

全国计算机等级考试二级C系列用书

全国计算机等级考试教程
 全国计算机等级考试上机考试题库
全国计算机等级考试笔试模拟考场
 全国计算机等级考试三合一
 全国计算机等级考试一本通
 全国计算机等级考试点分析、题解与模拟
 全国计算机等级考试试题精编四合一
 全国计算机等级考试历年试卷及详解汇编



智能考试学习软件

登录、抽题、答题、交卷与正式考试一模一样
 自动评分、提供详细的答题步骤和标准答案
 全部源自最新考试题库，命中率高
 临考前上网升级软件，惊喜获得考前冲刺试题



新浪教育



网易教育频道
education.163.com



搜狐教育
edu.qq.com



腾讯教育
edu.qq.com

强力推荐

ISBN 978-7-900223-12-8



9 787900 122312 >



全国计算机等级考试优秀图书

1CD+配套手册 不得拆分销售

08年4月
考试专用

全国计算机等级考试命题研究中心
未来教育考试与研究 中心

新大纲

最新大纲

精析最新考试大纲，掌握最新考试要点

考试题库

大量上机试题，完全覆盖真考题库

模拟考场

真考环境+智能评分，带您提前“进入”考场

视频教学

名师演示，像看电影一样轻松学习



全国计算机等级考试

笔试

模拟考场

二级C

7套全真模拟+2套专家预测+3套最新真题



金版电子出版社



金版电子出版社

丛书编委会

主任：熊化武

编委：(排名不分先后)

丁海艳	万克星	马立娟	亢艳芳
王伟	王亮	王强国	王磊
王海梅	王海霞	付红伟	卢文毅
卢继军	田建鲲	任海艳	刘丹
刘微	刘金丽	刘春波	孙海
孙爱琴	谷永生	严平	张莹
张强	张仪凡	李丽	李岩
李琴	李静	李剑锋	李东梅
李可诚	李可锋	李志红	张迪
杨力	杨闯	杨生喜	杨贵宾
花英	陈秋彤	周辉	孟祥勇
欧海升	武杰	范海双	郑新
姜涛	姜文宾	胡杨	胡天星
赵亮	赵东红	倪海宇	钱刚
高志军	高雪轩	聂钰楨	曾敏
董国明	蒋芳	谢公义	韩峻余
熊化武	谭彪	潘海杰	薛海东

内容简介

2008年计算机等级考试在新大纲标准下实施,为了向考生提供专业的辅导用书,全国计算机等级考试命题研究中心、未来教育教学与研究联合设计、开发了本套试卷。

本套试卷一改等级考试用书的常规编写方法,在对新大纲与历年试卷进行深入研究之后,精心设计了符合命题规律的试卷结构:“全真模拟试卷”、“命题专家预测试卷”及“历年试卷”3种试卷,科学搭配,结合详尽的名师解析,不仅提供给考生循序渐进的练习方式,同时通过个性化的解题指导让考生由浅入深地掌握应试的必备知识,顺利地通过考试。

书名: 全国计算机等级考试笔试模拟考场——二级 C

技术总监: 聂钰楨

文本编辑: 韩富仙

出版发行: 金版电子出版社

地址: 北京市实兴大街5号

电话: (010)88792916

全国代理: 北京金飞腾图书有限公司

发行电话: (010)82552266 82552277

经销: 各地新华书店、软件连锁店

文本印刷: 北京市昌平百善印刷厂

版次: 2007年10月第4版第1次印刷

定价: 15.00元 (1CD + 配套手册)

目 录

全国计算机等级考试

笔试模拟考场

二级 C

(08年4月考试专用)

全国计算机等级考试命题研究中心
未来教育教育与研究中心

金版电子出版社

全真模拟试卷

考试大纲及应试指导	(共4页)
全真模拟试卷(1)	(共8页)
全真模拟试卷(2)	(共8页)
全真模拟试卷(3)	(共8页)
全真模拟试卷(4)	(共8页)
全真模拟试卷(5)	(共8页)
全真模拟试卷(6)	(共8页)
全真模拟试卷(7)	(共8页)

命题专家预测试卷

命题专家预测试卷(1)	(共8页)
命题专家预测试卷(2)	(共8页)

历年真题试卷

2006年9月笔试试卷	(共12页)
2007年4月笔试试卷	(共12页)
2007年9月笔试试卷	(共12页)
参考答案及解析	(共40页,另分册)

公共基础知识(略)

C语言程序设计基本要求

1. 熟悉 Visual C++ 6.0 集成开发环境。
2. 掌握结构化程序设计的方法,具有良好的程序设计风格。
3. 掌握程序设计中简单的数据结构和算法并能阅读简单的程序。
4. 在 Visual C++ 6.0 集成环境下,能够编写简单的 C 程序,并具有基本的纠错和调试程序的能力。

C语言程序设计考试内容

一、C语言程序的结构

1. 程序的构成, main 函数和其他函数。
2. 头文件, 数据说明, 函数的开始和结束标志以及程序中的注释。
3. 源程序的书写格式。
4. C 语言的风格。

二、数据类型及其运算

1. C 的数据类型(基本类型, 构造类型, 指针类型, 无值类型)及其定义方法。
2. C 运算符的种类、运算优先级和结合性。
3. 不同类型数据间的转换与运算。
4. C 表达式类型(赋值表达式, 算术表达式, 关系表达式, 逻辑表达式, 条件表达式, 逗号表达式)和求值规则。

三、基本语句

1. 表达式语句, 空语句, 复合语句。
2. 输入输出函数的调用, 正确输入数据并正确设计输出格式。

四、选择结构程序设计

1. 用 if 语句实现选择结构。
2. 用 switch 语句实现多分支选择结构。
3. 选择结构的嵌套。

五、循环结构程序设计

1. for 循环结构。
2. while 和 do-while 循环结构。
3. continue 语句和 break 语句。
4. 循环的嵌套。

六、数组的定义和引用

1. 一维数组和二维数组的定义、初始化和数组元素的引用。
2. 字符串与字符数组。

七、函数

1. 库函数的正确调用。
2. 函数的定义方法。
3. 函数的类型和返回值。
4. 形式参数与实在参数, 参数值的传递。
5. 函数的正确调用, 嵌套调用, 递归调用。
6. 局部变量和全局变量。
7. 变量的存储类别(自动, 静态, 寄存器, 外部), 变量的作用域和生存期。

