2025 秋计算思维与实践期末考试（回忆版）

司翊, Fireale, 25 级学术群

一、选择题

1. 下列选项中，不属于冯 · 诺依曼计算机体系结构核心五大组成部分的是（　）。

   (a) 运算器

   (b) 控制器

   (c) 存储器

   (d) 系统总线

2. 此程序的运行结果是（　）。

```c
#include <stdio.h>
int main() {
    int i = 1;
    int j = i++;
    printf("%d%d", i, j);
    return 0;
}
```

   (a) 11

   (b) 12

   (c) 21

   (d) 22

3. 以下程序的输出结果是（　）。

```c
#include <stdio.h>

void fun() {
    static int i = 0;
    int j = 0;
    i++;
    j++;
    printf("%d%d", i, j);
}

int main() {
    fun();
    fun();
```

```
14      return 0;
15  }
```

(a) 0101

(b) 1112

(c) 1121

(d) 1221

4. 十进制数 292 对应的八进制数是（　）。

(a) 524

(b) 124

(c) 444

(d) 3A4

5. 以下 C 语言程序中，`fun(1)` 的输出结果是（　）。

```c
#include <stdio.h>

void fun(int n) {
    if (n < 5) {
        printf("%d", n);
        fun(n + 2);
        printf("%d", n);
    }
}
```

(a) 131

(b) 1331

(c) 121

(d) 1221

6. 已知 `int arr[2][5] = {{1,2,3,4,5}, {11,22,33,44,55}};`，下列选项中，**不能**正确取到数组中元素 33 的是（　）。

(a) `*(*(arr + 1) + 2)`

(b) `*(arr[1] + 2)`

(c) `*(&arr[0][0] + 1 * 5 + 2)`

(d) `*(arr + 1) + 2`

7. 数据结构需要支持在头部和尾部进行插入与删除操作，且这些操作的严格时间复杂度为 $O(1)$。下列哪种数据结构最符合此要求？（　）

(a) 带头尾指针的双向链表

(b) 带尾指针的循环单链表

(c) 带头指针的单向链表

(d) 顺序存储的线性表

8. 已知一个栈的初始状态为空。现将元素 A, B, C, D, E 按顺序依次入栈，在入栈过程中可以随时出栈（即出栈和入栈操作可交叉进行），则下列选项中，**不可能**得到的出栈序列是（　）。

   (a) C, B, A, E, D

   (b) A, B, C, D, E

   (c) E, D, C, B, A

   (d) D, C, E, A, B

9. 在一个带头结点的双向循环链表中，已知一个结点由指针 `p` 指向。若要在 `p` 所指结点**之后**插入一个由指针 `s` 所指的新结点，下列操作步骤正确的是（　）。

   (a) `p->next = s; s->prior = p; p->next->prior = s; s->next = p->next;`

   (b) `s->prior = p; s->next = p->next; p->next->prior = s; p->next = s;`

   (c) `s->next = p->next; p->next->prior = s; p->next = s; s->prior = p;`

   (d) `p->next->prior = s; p->next = s; s->prior = p; s->next = p->next;`

10. 缺失一道题

二、基础填空题

1. 在计算机中，带符号的数字用 _____ 码表示，便于将加减法统一为加法运算，从而简化硬件设计。

2. 已知：

```
1 int a = 5, b = 2;
2 float c = 1.5, d = 2.0;
```

   则表达式 `(a / b) + c + d` 的值为 _____。

3. 在 C/C++ 中，函数指针本质上是一个变量，它存储的是函数的 _____。通过函数指针可以间接调用其指向的函数，这增强了代码的灵活性，常用于回调函数、动态函数调用等场景。

4. 请你为二维数组 `arr[5][4]` 动态分配内存，使用 `malloc` 和 `calloc` 两种方式分别初始化该数组。(回忆注：这里在答题卡上只给出了两块一行约 6cm 的空间，请每个问题精简的写出一行代码作答)

