



C语言字符串与二维数组基础应用作业

一、选择题（每题5分，共20分）

1. 以下关于字符串初始化的语句，正确的是？

- A. `char str[5] = "hello";`
- B. `char str[] = {'h','e','l','l','o'};`
- C. `char str[5]; str = "hello";`
- D. `char *str; str = "hello";`

答案：D

解析：

- A错误：长度不足（需6字节存储"hello\0"）。
- B错误：不是字符串（缺少结束符'\0'）。
- C错误：数组名不能直接赋值。

2. 若有定义 `char s[] = "abc";`，则 `sizeof(s)` 的值是？

- A. 3
- B. 4
- C. 编译错误
- D. 取决于编译器

答案：B

解析：包含隐式结束符'\0'，共4字节。

3. 以下代码的输出是？

```
int arr[2][3] = {{1,2},{3}};
printf("%d", arr[1][2]);
```

- A. 0
- B. 3
- C. 随机值
- D. 编译错误

答案：A

解析：未显式初始化的元素自动补0。

4. 二维数组 `int a[3][4]` 的元素 `a[2][1]` 的地址计算公式是？

- A. `&a[0][0] + 2*4 + 1`

B. `a + 2*4 + 1`

C. `&a[0] + 2*4 + 1`

D. `a[0] + 2*4 + 1`

答案：A

解析：行优先存储，偏移量为 `行号*列数 + 列号`。

二、程序填空题（每空5分，共30分）

1. 字符串反转

```
#include <stdio.h>
#include <string.h>
void reverse(char s[]) {
    int len = _____;           // 填空1: 获取字符串长度
    for (int i = 0; i < _____; i++) { // 填空2: 循环条件
        char temp = s[i];
        s[i] = s[_____];           // 填空3: 对称位置索引
        s[len-1-i] = temp;
    }
}
int main() {
    char str[] = "hello";
    reverse(str);
    printf("%s", str); // 输出: olleh
    return 0;
}
```

答案：

- 填空1: `strlen(s)`
- 填空2: `len/2`
- 填空3: `len-1-i`

2. 二维数组求最大值

```

#include <stdio.h>
int main() {
    int arr[2][3] = {{5, 9, 2}, {7, 1, 4}};
    int max = _____; // 填空1: 初始化max
    for (int i = 0; i < 2; i++) {
        for (int j = 0; j < _____; j++) { // 填空2: 内层循环条件
            if (_____ > max) { // 填空3: 比较条件
                max = arr[i][j];
            }
        }
    }
    printf("最大值: %d", max); // 输出: 9
    return 0;
}

```

答案:

- 填空1: `arr[0][0]`
- 填空2: `3`
- 填空3: `arr[i][j]`

三、编程题 (50分)

1. 字符串连接函数 (20分)

- 编写函数 `void my_strcat(char dest[], char src[])` , 实现将 `src` 字符串连接到 `dest` 末尾。
- 要求: 不能使用 `<string.h>` 中的库函数。
- 示例:

```

char s1[20] = "hello";
char s2[] = "world";
my_strcat(s1, s2); // s1变为"helloworld"

```

参考代码:

```
void my_strcat(char dest[], char src[]) {
    int i = 0, j = 0;
    while (dest[i] != '\0') i++; // 找到dest末尾
    while (src[j] != '\0') {
        dest[i++] = src[j++];    // 逐个复制字符
    }
    dest[i] = '\0';             // 添加结束符
}
```

2. 矩阵转置 (30分)

- 编写程序，将二维数组 `a[3][3]` 转置（行列互换），结果存入数组 `b[3][3]` 并输出。
- 示例输入：

```
int a[3][3] = {{1,2,3}, {4,5,6}, {7,8,9}};
```

- 示例输出：

```
1 4 7
2 5 8
3 6 9
```

参考代码：

```
#include <stdio.h>
int main() {
    int a[3][3] = {{1,2,3}, {4,5,6}, {7,8,9}};
    int b[3][3];
    for (int i = 0; i < 3; i++) {
        for (int j = 0; j < 3; j++) {
            b[j][i] = a[i][j]; // 行列互换
        }
    }
    // 输出结果
    for (int i = 0; i < 3; i++) {
        for (int j = 0; j < 3; j++) {
            printf("%d ", b[i][j]);
        }
        printf("\n");
    }
    return 0;
}
```