八、编译预处理

1. 宏定义和调用(不带参数的宏, 带参数的宏)。
2. “文件包含”处理。

九、指针

1. 地址与指针变量的概念, 地址运算符与间址运算符。
2. 一维、二维数组和字符串的地址以及指向变量、数组、字符串、函数、结构体的指针变量的定义。通过指针引用以上各类型数据。
3. 用指针作函数参数。
4. 返回地址值的函数。
5. 指针数组, 指向指针的指针。

十、结构体(即“结构”)与共同体(即“联合”)

1. 用 typedef 说明一个新类型。
2. 结构体和共用体类型数据的定义和成员的引用。
3. 通过结构体构成链表, 单向链表的建立, 结点数据的输出、删除与插入。

十一、位运算

1. 位运算符的含义和使用。
2. 简单的位运算。

十二、文件操作

只要求缓冲文件系统(即高级磁盘 I/O 系统), 对非标准缓冲文件系统(即低级磁盘 I/O 系统)不要求。

1. 文件类型指针(FILE 类型指针)。
2. 文件的打开与关闭(fopen, fclose)。
3. 文件的读写(fputc, fgetc, fputs, fgets, fread, fwrite, fprintf, fscanf 函数的应用), 文件的定位(rewind, fseek 函数的应用)。

C语言程序设计考试方式

1. 笔试: 90 分钟, 满分 100 分, 其中含公共基础知识部分的 30 分。
2. 上机: 90 分钟, 满分 100 分。
上机操作包括:
(1) 填空。 (2) 改错。 (3) 编程。

应试指导

全国计算机等级考试,是对非计算机专业的人员进行计算机知识结构与水平认定的重要考试。全国计算机等级考试——二级考试涉及面广,覆盖内容多,而且具有一定的深度和难度。因此,考生要想通过考试关,并非一件容易事。为了帮助考生应试,我们特意撰写此文,对一些考试规则和考生普遍关心的问题进行了较为详细的说明和分析,希望对考生备考有所启发和裨益。

一 命题原则

《全国计算机等级考试二级大纲》(以下简称“大纲”)明确规定:二级(C语言程序设计)“二级公共基础知识”和“程序设计”两大部分组成,考试内容严格按照“宽口径、厚基础”的原则设计。试题主要测试考生对本学科的基础理论、基本知识和基本技能的掌握程度,以及运用所学理论知识解决实际问题的能力。

二 考试要求

根据命题原则,大纲对考生如何复习应试也提出了相应的要求。要求主要强调了两个方面:一是强调考生对基本概念、基本理论和基本知识的掌握程度。二是强调考生综合运用所学知识进行实际应用的能力。也就是说,考生要想通过全国计算机等级考试,不仅要熟练地掌握本学科的基本理论和操作技能,还要具有“较强”的分析与解决实际问题的能力,体现出“学以致用”的学习目的。否则,考试就难以取得好成绩。

三 应试对策

考生要针对大纲和考试要求进行复习准备,主要应注意以下几个方面:

● 牢固、清晰地掌握基本知识和理论

“二级”考试的重点是实际应用和操作,但是实际应用和操作的前提是基本知识的掌握。那么,考生对基本概念和原理的记忆与理解是否准确便是考试得分的关键。考生如何牢固、清晰地掌握呢?具体地讲,不外乎三点:一是在复习过程中要注意总结,特别是对一些关系复杂的知识点。二是对一些重要概念的理解要准确,尤其是一些容易混淆的概念,如多种操作方法等,一定要在复习中准确地把握住它们之间在步骤和实现意义上的细微区别。对这些易混淆概念的准确理解,考生不可忽视。三是通过联想记忆寻找考点,有些考点不是孤立的,而是相互联系的,由表及里、由此及彼便能顺利地找到答案。

● 多做练习

在复习过程中,多做一些模拟训练题,这是最快、最有效的方法之一,它能在最短的时间内收到最明显的效果。尤其是对那些复习时间有限和工作较繁忙的考生来说,这是应对考试

的最好方法,既能加深考生对考试内容的印象,又能帮助考生较快地掌握答题技巧。

四 题型分析与解题技巧

单选题和填空题一般是对基本知识和基本操作进行考查,它主要是测试考生对基本的和重要的理论点和概念是否掌握、理解是否准确、认识是否全面、思路是否清晰,而很少涉及理论知识的应用。具体地说,考试时应注意以下几个方面:

● 选择题分析

选择题为单选题,多选、选错或不选都不给分,但选错也不倒扣分,考生在做答选择题时要注意。

选择题试题部分覆盖面广,各类内容的试题均有,一般情况下,考生根本不可能做到对大部分的题目都有把握。这时,需要考生学会放弃,即不确定的题目不要在上面纠缠过多,应该在此题上做上标记,立即转移注意力,做答其他题目,最后有空余的时间再回过头来仔细考虑此题。但要注意,对于那些实在不清楚的题目,就不要浪费时间了,放弃继续思考,不要因小失大。注意,二级笔试题目众多,分值分散,考生一定要有全局观,合理安排时间。考生在做答选择题时可以使用一些小技巧,提高选择命中率:

- ◆ 直接选择:如果对题中的4个选项,一看就能肯定其中的一个是正确的,就可以直接得出答案。注意,必须有百分之百的把握才行。
- ◆ 排除选择:对4个选项,一看就知道其中的一个(或两个、3个)是错误的,可以使用排除法,即排除错误选项,剩下的为正确答案。
- ◆ 估计选择:在排除法中,如果最后还剩下几个选项,或对某题一无所知时,在剩下的选项中随机选一个,因为错选了也不倒扣分,所以应该不漏选,每题都选一个答案,这样可以一定程度上提高考试成绩。

