

AI 出的 C 语言程序设计期末考试试卷

一、单项选择题（本题共 25 小题，每题 2 分，共 50 分）

1. 以下关于 C 语言标识符命名的描述，错误的是：

- A. 标识符只能由字母、数字和下划线组成
- B. 标识符的第一个字符必须是字母或下划线
- C. 标识符区分大小写
- D. 标识符可以是 C 语言关键字

2. 以下哪个不是合法的常量表示形式？

- A. 3.14
- B. 0xFF
- C. 089
- D. 'A'

3. 以下代码的输出是：

```
#include <stdio.h>
int main() {
    int a = 5, b = 2;
    float c = a / b;
    printf("%.2f", c);
    return 0;
}
```

- A. 2.50
- B. 2.00
- C. 2
- D. 2.5

4. 以下关于 switch 语句的描述，正确的是：

- A. switch 后的表达式可以是浮点型
- B. case 后的常量表达式可以重复
- C. default 子句必须放在所有 case 之后
- D. 每个 case 分支结束后必须使用 break

5. 以下代码段中，循环体执行的次数是：

```
int i = 10;
while (i--) {
    printf("%d ", i);
}
```

- A. 9
- B. 10
- C. 11
- D. 无限循环

6. 以下关于函数的说法，错误的是：

- A. 函数可以没有返回值
- B. 函数可以调用自身
- C. 函数参数传递都是“按值传递”
- D. 函数可以返回多个值

7. 以下代码的输出是：

```
#include <stdio.h>
void fun(int x) {
    x = 20;
}
int main() {
    int a = 10;
    fun(a);
    printf("%d", a);
    return 0;
}
```

- A. 10
- B. 20
- C. 编译错误
- D. 不确定

8. 以下哪个是合法的数组初始化？

- A. `int arr[] = {1, 2, 3, 4, 5};`
- B. `int arr[3] = {1, 2, 3, 4};`
- C. `int arr[5] = {};`
- D. `int arr[];`

9. 以下关于指针的描述，错误的是：

- A. 指针变量存储的是内存地址

- B. 指针可以指向另一个指针
- C. 指针可以进行加减运算
- D. 指针可以直接赋值给整型变量

10. 以下代码的输出是：

```
#include <stdio.h>
int main() {
    int arr[] = {1, 2, 3, 4, 5};
    int *p = arr + 2;
    printf("%d", *(p + 1));
    return 0;
}
```

- A. 2
- B. 3
- C. 4
- D. 5

11. 以下关于字符串的描述，正确的是：

- A. 字符串以空格结尾
- B. `char str[] = "hello";` 等价于 `char str[5] = "hello";`
- C. `strlen` 函数返回字符串实际占用内存大小
- D. 字符串可以用 `==` 进行比较

12. 以下哪个函数能正确计算字符串长度？

A.

```
int len(char s[]) {
    int i = 0;
    while (s[i] != ' ') i++;
    return i;
}
```

B.

```
int len(char s[]) {
    int i = 0;
    while (s[i]) i++;
    return i;
}
```

- C. 以上都对
- D. 以上都错

13. 以下关于结构体的描述，错误的是：

- A. 结构体可以嵌套定义
- B. 结构体成员可以是不同数据类型

- C. 结构体变量可以直接用 = 赋值
- D. 结构体大小等于各成员大小之和

14. 以下代码的输出是：

```
#include <stdio.h>
union Data {
    int a;
    char b;
} d;
int main() {
    d.a = 65;
    printf("%c", d.b);
    return 0;
}
```

- A. A
- B. 65
- C. 编译错误
- D. 不确定

15. 以下哪个动态内存分配函数会自动初始化内存为 0？

- A. malloc
- B. calloc
- C. realloc
- D. free

16. 以下关于链表的描述，错误的是：

- A. 链表结点在内存中不连续
- B. 链表插入删除效率高于数组
- C. 链表只能单向遍历
- D. 链表需要额外空间存储指针

17. 以下代码的功能是：

```
int fun(int n) {
    if (n <= 1) return 1;
    return n * fun(n - 1);
}
```

- A. 计算阶乘
- B. 计算斐波那契数列
- C. 计算累加和
- D. 计算幂次

18. 以下关于文件的描述，正确的是：

- A. 文件打开后必须关闭
- B. 文本文件只能存储字符
- C. `fscanf` 可以读取二进制文件
- D. 文件指针可以直接进行算术运算

19. 以下代码的输出是：

```
#include <stdio.h>
#define SQUARE(x) x * x
int main() {
    int a = 3;
    printf("%d", SQUARE(a + 1));
    return 0;
}
```

