

## 829 部分考试题目

### 选择题

1. 函数递归调用可以**直接或者间接调用函数本身**
2. %只能是用于**整数**运算的字符
3. 数组下标的数据类型允许是：**整形常量和整形表达式**
4. 算法的五个特性：**有穷性、确定性、有零个或多个输入、有一个或多个输出、有效性**
5. 识别 `int *px[6]` **//定义指针数组**
6. 定义 `int n, a[n];` **//不能定义动态数组**
7. 判断 `10! = 9` 的值 **// 1**
8. 打开文件“w”是什么方式 **//只写打开，文件不存在则创建新文件**
9. C 语言中于**不可嵌套的是函数定义**
10. `x=-1; do{x=x*x;} while(!x);` 的循环次数 **//1 次**
11. 实参为数组名，则向函数传递的是？**数组起始的地址**
12. `int d[3][3]={ {1, 2}, {1, 2, 3}, {1, 2, 3, 4} }`; 等数组的定义 **//错误，不能超过 3 个元素**
13. 计算 1 至 50 中 7 的倍数的数值之和(选择是判别 7 的条件表达式)**//i%7==0**
14. 一个 for 语句中包含着 switch, 然后进行输出, 考查的是 case 项后没有 break 应该如何输出
15. `float x=213.82631; printf("%3d", (int)x);` 输出的值**//213, 只截取整数部分即可。【注】若"%4d", 则输出占四个位, 213, 前面有一个空格。**

### 判断题

16. 递归调用可以直接或间接调用该函数本身 **T**
17. c 语言的源程序必须翻译为汇编语言后才能正常执行 **F, 应该是机器码 (二进制文件)**
18. 数组中每一元素必须属于统一数据类型 **T**
19. C 语言中 break 语句只能结束本次循环, 而不是终止整个的循环 **F, 应该是 continue**
20. C 语言属于面向对象程序设计语言 **F, 应该是面向过程**
21. 算法是为了解决一个问题而采取的方法和步骤, 算法=数据结构+程序 **F, 应该是: 程序=数据结构+算法**
22. 在 C 语言中, 要调用函数必须在 main() 函数中定义 **F, 可以再外部定义**
23. Do...while 语句的执行过程是: 先判断循环条件是否成立, 然后再执行循环体 **F, 先执行, 后判断条件**
24. for 循环只能用于循环次数已经确定的情况 **F, 也可用于循环次数不定的情况。**
25. C 语言规定标识符只能由字符、数字和下划线三种字符组成, 第一个字符不能为数字 **T**

### 简答题

## 829 部分考试题目

26. 简述结构化程序设计的优缺点：

✚ 优点：

采用了模块分解与功能抽象和自顶向下、分而治之的方法，从而有效地将一个较复杂的程序系统设计任务分解成许多易于控制和处理的子程序，便于开发和维护。

✚ 缺点：

它不能很好的适应需求变化的开发过程。

27. 简述循环体中 break 和 continue 的区别

✚ Break 使流程跳到循环体之外

✚ Continue 结束本次循环

28. 简述算法的主要特征（5）

✚ 有穷性、确定性、有零个或多个输入、有一个或多个输出、有效性。

29. 机器语言和高级语言各有什么特点

✚ 机器语言特点：

■ 优点：

机器语言能够被计算机立即识别并加以执行，具有执行速度快，占用内存少等优点。

■ 缺点：

对于使用者来说，用机器语言编写程序具有编写难，识别难，记忆难，查错难，交流难等缺点。

✚ 高级语言特点：

■ 优点：

这种语言弥补了机器语言的不足，用汇编语言编写程序比用机器语言方便，直观，易懂，易用，易记，可以编写出结构紧凑，运行时间精确的程序。所以，这种语言非常适合于实时控制的需要。