● 填空题分析

填空题一般难度都比较大,一般需要考生填入准确的字符,往往需要非常的精确,错一个字都不得分。建议考生对填空题不要过于看重,为个别问题耽误时间,不如回过头检查一些不确定的选择题,如能检查出两题,在分值上就抵得上了一道填空题。在做填空题一定要注意:

- ◆ 认真填写答案,字迹要工整、清楚,格式要正确,文字上不能有一点点的错误。
 - ◆ 注意,填空题往答题卡填写时,是按照空号的顺序,不要弄错位置。
 - ◆ 对于那些有两种答案的只需填一种就可以了,多填并不多给分。
- 总之,考试要取得好成绩,从根本上取决于考生对应试内容掌握的扎实程度。否则,即使有再好的技巧也只能是撞大运,是不可能考出理想成绩的。但是,在比较扎实地掌握了应试内容的前提下,了解一些应试的技巧则能起到使考试成绩锦上添花的作用。

全国计算机等级考试全真模拟试题卷(1)

二级公共基础知识和C语言程序设计

(考试时间90分钟,满分100分)

一、选择题(每小题2分,共70分)

在下列各题的A)、B)、C)、D)四个选项中,只有一个选项是正确的,请将正确选项填涂在答题卡相应位置上,答在试卷上不得分。

- (1) 栈和队列的共同特点是
 A) 都是先进先出 B) 都是先进后出
 C) 只允许在端点处插入和删除元素 D) 没有共同点
- (2) 已知二叉树后序遍历序列是 dabec, 中序遍历序列是 debac, 它的前序遍历序列是
 A) acbed B) decab C) deabc D) cedba
- (3) 链表不具有的特点是
 A) 不必事先估计存储空间 B) 可随机访问任一元素
 C) 插入删除不需要移动元素 D) 所需空间与线性表长度成正比
- (4) 结构化程序设计的3种结构是
 A) 顺序结构、选择结构、转移结构 B) 分支结构、等价结构、循环结构
 C) 多分支结构、赋值结构、等价结构 D) 顺序结构、选择结构、循环结构
- (5) 为了提高测试的效率,应该
 A) 随机选取测试数据
 B) 取一切可能的输入数据作为测试数据
 C) 在完成编码以后制定软件的测试计划
 D) 集中对付那些错误群集的程序
- (6) 算法的时间复杂度是指
 A) 执行算法程序所需要的时间
 B) 算法程序的长度
 C) 算法执行过程中所需要的基本运算次数
 D) 算法程序中的指令条数
- (7) 软件生命周期中所花费费用最多的阶段是
 A) 详细设计 B) 软件编码 C) 软件测试 D) 软件维护
- (8) 数据库管理系统中用来定义模式、内模式和外模式的语言为
 A) C B) Basic C) DDL D) DML
- (9) 下列有关数据库的描述,正确的是
 A) 数据库是一个DBF文件 B) 数据库是一个关系
 C) 数据库是一个结构化的数据集合 D) 数据库是一组文件
- (10) 下列有关数据库的描述,正确的是

- A) 数据处理是将信息转化为数据的过程
 B) 数据的物理独立性是指当数据的逻辑结构改变时,数据的存储结构不变
 C) 关系中的每一列称为元组,一个元组就是一个字段
 D) 如果一个关系中的属性或属性组并非该关系的关键字,但它是另一个关系的关键字,则称其为本关系的外关键字

(11) 以下选项中可作为C语言合法常量的是
 A) -80 B) -080 C) -8e1.0 D) -80.0e

(12) 以下叙述中正确的是
 A) 用C程序实现的算法必须要有输入和输出操作
 B) 用C程序实现的算法可以没有输出但必须要有输入
 C) 用C程序实现的算法可以没有输入但必须要有输出
 D) 用C程序实现的算法可以既没有输入也没有输出

(13) 下列可用于C语言用户标识符的一组是
 A) void, define, WORD B) a3_b3, _123, Car
 C) For, -abc, IF Case D) 2a, DO, sizeof

(14) 有以下程序

```
main()
{
    int i = 1, j = 1, k = 2;
    if((j++ || k++) && i++) printf("%d, %d, %d\n", i, j, k);
}

```

执行后输出结果是
 A) 1, 1, 2 B) 2, 2, 1 C) 2, 2, 2 D) 2, 2, 3

(15) 有定义语句: int b; char c[10];, 则正确的输入语句是
 A) scanf("%d%s", &b, &c); B) scanf("%d%s", &b, c);
 C) scanf("%d%s", b, c); D) scanf("%d%s", b, &c);

(16) 有以下程序:

```
main()
{
    int i, s = 1;
    for (i = 1; i < 50; i++)
        if (!(i%5) && (i%3)) s += i;
    printf("%d\n", s);
}

```

程序的输出结果是
 A) 409 B) 277 C) 1 D) 91

(17) 若运行时给变量x输入12, 则以下程序的运行结果是

```
main()
{
    int x, y;
    scanf("%d", &x);
    y = x > 12 ? x + 10 : x - 12;
    printf("%d\n", y);
}

```

- (23) 若有语句: `char *line[5]`; 以下叙述中正确的是
- A) 定义 `line` 是一个数组, 每个数组元素是一个基类型为 `char` 的指针变量
 - B) 定义 `line` 是一个指针变量, 该变量可以指向一个长度为 5 的字符型数组
 - C) 定义 `line` 是一个指针数组, 语句中的 * 号为间址运算符
 - D) 定义 `line` 是一个指向字符型函数的指针

(24) 下列程序执行后的输出结果是

```
void func(int *a, int b[])
{ b[0] = *a + 6; }
main()
{ int a, b[5];
  a = 0; b[0] = 3;
  func(&a, b); printf("%d\n", b[0]); }
```

- A) 6 B) 7 C) 8 D) 9

(25) 有如下程序

```
main()
{ char ch[2][5] = {"6937", "8254"}, *p[2];
  int i, j, s = 0;
  for(i = 0; i < 2; i++) p[i] = ch[i];
  for(i = 0; i < 2; i++)
    for(j = 0; p[i][j] > '\0'; j += 2)
      s = 10 * s + p[i][j] - '\0';
  printf("%d\n", s); }
```