- A. 16
- B. 10
- C. 12
- D. 7

20. 以下关于枚举类型的描述，正确的是：

- A. 枚举常量是整型常量
- B. 枚举变量只能取枚举列表中的值
- C. 枚举常量可以重复
- D. 枚举类型占用 8 字节内存

21. 以下代码的输出是：

```
#include <stdio.h>
int main() {
    int a = 5, b = 10;
    int *p1 = &a, *p2 = &b;
    p1 = p2;
    *p1 = 20;
    printf("%d %d", a, b);
    return 0;
}
```

- A. 5 10
- B. 5 20
- C. 20 10
- D. 20 20

22. 以下哪个算法的时间复杂度最低？

- A. 冒泡排序
- B. 线性查找
- C. 二分查找
- D. 选择排序

23. 以下关于递归和迭代的描述，正确的是：

- A. 递归总比迭代效率高
- B. 递归必须有终止条件
- C. 迭代不能实现递归所有功能
- D. 递归不会导致栈溢出

24. 以下代码的输出是：

```
#include <stdio.h>
int main() {
    char s[] = "Hello";
    printf("%d", sizeof(s));
    return 0;
}
```

- A. 5
- B. 6
- C. 7
- D. 8

25. 以下关于多文件编程的描述，错误的是：

- A. 头文件通常包含函数声明和宏定义
- B. 源文件包含函数实现
- C. `static` 函数只能在本文件内调用
- D. 全局变量默认在所有文件中可见

二、程序填空题（本题共 5 小题，每空 2 分，共 36 分）

1. 以下程序用于计算并输出一个整数的阶乘，请补全代码：

```
#include <stdio.h>

int factorial(int n) {
    if (n <= 1)
        return _____(1);
    else
        return n * _____(2);
}
```

```

int main() {
    int num;
    printf("请输入一个正整数: ");
    scanf("%d", &num);
    if (num < 0) {
        printf("输入错误! \n");
    } else {
        printf("%d的阶乘是: %d\n", num, fact(num));
    }
    return 0;
}

```

2. 以下程序用于统计字符串中大写字母的个数，请补全代码：

```

#include <stdio.h>
#include <ctype.h>

int countUpper(char str[]) {
    int count = 0, i = 0;
    while (str[i] != '\0') {
        if (isupper(str[i]))
            count++;
        i++;
    }
    return count;
}

int main() {
    char str[100];
    printf("请输入一个字符串: ");
    gets(str);
    printf("大写字母个数: %d\n", countUpper(str));
    return 0;
}

```

3. 以下程序用于将两个已排序的数组合并为一个排序数组，请补全代码：

```

#include <stdio.h>

void merge(int a[], int m, int b[], int n, int c[]) {
    int i = 0, j = 0, k = 0;
    while (i < m & j < n) {
        if (a[i] < b[j])
            c[k++] = a[i++];
        else
            c[k++] = b[j++];
    }
    while (i < m)
        c[k++] = a[i++];
    while (j < n)
        c[k++] = b[j++];
}

```

```

    c[k++] = a[i++];
    while (_____(3))
        c[k++] = b[j++];
}

int main() {
    int a[] = {1, 3, 5, 7, 9};
    int b[] = {2, 4, 6, 8, 10};
    int c[10];
    _____(4)(a, 5, b, 5, c);
    for (int i = 0; i < 10; i++)
        printf("%d ", c[i]);
    return 0;
}

```

4. 以下程序用于计算学生成绩的平均分和最高分，使用结构体数组存储学生信息，请补全代码：

```

#include <stdio.h>

struct Student {
    char name[20];
    int score;
};

int main() {
    struct Student stu[5];
    int i, sum = 0, max = 0, maxIndex = 0;
    for (i = 0; i < 5; i++) {
        printf("输入第%d个学生的姓名: ", i+1);
        scanf("%s", _____(1));
        printf("输入第%d个学生的成绩: ", i+1);
        scanf("%d", _____(2));
        sum += stu[i].score;
        if (stu[i].score > max) {
            max = stu[i].score;
            _____(3) = i;
        }
    }
    printf("平均分: %.2f\n", _____(4));
    printf("最高分: %d, 姓名: %s\n", max, _____(5));
    return 0;
}

