期中复习

1. 2017 期中第 4 题

1.1. 背景：字符串和字符串数组

在C语言中，字符串可以通过以下几种方式表示：

1. **字符数组**：字符串可以被存储在字符数组中。每个字符占据数组中的一个元素，字符串的末尾通常使用空字符（`'\0'`）来标识。

```c
char str[] = "Hello, World!";
```

2. **字符指针**：字符串也可以通过字符指针来表示，指针指向字符串的第一个字符。

```c
char *str = "Hello, World!";
```

上面这两种方案的本质都是单一字符储存在数组的一个单元格中。

而如果想要实现字符串数组（也就是单一字符串存在数组的单元格中），可以使用如下方案：

```c
char *arr[] = {"Hello","World"}
```

通过前面的理解可以很清晰的看到，char *arr[ ] 本质就是一个 arr[ ]，里面储存了字符串 char *

1.2. 解题

```c
char *args[] = {"hello", "world"};
void *v = args;
printf("%s %s\n", *(char **)v, *((char **)v + 1));
```

1. `char *args[] = {"hello", "world"};` 这行代码定义了一个字符指针数组 `args`，包含两个元素，分别指向字符串 `"hello"` 和 `"world"`。
2. `void *v = args;` 这行代码将 `args` 数组的地址赋值给一个 `void` 类型的指针 `v`。在C语言中，`void*` 是一个通用指针类型，可以指向任何类型的数据（C语言允许将任何类型的指针赋值给(void *)变量，所以这行代码本身无意义）。
3. `printf("%s %s\n", *(char **)v, *((char **)v + 1));` 这行代码使用 `printf` 函数来打印两个字符串。这里有几个关键点：
    - `*(char **)v`：首先，`v` 是一个 `void*` 类型的指针，通过强制类型转换 `(char **)` 将其转换为 `字符串数组` 类型的指针（char *arr[ ] 本质就是 char **，一个*对应数组，一个*对应字符串指针）。然后，通过解引用操作符 `*` 获取它指向的值，即 `args` 数组的第一个元素，也就是指向 `"hello"` 的指针。
    - `*((char **)v + 1)`：这部分首先将 `v` 转换为 `char*` 类型的指针，然后通过加1操作移动到数组的下一个元素，即指向 `"world"` 的指针。再次解引用获取指向的字符串。