三、程序填空题

1. 请补全以下程序，实现计算字符串长度的 `My_strlen` 函数。

```
1 #include <stdio.h>
2
3 int My_strlen(char *p)
```

```
{
    int len = 0;
    while (________)
    {
        len++;
        ________;
    }
    return len;
}

int main()
{
    char str[] = "Hello";
    int length = My_strlen(________);
    printf("字符串␣'%s'␣的长度是: %d\n", str, length);
    return 0;
}
```

2. 请补全以下递归函数，使其能计算并返回整数 n 的各位数字之和。例如，输入 123，应返回 6（即 1+2+3）。

```
int sum(int n) {
    if (n < 10) {
        return ________;
    } else {
        return ________ + sum(n / 10);
    }
}
```

3. 缺失一道题

四、简答与分析题

1. 线性表是数据结构的基础，栈是一种操作受限的线性表。请回答以下问题：

   (a) 请简要说明线性表的两种基本存储类型。

   (b) 栈的插入和删除操作与普通数组的插入和删除操作有何本质区别？

   (c) 相较于数组，使用链表实现栈有哪些主要优点？

   (d) 在一个网页浏览器中，需要实现网页标签页的相邻切换（前后切换）和拖拽调整顺序功能。你认为上述哪种数据结构（数组、栈、单向链表、双向链表）最适合用来管理这些标签页？请从这两种功能分析理由。

2. 简述选择排序和直接插入排序的基本思想，并针对序列 {15, 12, 10, 38, 45} 说明第一趟排序后的结果。进一步分析两种算法的平均时间复杂度、平均空间复杂度和稳定性。

3. 简述循环队列相比普通队列的两个主要优点，并说明循环队列中判断队列为空和队列为满的条件，用 front 表示头指针，rear 表示尾指针，用 maxsize 表示队列最大大小。

4. 在单向链表中，已知节点 p 指向某个现有节点，现在需要在 p 之后插入新节点 k。现有两种操作方案：

   - 方案 A：`p->next = k; k->next = p->next;`
   - 方案 B：`k->next = p->next; p->next = k;`

   请问：

   (a) 哪种操作顺序是正确的？

   (b) 如果采用另一种方案进行操作，具体会导致什么后果？请从链表结构状态和后续操作风险两方面说明。

5. 在单向链表中，已知节点 k 指向某个现有节点，若直接执行语句 `k->next = k->next->next;` 来实现删除 k 的下一个节点。

   请问：

   (a) 这条语句能实现删除效果吗？为什么？

   (b) 这种删除方式存在哪些主要缺点？可能引发什么后果？

五、编程实现题

1. 给定一个已按**从小到大排序**的整型数组 `numbers[]`，数组大小为 N，数组索引从 1 开始计数。请找到数组中**两个不同位置**的数，使它们的和等于目标值 `target`。题目保证若解存在则唯一。请将找到的两个下标 `index1` 和 `index2` 存入数组 `arr[]` 中，其中 `arr[0] = index1`，`arr[1] = index2`，且满足 `index1 < index2`。

   请完成以下任务：

   (a) 请写出完整的函数原型。

   (b) 在第 1 问的基础上，使用二分查找的思路，用 C 语言实现该函数。

   (c) 写出第 1 问和第 2 问实现的算法的时间复杂度和空间复杂度（计算时忽略存储结果数组 `arr` 本身的大小）。

2. 假设你正在爬一段有 n 阶台阶的楼梯，从第一阶开始爬，每次只能爬 1 阶或 2 阶。请设计递归函数 `ClimbStairs`，计算爬到楼顶有多少种不同的方法。

   请回答以下问题：

   (a) 什么是递归？一个递归函数必须包含哪两个核心部分？它们各自的作用是什么？

   (b) 根据问题描述，写出计算 `ClimbStairs(n)` 的递归表达式（需包含**基准条件**和**递归条件**）。

   (c) 请用 C 语言编写完整的程序，实现 `ClimbStairs` 递归函数，并在 `main` 函数中测试输入输出。