```

5. 以下程序用于从文件中读取整数并计算总和，请补全代码：

```

#include <stdio.h>
#include <stdlib.h>

```



```

int main() {
    FILE *fp;
    int num, sum = 0;
    fp = fopen("data.txt", _____(1));
    if (fp == NULL) {
        printf("文件打开失败! \n");
        _____(2);
    }
    while (_____(3)) {
        fscanf(fp, "%d", &num);
        sum += num;
    }
    _____(4);
    printf("总和: %d\n", sum);
    return 0;
}

```

三、编程题（14 分）

题目：学生成绩管理系统

请编写一个 C 语言程序，实现一个简单的学生成绩管理系统。要求使用结构体数组存储学生信息，并实现以下功能：

1. 输入学生信息：输入学生数量（不超过 100 人），然后输入每个学生的学号、姓名和三门课程成绩（数学、英语、C 语言）
2. 计算平均分和总分：为每个学生计算总分和平均分
3. 成绩排序：按总分从高到低排序
4. 查找功能：按学号查找学生信息
5. 统计功能：统计每门课程的平均分、最高分和最低分
6. 输出所有学生信息：按排序后的顺序输出所有学生信息

要求：

1. 使用以下结构体定义：

```

struct Student {
    char id[15];      // 学号
    char name[20];    // 姓名
    float math;       // 数学成绩
    float english;    // 英语成绩
    float c_program;  // C语言成绩
    float total;      // 总分
    float average;    // 平均分
    int rank;         // 名次
}

```

```
};
```

2. 使用以下函数原型：

```
void inputStudents(struct Student stu[], int n); // 输入学生信息
void calculateScore(struct Student stu[], int n); // 计算总分和平均分
void sortStudents(struct Student stu[], int n); // 按总分排序
void searchStudent(struct Student stu[], int n); // 按学号查找
void statistics(struct Student stu[], int n); // 统计各科成绩
void outputStudents(struct Student stu[], int n); // 输出所有学生信息
void updateRank(struct Student stu[], int n); // 更新名次
```

3. 主函数框架如下：

```
int main() {
    struct Student students[MAX_SIZE];
    int n;

    printf("请输入学生数量（不超过%d）：", MAX_SIZE);
    scanf("%d", &n);

    if (n <= 0 || n > MAX_SIZE) {
        printf("输入错误！\n");
        return 1;
    }

    inputStudents(students, n);
    calculateScore(students, n);
    sortStudents(students, n);
    updateRank(students, n);

    // 菜单系统
    int choice;
    do {
        printf("\n===== 学生成绩管理系统 =====\n");
        printf("1. 显示所有学生信息\n");
        printf("2. 按学号查找学生\n");
        printf("3. 统计各科成绩\n");
        printf("4. 退出系统\n");
        printf("请选择功能（1-4）：");
        scanf("%d", &choice);

        switch (choice) {
            case 1:
                outputStudents(students, n);
                break;
            case 2:
                searchStudent(students, n);
                break;
```

```

        case 3:
            statistics(students, n);
            break;
        case 4:
            printf("感谢使用! \n");
            break;
        default:
            printf("输入错误, 请重新选择! \n");
    }
} while (choice != 4);

return 0;
}

```

示例输出格式:

请输入学生数量 (不超过100): 3

输入第1个学生信息:

学号: 2023001

姓名: 张三

数学成绩: 85.5

英语成绩: 78.0

C语言成绩: 92.5

输入第2个学生信息:

...

===== 学生成绩管理系统 =====

1. 显示所有学生信息

2. 按学号查找学生

3. 统计各科成绩

4. 退出系统

请选择功能 (1-4): 1

所有学生信息 (按总分排序):

名次	学号	姓名	数学	英语	C语言	总分	平均分
1	2023001	张三	85.5	78.0	92.5	256.0	85.3
2	2023003	李四	90.0	82.5	80.0	252.5	84.2
3	2023002	王五	78.0	85.0	88.5	251.5	83.8

===== 学生成绩管理系统 =====

1. 显示所有学生信息

2. 按学号查找学生

3. 统计各科成绩

4. 退出系统

请选择功能（1-4）： 3

各科成绩统计：

数学：平均分=84.5，最高分=90.0，最低分=78.0

英语：平均分=81.8，最高分=85.0，最低分=78.0

C语言：平均分=87.0，最高分=92.5，最低分=80.0

注意：

- 请使用`#define MAX_SIZE 100`定义最大学生数量
- 排序时，总分相同的学生按学号升序排列
- 查找时，如果找不到对应学号的学生，应给出提示信息
- 统计功能需要计算每门课程的平均分、最高分和最低分