- A) 69825 B) 63825 C) 6385 D) 693825

(26) 有以下程序

```
void ss(char *s, char t)
{ while(*s)
  { if(*s == t) *s = t - 'a' + 'A';
    s++; } }
main()
{ char str1[100] = "abcdffefdbd", c = 'd';
  ss(str1, c); printf("%s\n", str1); }
```

- A) ABCDEFEDBD B) abcDDffefdbd
- C) abcAAfefAbA D) Abcdffefdbd

(27) 下面程序的输出结果是

```
main()
{ int a[10] = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10}, *p = a;
```

- A) 0 B) 22 C) 12 D) 10
- (18) 已知
- ```
int t = 0;
while (t = 1)
{ ... }
```

则以下叙述正确的是

- A) 循环控制表达式的值为 0
- B) 循环控制表达式的值为 1
- C) 循环控制表达式不合法
- D) 以上说法都不对

(19) 有以下程序:

```
#include <stdio.h >
#include <string.h >
main()
{ char a[] = {'a', 'b', 'c', 'd', 'e', 'f', 'g', 'h', '\0'}; int i, j;
 i = sizeof(a); j = strlen(a);
 printf("%d, %d\n", i, j); }
```

- A) 9, 9                      B) 8, 9                      C) 1, 8                      D) 9, 8

(20) 以下能正确定义一维数组的选项是

- A) `int a[5] = {0, 1, 2, 3, 4, 5};`
- B) `char a[] = {'0', '1', '2', '3', '4', '5', '\0'};`
- C) `char a = {'A', 'B', 'C'};`
- D) `int a[5] = "0123";`

(21) 有以下程序

```
float fun(int x, int y)
{ return(x + y); }
main()
{ int a = 2, b = 5, c = 8;
 printf("%3.0f\n", fun((int)fun(a + c, b), a - c)); }
```

- A) 编译出错                  B) 9                      C) 21                      D) 9.0

(22) 下面的程序段运行后, 输出结果是

```
int i, j, x = 0;
static int a[8][8];
for (i = 0; i < 3; i++)
 for (j = 0; j < 3; j++)
 a[i][j] = 2 * i + j;
for (i = 0; i < 8; i++)
 x += a[i][j];
printf("%d", x);
```

- A) 9                      B) 不确定值                  C) 0                      D) 18

```
printf("%d\n", *(p+2));
```

A) 3                      B) 4                      C) 1                      D) 2

(28) 有如下程序:

```
#define N 2
#define M N + 1
#define NUM 2 * M + 1
main()
{ int i;
 for(i = 1; i <= NUM; i++) printf("%d\n", i); }
```

该程序中的 for 循环执行的次数是

A) 5                      B) 6                      C) 7                      D) 8

(29) 有如下程序:

```
#include <stdio.h >
union pw
{ int i;
 char ch[2]; } a;
main()
{ a.ch[0] = 13; a.ch[1] = 0; printf("%d\n", a.i); }
```

程序的输出结果是

A) 13                      B) 14                      C) 208                      D) 209

(30) 设有以下说明语句

```
typedef struct
{ int n;
 char ch[8]; } PER;
则下面叙述中正确的是

```

A) PER 是结构体变量名                      B) PER 是结构体类型名  
C) typedef struct 是结构体类型                      D) struct 是结构体类型名

(31) 下列程序的输出结果是

```
#include "stdio.h"
main()
{ int i, a = 0, b = 0;
 for(i = 1; i < 10; i++)
 { if(i%2 == 0)
 { a++;
 continue; }
 b++; }
 printf("a = %d, b = %d", a, b); }
```

A) a = 4, b = 4                      B) a = 4, b = 5                      C) a = 5, b = 4                      D) a = 5, b = 5

(32) 下面程序段的运行结果是

```
char str[] = "ABC", *p = str;
printf("%d\n", *(p+3));
```

A) 67                      B) 0  
C) 字符'C'的地址                      D) 字符'C'

(33) 设有以下说明语句

```
struct stu
{ int a;
 float b;
 stutype;
```

则下面的叙述不正确的是

A) struct 是结构体类型的关键字                      B) struct stu 是用户定义的结构体类型  
C) stutype 是用户定义的结构体类型名                      D) a 和 b 都是结构体成员名

(34) fseek 函数的正确调用形式是

A) fseek(文件指针, 起始点, 位移量)                      B) fseek(文件指针, 位移量, 起始点)  
C) fseek(位移量, 起始点, 文件指针)                      D) fseek(起始点, 位移量, 文件指针)

(35) 对于下述程序, 在方式串分别采用 "wt" 和 "wb" 运行时, 两次生成的文件 TEST 的长度是

```
#include <stdio.h >
void main()
{ FILE *fp = fopen("TEST", "w");
 fputs('A', fp); fputs('\n', fp);
 fputs('B', fp); fputs('\n', fp);
 fputs('C', fp);
 fclose(fp); }
```

A) 7 字节、7 字节                      B) 7 字节、5 字节                      C) 5 字节、7 字节                      D) 5 字节、5 字节

二、填空题(每空 2 分, 共 30 分)

请将每一个空的正确答案写在答题卡【1】~【15】序号的横线上, 答在试卷上不得分。

(1) 算法的基本特征是可行性、确定性、【1】和拥有足够的情报。  
(2) 某二叉树中度为 2 的结点有 18 个, 则该二叉树中有 【2】 个叶子结点。  
(3) 在面向对象程序设计中, 类描述的是具有相似性质的一组 【3】。  
(4) 通常, 将软件产品从提出、实现、使用维护到停止使用退役的过程称为 【4】。  
(5) 数据库管理系统常见的数据模型有层次模型、网状模型和 【5】 3 种。

(6) 在 scanf() 函数调用语句中, 可以在格式字符和 % 号之间加上一个星号, 它的作用是跳过

对应的输入数据; 当输入以下数据时, 100 200 300 400 500 < 回车 >, 下面语句的执行结果

为 【6】。

```
main()
{ int a, b, c;
```



```
scanf("%d %d *d %d %d", &a, &b, &c);
printf("%d %d %d", a, b, c); }
```

