

江西师范大学硕士研究生入学考试初试科目
考试大纲

科目代码、名称： 869C 语言程序设计与数据结构；

适用专业： 120100 管理科学与工程 01、04 方向

一、考试形式与试卷结构

(一) 试卷满分 及 考试时间

本试卷满分为 150 分，考试时间为 180 分钟。

(二) 答题方式

答题方式为闭卷、笔试。

试卷由试题和答题纸组成；答案必须写在答题纸相应的位置上。

(三) 试卷题型结构

简答题（5-10 小题，共 50 分）

填空题（5 个空，共 10 分）

判断题（5 小题，共 10 分）

应用题（2-3 小题，共 30 分）

程序设计题（2-5 小题，共 50 分）

二、考查目标（复习要求）

全日制攻读硕士学位研究生入学考试 C 语言程序设计与数据结构科目考试内容
包括 C 语言程序设计、数据结构等 2 门学科基础课程，要求考生系统掌握相关学科
的基本知识、基础理论和基本方法，并能运用相关理论和方法分析、解决实际问题。

三、考查范围或考试内容概要

第一部分：C 语言程序设计

第一章 程序设计和 C 语言

1. 什么是计算机程序
2. 什么是计算机语言
3. C 语言的发展及其特点
4. 最简单的 C 语言程序
5. 运行 C 程序的步骤与方法
6. 程序设计的任务

第二章 算法

1. 什么是算法
2. 简单的算法举例
3. 算法的特性

4. 怎样表示一个算法
5. 结构化程序设计方法

第三章 顺序程序设计

1. 数据的表现形式及其运算
2. C 语句
3. 数据的输入输出

第四章 选择结构程序设计

1. 选择结构和条件判断
2. 用 if 语句实现选择结构
3. 关系运算符和关系表达式
4. 逻辑运算符和逻辑表达式
5. 条件运算符和条件表达式
6. 选择结构的嵌套
7. 用 switch 语句实现多分支选择结构

第五章 循环结构程序设计

1. 用 while 语句实现循环
2. 用 do...while 语句实现循环
3. 用 for 语句实现循环
4. 循环的嵌套
5. 几种循环的比较
6. 改变循环执行的状态

第六章 利用数组处理批量数据

1. 一维数组
2. 二维数组
3. 字符数组

第七章 用函数实现模块化程序设计

1. 为什么要用函数
2. 怎样定义函数
3. 调用函数
4. 对被调用函数的声明和函数原型
5. 函数的嵌套调用
6. 函数的递归调用
7. 数组作为函数参数
8. 局部变量和全局变量
9. 变量的存储方式和生存期
10. 关于变量的声明和定义
11. 内部函数和外部函数

第八章 指针

1. 指针是什么

2. 指针变量
3. 通过指针引用数组
4. 通过指针引用字符串
5. 指向函数的指针
6. 返回指针值的函数
7. 指针数组和多重指针
8. 动态内存分配与指向它的指针变量

第九章 自定义数据类型

1. 结构体变量
2. 结构体数组
3. 结构体指针
4. 用指针处理链表
5. 共用体类型
6. 枚举类型
7. 用 typedef 声明新类型名

第十章 文件的输入输出

1. C 文件的基本知识
2. 打开与关闭文件
3. 顺序读写数据文件
4. 随机读写数据文件
5. 文件读写的出错检测

参考教材或主要参考书:

1. C 程序设计(第四版), 谭浩强, 清华大学出版社, 2010 年 6 月

第二部分: 数据结构

第一章 绪论

1. 数据结构的基本概念
2. 抽象数据类型
3. 算法和算法的时间复杂度
4. 算法书写规范

第二章 线性表

1. 线性表概述
2. 线性表的顺序表示和实现
3. 线性表的链式表示和实现
4. 静态链表
5. 算法设计举例

第三章 堆栈和队列

1. 堆栈
2. 堆栈应用
3. 队列

4. 优先级队列

第四章 串

1. 串概述
2. 串的存储结构
3. 串基本操作的实现算法
4. 串的模式匹配算法

第五章 数组

1. 数组概述
2. 动态数组
3. 特殊矩阵的压缩存储
4. 稀疏矩阵的压缩存储

第六章 递归算法

1. 递归的概念
2. 递归算法的执行过程
3. 递归算法的设计方法
4. 递归过程和运行时栈
5. 递归算法的时间效率分析
6. 递归算法到非递归算法的转换

第七章 广义表

1. 广义表概述
2. 广义表的存储结构
3. 广义表的操作实现

第八章 树和二叉树

1. 树
2. 二叉树
3. 二叉树的设计和实现
4. 二叉树遍历
5. 线索二叉树
6. 哈夫曼树
7. 等价问题
8. 树与二叉树的转换
9. 树的遍历

第九章 图

1. 图概述
2. 图的存储结构
3. 图的实现
4. 图的遍历
5. 最小生成树
6. 最短路径

7. 拓扑排序
8. 关键路径

第十章 排序

1. 排序的基本概念
2. 插入排序
3. 选择排序
4. 交换排序
5. 归并排序
6. 基数排序
7. 排序算法性能比较

第十一章 查找

1. 查找的基本概念
2. 静态查找
3. 动态查找
4. 哈希查找

第十二章 文件

1. 文件概述
2. 顺序文件
3. 索引文件
4. ISAM 文件
5. VSAM 文件
6. 散列文件

参考教材或主要参考书:

1. 《数据结构—使用 C 语言 (第 4 版)》, 朱战立, 电子工业出版社, 2009 年 1 月

四、样卷

江西师范大学硕士研究生入学考试试题样卷

一、简答题 (10 小题, 每小题 5 分, 共 50 分)

- 1、 算法具有哪些特点?
- 2、 C 语言中的主要数据类型有哪些?
- 3、 C 语言中 break 语句与 continue 语句有何区别?
- 4、 什么是形式参数? 什么是实际参数?
- 5、 将以下 for 循环改写为 while 循环:

```
for (sum=0, i=0; i<1000; sum+=2*I, i++);  
printf(“sum=%d\n”, sum);
```

- 6、 简述结构化程序设计的基本思想。
- 7、 写出抽象数据类型栈的主要操作。

8、什么是满二叉树？什么是完全二叉树？

9、什么是内部排序？什么是外部排序？

10、写出二叉树三种遍历操作的定义。

二、 填空题（5个空，每空2分，共10分）

1、组成C语言源程序的基本单位是_____。

2、以下定义中，值为5的枚举常量是_____。

```
enum week {sun, mon=4, tue, wed, thu, fri, sat};
```

3、假设在程序中 a, b, c 均被定义成 int 型，则能正确表示代数式 $a + \frac{1}{b+c}$ 的 C 语言表达式是_____。

4、若已定义 a=8, b=2, 则表达式 a && (a-b*4) || b 的值为_____。

5、若有下列说明和赋值语句：

```
int a=8, b;
```

```
b=++a;
```

则变量 b 的值是_____。

三、 判断题（5小题，每小题2分，共10分）

1、队列是一种插入与删除操作分别在表的两端进行的线性表，是一种先进后出型结构。

2、用散列方法存储，查找效率不高。

3、线性表的存储方式既可以是顺序方式，也可以是链式方式。

4、二叉树中所有结点，如果不存在非空左子树，则不存在非空右子树。

5、用二叉链表法存储包含 n 个结点的二叉树，结点的 2n 个指针区域中有 n+1 个为空指针。

四、 应用题（3小题，每小题10分，共30分）

1、试分别画出具有 3 个结点的树和具有 3 个结点的二叉树的所有不同形态。

2、已知序列 40, 30, 50, 24, 28, 46, 60, 10。试画出由该输入序列构成的二叉排序树，并分别给出依次执行下列操作后的二叉排序树（共画四棵树）

(1) 插入数据 42 和 80；(2) 删除数据 30；(3) 删除数据 50。

3、请简述直接插入排序的基本思想，并对于下列给定数据 {27, 46, 5, 18, 16, 51, 32, 26} 写出其排序过程。

五、 程序设计题（5小题，每小题10分，共50分）

1、设计一个函数，利用单链表原来的结点空间将一个带头结点的单链表就地转置。假定程序中已有说明如下：

```
typedef struct node  
{  
    int data;  
    struct node *next;
```

```
}linknode;
```

```
typedef linknode *linklist;
```

2、 试编写一个函数，实现在顺序存储方式下字符串比较运算的函数 `strcmp(seqstring s1, seqstring s2)`。假定程序中已有说明如下：

```
#define MAXSIZE 100
```

```
typedef struct{
```

```
    char str[MAXSIZE];
```

```
    int length;
```

```
}seqstring;
```

3、 试编写一个 C 程序，输入一个 10 进制非负整数，将该数转换为 8 进制整数，并计算其各位数字之和。

4、 试编写一个 C 程序，利用公式 $\pi = 4 * (1 - \frac{1}{3} + \frac{1}{5} - \frac{1}{7} + \frac{1}{9} - \dots + \frac{1}{10001})$ ，计算 π 的近似值。

5、 试编写一个递归函数，将一颗给定二叉树中所有结点的左右子女互换。假定程序中已有说明如下：

```
typedef struct_BiTNode{
```

```
    int data;
```

```
    struct_BiTNode *lchild;
```

```
    struct_BiTNode *rchild;
```

```
}BiTNode, *pBiTNode;
```