一、单项选择题（每题 2 分，共 50 分）

1. D 标识符不能是 C 语言关键字。
2. C 089 以 0 开头表示八进制，但八进制中不能出现数字 8 或 9。
3. B a / b 是整数除法，结果为 2，赋给 float c 后为 2.00。
4. C default 子句可以放在任何位置，但通常放在最后。
5. B while (i--) 执行次数为 10 (i 从 10 到 1, i=0 时退出)。
6. D C 语言函数只能返回一个值。
7. A 函数参数是值传递，fun 内部修改不影响 main 中的 a。
8. A 只有 A 是合法的数组初始化方式。
9. D 指针不能直接赋值给整型变量，需要强制类型转换。
10. C p 指向 arr[2]，*(p + 1) 即 arr[3]，值为 4。
11. B char str[] = "hello"; 会自动分配 6 个字节（含 \0）。
12. B A 中判断条件应为 s[i] != '\0'，B 正确。
13. D 结构体大小可能因内存对齐而不等于成员大小之和。
14. A 共用体共用内存，d.a = 65 后 d.b 是 'A'。
15. B calloc 会初始化为 0。
16. C 链表可以单向也可以双向。
17. A 递归计算阶乘。
18. A 文件打开后必须关闭。
19. D SQUARE(a + 1) 展开为 $a + 1 * a + 1$ ，即 $3 + 1*3 + 1 = 7$ 。
20. A 枚举常量本质是整型常量。
21. B p1 指向 b，*p1 = 20 修改的是 b。
22. C 二分查找时间复杂度为 $O(\log n)$ ，最低。
23. B 递归必须有终止条件。
24. B sizeof(s) 包括 \0，所以是 6。
25. D 全局变量默认具有文件作用域，需用 extern 才能跨文件访问。

二、程序填空题（每空 2 分，共 36 分）

1. 阶乘计算

- (1) 1
- (2) factorial(n - 1)
- (3) &num
- (4) factorial(num)

2. 统计大写字母

- (1) str[i] != '\0'
- (2) count++
- (3) count
- (4) scanf
- (5) countUpper(str)

3. 合并有序数组

- (1) i < m && j < n
- (2) i < m
- (3) j < n
- (4) merge

4. 学生成绩统计

- (1) stu[i].name
- (2) &stu[i].score
- (3) maxIndex
- (4) sum / 5.0
- (5) stu[maxIndex].name

5. 文件读取求和

- (1) "r"
- (2) return 1 或 exit(1)
- (3) !feof(fp) 或 fscanf(fp, "%d", &num) != EOF
- (4) fclose(fp)

三、编程题（14 分）

参考以下代码框架：

```
#include <stdio.h>
#include <string.h>
```

```

#define MAX_SIZE 100
#include <stdio.h>
#include <string.h>

#define MAX_SIZE 100

struct Student {
    char id[15];
    char name[20];
    float math;
    float english;
    float c_program;
    float total;
    float average;
    int rank;
};

void inputStudents(struct Student stu[], int n) {
    for (int i = 0; i < n; i++) {
        printf("\n输入第%d个学生信息: \n", i + 1);
        printf("学号: "); scanf("%s", stu[i].id);
        printf("姓名: "); scanf("%s", stu[i].name);
        printf("数学成绩: "); scanf("%f", &stu[i].math);
        printf("英语成绩: "); scanf("%f", &stu[i].english);
        printf("C语言成绩: "); scanf("%f", &stu[i].c_program);
    }
}

void calculateScore(struct Student stu[], int n) {
    for (int i = 0; i < n; i++) {
        stu[i].total = stu[i].math + stu[i].english + stu[i].c_program;
        stu[i].average = stu[i].total / 3;
    }
}

void sortStudents(struct Student stu[], int n) {
    for (int i = 0; i < n - 1; i++) {
        for (int j = 0; j < n - i - 1; j++) {
            if (stu[j].total < stu[j + 1].total ||
                (stu[j].total == stu[j + 1].total && strcmp(stu[j].id, stu[j + 1].id) > 0)) {
                struct Student temp = stu[j];
                stu[j] = stu[j + 1];
                stu[j + 1] = temp;
            }
        }
    }
}

void updateRank(struct Student stu[], int n) {