(7) 设有如下宏定义

```
#define MYSWAP(z, x, y)
{ z = x; x = y; y = z; }
```

以下程序段通过宏调用实现变量 a, b 内容交换, 请填空。

```
float a = 5, b = 16, c;
```

```
MYSWAP([7] , a, b);
```

(8) 阅读下面语句, 则程序的执行结果是 [8]。

```
#include "stdio.h"
```

```
main()
```

```
{ int a = -1, b = 1, k;
```

```
if((++a < 0) &&! (b-- <= 0))
```

```
printf("%d, %d\n", a, b);
```

```
else printf("%d, %d\n", b, a); }
```

(9) 以下程序的输出结果是 [9]。

```
main()
```

```
{ int a = 0;
```

```
a += (a = 8);
```

```
printf("%d\n", a); }
```

(10) 下列程序的输出结果是 [10]。

```
main()
```

```
{ int i;
```

```
for(i = 1; i + 1; i++)
```

```
{ if(i > 4)
```

```
{ printf("%d\n", i);
```

```
break; }
```

```
printf("%d\n", i++); }
```

(11) 函数 void fun(float \*sn, int n) 的功能是: 根据以下公式计算 s, 计算结果通过形参指针

sn 传回; n 通过形参传入, n 的值大于等于 0。请填空。

$$S = \sum_{i=0}^n \frac{(-f)^i}{2 \times i + 1}$$

```
void fun(float *sn, int n)
```

```
{ float s = 0.0, w, f = -1.0;
```

```
int i = 0;
```

```
for(i = 0; i <= n; i++)
```

```
{ f = [11] * f;
```

```
w = f / (2 * i + 1);
```

```
s += w; }
[12] = s; }
```

(12) 以下程序从输入的 10 个字符串中找出最长的那个串, 请填空。

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

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

```
#define N 10
```

```
main()
```

```
{ char str[N][81], *sp;
```

```
int i;
```

```
for(i = 0; i < N; i++) gets(str[i]);
```

```
sp = str[0];
```

```
for(i = 1; i < N; i++)
```

```
if(strlen(sp) < strlen(str[i]))
```

```
[13];
```

```
printf("输出最长的那个串:\n%s\n", sp);
```

```
printf("输出最长的那个串的长度:%d\n", strlen(sp)); }
```

(13) 下列程序执行输出的结果是 [14]。

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

```
f(int a)
```

```
{ int b = 0;
```

```
static c = 3;
```

```
a = c++, b++;
```

```
return(a); }
```

```
main()
```

```
{ int a = 2, i, k;
```

```
for(i = 0; i < 2; i++)
```

```
k = f(a++);
```

```
printf("%d\n", k); }
```

(14) 有如图所示的双链表结构, 请根据图示完成结构体的定义:

|       |      |        |
|-------|------|--------|
| lhead | data | rchild |
|-------|------|--------|

```
struct aa
```

```
{ int data;
```

```
[15] | node;
```

# 全国计算机等级考试全真模拟试题卷(2)

## 二级公共基础知识和C语言程序设计

(考试时间90分钟,满分100分)

### 一、选择题(每小题2分,共70分)

在下列各题的A)、B)、C)、D)四个选项中,只有一个选项是正确的,请将正确选项填涂在答题卡相应位置上,答在试卷上不得分。

(1)已知一棵二叉树前序遍历和中序遍历分别为 ABDEGCFH 和 DBGAEACHF,则该二叉树的后序遍历为

- A) GEDHFBCA
- B) DGEHBHFC A
- C) ABCDEFHG
- D) ACBFEDHG

(2)树是结点的集合,它的根结点数目是

- A) 有且只有1
- B) 1或多于1
- C) 0或1
- D) 至少2

(3)如先进栈序列为 e1, e2, e3, e4, 则可能的出栈序列是

- A) e3, e1, e4, e2
- B) e2, e4, e3, e1
- C) e3, e4, e1, e2
- D) 任意顺序

(4)在设计程序时,应采纳的原则之一是

- A) 不限制 goto 语句的使用
- B) 减少或取消注释行
- C) 程序越短越好
- D) 程序结构应有助于读者理解

(5)程序设计语言的基本成分是数据成分、运算成分、控制成分和

- A) 对象成分
- B) 变量成分
- C) 语句成分
- D) 传输成分

(6)下列叙述中,不属于软件需求规格说明书的作用的是

- A) 便于用户、开发人员进行理解和交流
- B) 反映出用户问题的结构,可以作为软件开发工作的基础和依据
- C) 作为确认测试和验收的依据
- D) 便于开发人员进行分析

(7)下列不属于软件工程的3个要素的是

- A) 工具
- B) 过程
- C) 方法
- D) 环境

(8)单个用户使用的数据视图的描述称为

- A) 外模式
- B) 概念模式
- C) 内模式
- D) 存储模式

(9)将 E-R 图转换到关系模式时,实体与联系都可以表示成

- A) 属性
- B) 关系
- C) 键
- D) 域

(10)SQL 语言又称为

- A) 结构化定义语言
- B) 结构化控制语言
- C) 结构化查询语言
- D) 结构化操纵语言

(11)以下不正确的叙述是

- A) 在C程序中,逗号运算符的优先级最低
- B) 在C程序中,APH 和 aph 是两个不同的变量
- C) 若 a 和 b 类型相同,在计算了赋值表达式 a = b 后 b 中的值将放入 a 中,而 b 中的值不变
- D) 当从键盘输入数据时,对于整型变量只能输入整型数值,对于实型变量只能输入实型数值

(12)请选出可用作C语言用户标识符的是

- A) void, define, WORD
- B) a3\_b3, \_123, IF
- C) FOR, --abc, Case
- D) 2a, Do, sizeof

(13)以下选项中,不能作为合法常量的是

- A) 1.234e04
- B) 1.234e0.4
- C) 1.234e+4
- D) 1.234e0

(14)已知 int a; float b; 所用的 scanf 调用语句格式为:

scanf("a//%d,b=%f",&a,&b);