■ 缺点：

不能被计算机立即识别并加以执行，具有执行速度相对较慢，占用内存更多等缺点。

30. 简要阐述使用指针的优点和缺点

✚ 使用指针可以带来如下的好处：

■ 可以提高程序的编译效率和执行速度，使程序更加简洁。

■ 通过指针被调用函数可以向调用函数处返回除正常的返回值之外的其他数据，从而实现两者间的双向通信。

■ 利用指针可以实现动态内存分配。

■ 指针还用于表示和实现各种复杂的数据结构，从而为编写出更加高质量的程序奠定基础。

■ 利用指针可以直接操纵内存地址，从而可以完成和汇编语言类似的工作。

■ 更容易实现函数的编写和调用。

✚ 使用指针可以带来如下的缺点：

■ 如果对指针不能正确理解和灵活有效的应用，利用指针编写的程序也更容易隐含各式各样的错误。

■ 同时程序的可读性也会大打折扣。

31. 简述函数的参数传递时传值和传地址的区别：

## 829 部分考试题目

### ✚ 值传递:

主调函数传递给被调函数的是值的拷贝, 不是原值, 只是将变量的内容复制一份而已, 函数进行操作的其实是另一个变量, 只是另一个变量的值和传递的变量值是相同的。

### ✚ 地址传递:

主调函数传递给被调函数的是值的地址。直接把变量的地址传递给函数, 这时函数是直接对原来的变量进行操作的, 所以值会变化。

### ✚ 区别:

值传递被调函数中的操作不改变主调函数的值, 而地址传递则不同。

### 32. 简述#include<stdio.h>与#include"stdio.h"的区别

#### ✚ 用尖括号 #include <>:

一般用于包含标准的库头文件, 编译器会去系统配置的库环境变量和者用户配置的路径去搜索, 而不会在项目的当前目录去查找。

#### ✚ 用双引号 #include "":

一般用于包含用户自己编写的头文件, 编译器会先在项目的当前目录查找, 找不到后才会去系统配置的库环境变量和用户配置的路径去搜索。

## 案例分析题

### 33. 给一个四位数, 先判断是不是 11 的整数倍, 若是则输出, 不是就把四位数重新排列, 在形成的新的四位数中判断是不是 11 的整数倍, 若是则存到文件"output.dat"; 然后再求文件"output.dat"中最大的数, 输出。问答有两问每问 10 分。

(1) 简述算法思路和画出流程图

(2) 代码

参考代码如下:

```
void main() {
    int n, max = 0;
    scanf("%d", &n);
    if (n % 11 == 0) {
        printf("n=:%d\n", n);
    }
    else{
        FILE* fp = fopen("output.dat", "w+");
        if (fp == NULL) {
            printf("file open failure!\n");
            return;
        }
        else{
            int num[4];
            for (int i = 0; i < 4; i++)
            {
                num[3 - i] = n % 10;
```

## 829 部分考试题目

```
        n /= 10;
    }
    for (int i = 0; i < 4; i++)
    {
        for (int j = 0; j < 4; j++)
        {
            for (int k = 0; k < 4; k++)
            {
                for (int p = 0; p < 4; p++)
                {
                    if (i == j || i == k || i == p || j == k ||
j == p || k == p)
                    {
                        continue;
                    }
                    else
                    {
                        int temp = num[i] * 1000 + num[j] * 100
+ num[k] * 10 + num[p] * 1;
                        if (temp % 11 == 0)
                        {
                            if (max < temp)
                                max = temp;
                            fprintf(fp, "%d ", temp);
                        }
                    }
                }
            }
        }
    }
    fclose(fp);
    printf("max=:%d\n", max);
}
```

34. 一物体从 200 落下，每次弹回高度减半，问第 10 次的总路程和第 10 次的高度。

**参考代码：**

```
double s=100,h=s/2;
for(int i=2;i<=10;i++){
    s += 2 * h;
    h /= 2;
}
cout<<s<<" \n" <<h<<endl;
```