```

```

    for (int i = 0; i < n; i++) {
        stu[i].rank = i + 1;
    }
}

void searchStudent(struct Student stu[], int n) {
    char searchId[15];
    printf("请输入要查找的学号: ");
    scanf("%s", searchId);
    for (int i = 0; i < n; i++) {
        if (strcmp(stu[i].id, searchId) == 0) {
            printf("找到学生: \n");
            printf("学号: %s, 姓名: %s, 数学: %.1f, 英语: %.1f, C语言: %.1f, 总分: %.1f, 平均",
                stu[i].id, stu[i].name, stu[i].math, stu[i].english,
                stu[i].c_program, stu[i].total, stu[i].average, stu[i].rank);
            return;
        }
    }
    printf("未找到学号为 %s 的学生。 \n", searchId);
}

void statistics(struct Student stu[], int n) {
    float mathSum = 0, engSum = 0, cSum = 0;
    float mathMax = stu[0].math, mathMin = stu[0].math;
    float engMax = stu[0].english, engMin = stu[0].english;
    float cMax = stu[0].c_program, cMin = stu[0].c_program;

    for (int i = 0; i < n; i++) {
        mathSum += stu[i].math;
        engSum += stu[i].english;
        cSum += stu[i].c_program;

        if (stu[i].math > mathMax) mathMax = stu[i].math;
        if (stu[i].math < mathMin) mathMin = stu[i].math;

        if (stu[i].english > engMax) engMax = stu[i].english;
        if (stu[i].english < engMin) engMin = stu[i].english;

        if (stu[i].c_program > cMax) cMax = stu[i].c_program;
        if (stu[i].c_program < cMin) cMin = stu[i].c_program;
    }

    printf("\n各科成绩统计: \n");
    printf("数学: 平均分=%.1f, 最高分=%.1f, 最低分=%.1f\n", mathSum / n, mathMax, mathMin);
    printf("英语: 平均分=%.1f, 最高分=%.1f, 最低分=%.1f\n", engSum / n, engMax, engMin);
    printf("C语言: 平均分=%.1f, 最高分=%.1f, 最低分=%.1f\n", cSum / n, cMax, cMin);
}

void outputStudents(struct Student stu[], int n) {
    printf("\n所有学生信息 (按总分排序): \n");

```



```

printf("| 名次 | 学号      | 姓名 | 数学 | 英语 | C语言 | 总分 | 平均分 |\n");
for (int i = 0; i < n; i++) {
    printf("| %-4d | %-9s | %-3s | %-5.1f | %-5.1f | %-6.1f | %-6.1f | %-6.1f |\n",
        stu[i].rank, stu[i].id, stu[i].name,
        stu[i].math, stu[i].english, stu[i].c_program,
        stu[i].total, stu[i].average);
}
}

int main() {
    struct Student students[MAX_SIZE];
    int n;

    printf("请输入学生数量（不超过%d）：", MAX_SIZE);
    scanf("%d", &n);

    if (n <= 0 || n > MAX_SIZE) {
        printf("输入错误! \n");
        return 1;
    }

    inputStudents(students, n);
    calculateScore(students, n);
    sortStudents(students, n);
    updateRank(students, n);

    int choice;
    do {
        printf("\n===== 学生成绩管理系统 =====\n");
        printf("1. 显示所有学生信息\n");
        printf("2. 按学号查找学生\n");
        printf("3. 统计各科成绩\n");
        printf("4. 退出系统\n");
        printf("请选择功能（1-4）：");
        scanf("%d", &choice);

        switch (choice) {
            case 1: outputStudents(students, n); break;
            case 2: searchStudent(students, n); break;
            case 3: statistics(students, n); break;
            case 4: printf("感谢使用! \n"); break;
            default: printf("输入错误，请重新选择! \n");
        }
    } while (choice != 4);

    return 0;
}

```

功能说明

- `inputStudents`: 输入学生信息。
- `calculateScore`: 计算总分和平均分。
- `sortStudents`: 按总分从高到低排序，总分相同按学号升序。
- `updateRank`: 更新名次。
- `searchStudent`: 按学号查找学生。
- `statistics`: 统计各科平均分、最高分、最低分。
- `outputStudents`: 输出所有学生信息表格。
- 菜单系统: 循环选择功能，直到退出。