为了将数据 3 和 25.08 分别赋给 x 和 y,正确的输入应当是

- A) 3,25.08 <Enter >
- B) a=3,b=25.08 <Enter >
- C) a//3,b=25.08 <Enter >
- D) a//3 <Enter > b = 25.08 <Enter >

(15)在C语言中,运算对象必须是整型数的运算符是

- A) %
- B) \
- C) % 和 \
- D) \*\*

(16)若有条件表达式 (exp)? a++:b--,则以下表达式中能完全等价于表达式(exp)的是

- A) (exp == 0)
- B) (exp != 0)
- C) (exp == 1)
- D) (exp != 1)

(17)设有以下程序段

```
int x=0,s=0;
while(!x!=0)s+=++x;
printf("%d",s);
```

则

- A) 运行程序段后输出 0
- B) 运行程序段后输出 1
- C) 程序段中的控制表达式是非法的
- D) 程序段执行无限次

(18)请选出以下语句的输出结果

```
printf("%d\n",strlen("\t" "065\xff\n"));
```

- A) 5
- B) 14
- C) 8
- D) 输出项不合法,无正常输出

(19)以下不正确的定义语句是

- A) double x[5] = {2.0,4.0,6.0,8.0,10.0};
- B) int y[5] = {0,1,3,5,7,9};
- C) char c1[] = {'1','2','3','4','5'};
- D) char c2[] = {'\x10','\xa','\x8'};

(20)设函数 fun 的定义形式为

```
void fun(char ch, float x){...}
```

则以下不正确的调用形式是

- A) fun('a', 1.23)
- B) fun(1.23, 'a')
- C) fun(1.23, 1.23)
- D) fun('a', 'a')

则以下对函数 fun 的调用语句中,正确的是

- A) fun("abc", 3.0);
- B) t = fun('D', 16.5);
- C) fun('65', 2.8);
- D) fun(32, 32);

(21) 下面判断正确的是

- A) char \* a = "china"; 等价于 char \* a; \* a = "china";
- B) char str[5] = {"china"}; 等价于 char str[] = {"china"};
- C) char \* s = "china"; 等价于 char \* s; s = "china";
- D) char c[4] = "abc", d[4] = "abc"; 等价于 char c[4] = d[4] = "abc";

(22) 若已定义:

```
int a[] = {0, 1, 2, 3, 4, 5, 6, 7, 8, 9}, * p = a, i;
```

其中  $0 \leq i \leq 9$ , 则对 a 数组元素不正确的引用是

- A) a[p - a]
- B) \*(&a[i])
- C) p[i]
- D) a[10]

(23) 以下程序的输出结果是

```
void reverse(int a[], int n)
{
 int i, t;
 for(i=0; i < n/2; i++)
 { t = a[i]; a[i] = a[n-1-i]; a[n-1-i] = t; }
}
```

main()

```
{ int b[10] = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10}; int i, s = 0;
 reverse(b, 8);
 for(i=6; i < 10; i++) s += b[i];
 printf(" %d\n", s);
}
```

- A) 22
- B) 10
- C) 34
- D) 30

(24) 阅读下列程序, 则执行后的结果为

```
#include "stdio. h"
main()
{ int c[][4] = {1, 2, 3, 4, 5, 6, 7, 34, 213, 56, 62, 3, 23, 12, 34, 56};
 printf(" %x, %x\n", c[2][2], *(*(c+1) + 1));
}
```

- A) 3e, 6
- B) 62, 5
- C) 56, 5
- D) 3E, 6

(25) 阅读下列程序, 当运行函数时, 输入 asd af aa z67, 则输出为

```
#include <stdio. h >
#include <ctype. h >
#include <string. h >
int fun (char * str)
{ int i, j = 0;
 for(i=0; str[i] != '\0'; i++)
 if(str[i] != ' ') str[j++] = str[i];
}
```

```
str[j] = '\0';
```

```
main()
{
```

```
char str[81];
```

```
int n;
```

```
printf(" Input a string : ");
```

```
gets(str);
```

```
puts(str);
```

```
fun(str);
```

```
printf(" %s\n", str);
```

```
}
```

```
A) asdafaaz67
```

```
B) asd af aa z67
```

```
C) asd
```

```
D) z67
```

(26) 以下程序的运行结果是

```
#define MIN(x,y) (x) < (y) ? (x) : (y)
```

```
main()
```

```
{ int i = 10, j = 15, k;
```

```
k = 10 * MIN(i, j);
```

```
printf(" %d\n", k);
```

```
}
```

```
A) 10
```

```
B) 15
```

```
C) 100
```

```
D) 150
```

(27) 设有以下语句:

```
char x = 3, y = 6, z;
```

```
z = x * y < < 2;
```

则 z 的二进制值是

```
A) 00010100
```

```
B) 00011011
```

```
C) 00011100
```

```
D) 00011000
```

(28) 若有以下说明和定义

```
union dt
```

```
{ int a; char b; double c; } data;
```

以下叙述中错误的是

A) data 的每个成员起始地址都相同

B) 变量 data 所占内存字节数与成员 c 所占字节数相等

C) 程序段: data. a = 5; printf(" %f\n", data. c); 输出结果为 5.000000

D) data 可以作为函数的实参

(29) 设有以下说明语句

```
typedef struct
```

```
{ int n;
```

```
char ch[8];
```

```
}; PER;
```

则下面叙述中正确的是

- A) PER 是结构体变量名
- B) PER 是结构体类型名
- C) typedef struct 是结构体类型
- D) struct 是结构体类型名

(30) 下列说法中错误的是

- A) 只能在循环体内使用 break 语句
- B) 在循环体内使用 break 语句可以使流程跳出本层循环体,从而提前结束本层循环
- C) 在 while 和 do...while 循环中,continue 语句并没有使整个循环终止
- D) continue 的作用是结束本次循环,即跳过本次循环体中尚未执行的语句,接着再进行一次进行循环判断

