + 2` 指向第3个字符 'l'。

---

## 三、编程题 (50分)

### 1. 指针基础操作 (20分)

- 声明一个整数变量 `num` 并赋值为 `50`
- 声明指针 `p` 指向 `num`
- 通过指针修改 `num` 的值为 `100`
- 输出 `num` 的值和地址

参考代码:

```
#include <stdio.h>  
int main() {  
    int num = 50;  
    int *p = &num; // 定义一个指针变量, 指针指向num  
    *p = 100; // 对p指向的变量赋值  
    printf("值: %d, 地址: %p", num, &num);  
    return 0;  
}
```

---

### 2. 指针与数组 (30分)

- 定义一个包含 `{3, 7, 2, 9}` 的整型数组
- 使用指针遍历数组并计算元素总和

- 输出总和及最大值的地址

参考代码：

```
#include <stdio.h>

int main() {
    int arr[] = {3, 7, 2, 9};
    int *ptr = arr, sum = 0, max = *ptr;
    int *max_ptr = ptr;

    for (int i = 0; i < 4; i++) {
        sum += *(ptr + i); /*(ptr+i)<=>arr[i]
        if (*(ptr + i) > max) {
            max = *(ptr + i);
            max_ptr = ptr + i;
        }
    }

    printf("总和: %d\n最大值地址: %p", sum, (void*)max_ptr);
    return 0;
}
```

## 答案及解析

### 选择题解析

1. **A**：指针的本质是存储地址的变量。
2. **B**：`*p` 解引用获取值。
3. **B**：通过指针修改了 `x` 的值。
4. **C**：`NULL` 是空指针常量，与整数 `0` 类型不同但值相同。

### 编程题要点

- **指针声明**：类型 `*` 变量名（如 `int *p`）
- **解引用**：用 `*` 访问指针指向的值
- **指针运算**：`ptr + i` 等价于 `&arr[i]`
- **地址输出**：使用 `%p` 和 `(void*)` 强制转换