(31) 下面程序段的运行结果是

```
char *s = "abcde";
s += 2; printf("%d", s);
```

- A) cde
- B) 字符'c'
- C) 字符'c'的地址
- D) 无确定的输出结果

(32) 假定以下程序经编译和连接后生成可执行文件 PROG.EXE, 如果在此可执行文件所在

目录的 DOS 提示符下键入:

```
PROG ABCDEFGH IJKL <回车>, 则输出结果为
```

```
main(int argc, char * argv0[])
```

```
{ while(--argc > 0)
```

```
printf("%s", argv[argc]);
```

```
printf("\n");
```

```
}
```

A) ABCDEFG

B) IJHL

C) ABCDEFGHIJKL

D) IJKLMNOP

(33) 下面函数的功能是将指针 t2 所指向的线性链表, 链接到 t1 所指向的链表的末端。假定 t1 所

指向的链表非空

```
struct node { float x; struct node * next; };
```

```
connect(struct node * t1, struct node * t2)
```

```
{ if(t1 -> next == NULL) t1 -> next = t2;
```

```
else connect(_____, t2); }
```

要实现此功能则应该填入的选项是

- A) t1. next
- B) ++t1. next
- C) t1 -> next
- D) ++t1 -> next

(34) C 语言结构体类型变量在程序运行期间

- A) TC 环境在内存中仅仅开辟一个存放结构体变量地址的单元
- B) 所有的成员一直驻留在内存中
- C) 只有最开始的成员驻留在内存中
- D) 部分成员驻留在内存中

(35) 已知函数的调用形式: fread(buf, size, count, fp), 参数 buf 的含义是

- A) 一个整型变量, 代表要读入的数据项总数
- B) 一个文件指针, 指向要读的文件
- C) 一个指针, 指向要读入数据的存放地址
- D) 一个存储区, 存放要读的数据项

二、填空题(每空 2 分, 共 30 分)

请将每一个空的正确答案写在答题卡【1】~【15】序号的横线上, 答在试卷上不得分。

- (1) 数据结构分为逻辑结构与存储结构, 线性链表属于【1】。
- (2) 在面向对象方法中, 类之间共享属性和操作的机制称为【2】。
- (3) 耦合和内聚是评价模块独立性的两个主要标准, 其中【3】反映了模块内各成分之间的联系。
- (4) 一个项目具有一个项目主管, 一个项目主管可管理多个项目, 则实体“项目主管”与实体“项目”的联系属于【4】的联系。
- (5) 数据库设计分为以下 6 个设计阶段: 需求分析阶段、【5】、逻辑设计阶段、物理设计阶段、实施阶段、运行和维护阶段。
- (6) 如下程序片段:  
int a, b, c;  
printf("input a,b,c:");  
scanf("a=%d,b=%d,c=%d",&a,&b,&c);  
欲使输出结果为 1,2,3(分别对应于 a,b,c), 输入数据的正确形式为【6】。
- (7) 以下程序的运行结果是【7】。  
#define MAX(a,b)(a>b? a:b) + 1  
main()  
{ int i=6,j=8,k;  
printf("%d\n",MAX(i,j));  
}
- (8) 以下程序输出的结果是【8】。  
main()  
{ int a=5,b=4,c=3,d;  
d=(a>b>c);  
printf("%d\n",d);  
}
- (9) 以下程序的功能是: 删去一维数组中所有相同的数, 使之只剩一个。数组中的数已由小到大的顺序排列, 函数返回删除后数组中数据的个数。  
例如, 若一维数组中的数据是:  
2 2 2 3 4 4 5 6 6 6 6 7 7 8 9 9 10 10  
删除后, 数组中的内容应该是:  
2 3 4 5 6 7 8 9 10。  
请填写。

```

#include <stdio.h>
#define N 80
int fun(int a[], int n)
{ int i, j = 1;
 for(i = 1; i < n; i++)
 if(a[j-1] 【9】 a[i])
 a[j++] = a[i];
 【10】;
}
main()
{ int a[N] = { 2, 2, 2, 3, 4, 4, 4, 5, 6, 6, 6, 6, 7, 7, 8, 8, 9, 9, 10, 10, 10 }; i, n = 19;
 printf("The original data : \n");
 for(i = 0; i < n; i++) printf("%3d", a[i]);
 n = fun(a, n);
 printf("The data after deleted : \n");
 for(i = 0; i < n; i++) printf("%3d", a[i]); printf("\n\n");
}

```

(10) 以下程序的输出结果是 【11】。

```

fun (int x, int y, int z)
{ z = x * x + y * y; }
main ()
{ int a = 31;
 fun (6, 3, a)
 printf ("%d", a)
}

```

(11) 有下列程序, 功能是把输入的十进制长整型数以十六进制的形式输出, 完成程序

```

#include "stdio.h"
main()
{ char b[17] = {"0123456789ABCDEF"};
 int c[50], d, i = 0, base = 16;
 long n;
 scanf ("%ld", &n);
 do { c[i] = n % base; i++; n = 【12】; }
 while (n != 0);
 for (--i; i >= 0; --i)
 { d = 【13】; printf ("%c", b[d]); }
 printf ("H\n"); }

```

(12) 以下程序的输出结果是 【14】。

```

void fun()

```

```

static int a = 0;
a += 2; printf ("%d", a);
}
main()
{ int cc;
 for (cc = 1; cc < 4; cc++) fun();
 printf ("\n");
}

```

(13) 函数 my\_cmp() 的功能是比较字符串 s 和 t 的大小, 当 s 等于 t 时返回 0, 否则返回 s 和 t 的第一个不同字符的 ASCII 码差值, 即 s > t 时返回正值, 当 s < t 时返回负值。请填写空。

```

my_cmp(char *s, char *t)
{ while (*s == *t)
 { if (*s == '\0') return 0;
 ++s; ++t;
 }
 return 【15】;
}

```

密 封 线 内 不 要 答 题

# 全国计算机等级考试全真模拟试题卷(3)

## 二级公共基础知识和C语言程序设计

(考试时间 90 分钟, 满分 100 分)

### 一、选择题(每小题 2 分, 共 70 分)

在下列各题的 A)、B)、C)、D) 四个选项中, 只有一个选项是正确的, 请将正确选项填涂在答题卡相应位置上, 答在试卷上不得分。

- (1) 算法的空间复杂度是指
- A) 算法程序的长度  
B) 算法程序中的指令条数  
C) 算法程序所占的存储空间  
D) 执行算法需要的内存空间
- (2) 用链表表示线性表的优点是
- A) 便于随机存取  
B) 花费的存储空间较顺序存储少  
C) 便于插入和删除操作  
D) 数据元素的物理顺序与逻辑顺序相同
- (3) 数据结构中, 与所使用的计算机无关的是数据的
- A) 存储结构  
B) 物理结构  
C) 逻辑结构  
D) 物理和存储结构
- (4) 结构化程序设计主要强调的是
- A) 程序的规模  
B) 程序的效率  
C) 程序设计语言的先进性  
D) 程序易读性
- (5) 软件设计包括软件的结构、数据接口和过程设计, 其中软件的过程设计是指
- A) 模块间的关系  
B) 系统结构部件转换成软件的过程描述  
C) 软件层次结构  
D) 软件开发过程
- (6) 检查软件产品是否符合需求定义的过程称为
- A) 确认测试  
B) 集成测试  
C) 验证测试  
D) 验收测试
- (7) 数据流图用于抽象描述一个软件的逻辑模型, 数据流图由一些特定的图符构成。下列图符名标识的图符不属于数据流图合法图符的是
- A) 控制流  
B) 加工  
C) 数据存储  
D) 源和潭
- (8) 应用数据库的主要目的是
- A) 解决数据保密问题  
B) 解决数据完整性问题  
C) 解决数据共享问题  
D) 解决数据量大的问题
- (9) 在数据库设计中, 将 E-R 图转换成关系数据模型的过程属于
- A) 需求分析阶段  
B) 逻辑设计阶段  
C) 概念设计阶段  
D) 物理设计阶段
- (10) 在数据库管理技术的发展过程中, 经历了人工管理阶段、文件系统阶段和数据库系统阶段。其中数据独立性最高的阶段是
- A) 数据库系统  
B) 文件系统  
C) 人工管理  
D) 数据项管理
- (11) 下面四个选项中, 均是不合法的标识符的选项是
- A) A\_P\_0\_do  
B) float la0\_A  
C) b - a goto int  
D) \_123 temp int

(12) 以下 4 个选项中, 不能看作一条语句的是

- A) ;  
B) a = 5, b = 2.5, c = 3.6;  
C) if(a < 5);  
D) if(b! = 5) x = 2; y = 6;

(13) 下列说法错误的是

- A) 函数调用时, 函数名必须与所调用的函数名字完全相同  
B) 实参的个数必须与形式参数的个数一致  
C) 实参可以是表达式, 在类型上必须与形参一一对应匹配  
D) C 语言规定, 函数必须先定义, 后调用(函数的返回值类型为 int 或 char 时除外)
- (14) 有说明 char \*c[] = {"European", "Asian", "American", "African"}; 则下列叙述中正确的是

- A) \*(c+1) = 'A'  
B) c 是一个字符型指针数组, 所包含 4 个元素的初值分别为 "European", "Asian", "American" 和 "African"

C) c[3] 表示字符串 "American" 的首地址

D) c 是包含 4 个元素的字符型指针数组, 每个元素都是一个字符串的首地址

(15) 当把以下 4 个表达式用作 if 语句的控制表达式时, 有一个选项与其他 3 个选项含义不同, 这个选项是

- A) k%2  
B) k%2 == 1  
C) (k%2) != 0  
D) !k%2 == 1

(16) 若变量已正确定义并赋值, 以下符合 C 语言语法的表达式是

- A) a: = b + 1  
B) a = b = c + 2  
C) int 18.5%3  
D) a = a + 7 = c + b

(17) 以下叙述中正确的是

- A) 构成 C 程序的基本单位是函数  
B) 可以在一个函数中定义另一个函数  
C) main() 函数必须放在其他函数之前  
D) 所有被调用的函数一定要在调用之前进行定义

(18) 设 x, y, z, t 均为整型变量, 现有如下语句 x = y = z = 1; t = ++x || ++y && ++z; 则执行这个语句后 t 的值为

- A) 2  
B) 1  
C) 0  
D) 不定值

(19) 若有定义语句: int x = 3, y = 2, z = 1; 表达式: z \* \* = (x > y? ++x: y +) 的值是

- A) 4  
B) 0  
C) 1  
D) 3

(20) 以下描述中正确的是

- A) 由于 do - while 循环中循环体语句只能是一条可执行语句, 所以循环体内不能使用复合语句  
B) do - while 循环由 do 开始, 用 while 结束, 在 while(表达式) 后面不能写分号  
C) 在 do - while 循环体中, 是先执行一次循环, 再进行判断  
D) do - while 循环中, 根据情况可以省略 while

(21) 阅读如下程序段

```
#include "stdio.h"
main()
```

```

int x, a, b;
scanf("%d", &x);
a = b = x;
for(; x != 0;)
{ if(x < b) b = x;
 if(x > a) a = x;
 scanf("%d", &x); }
printf("a = %d, b = %d\n", a, b);

```

现输入如下。[注]：“ ”表示空格

34 6 23 45 5 56 7 12 365 49 48 57 87  
 7 6 7569 789 0 < Enter >

输入后输出结果是

- A) a = 7569, b = 789    B) a = 5, b = 365    C) a = 7, b = 789    D) a = 7569, b = 5

(22) 执行以下程序后, a, b 的值分别为

```

main()
{ int a, b, k = 4, m = 6, *p1 = &k, *p2 = &m;
 a = p1 == &m;
 b = (*p1) / (*p2) + 7;
 printf("a = %d\n", a);
 printf("b = %d\n", b); }

```

- A) -1, 5    B) 1, 6    C) 0, 7    D) 4, 10

(23) 下面说明不正确的是

- A) char a[10] = "china";  
 B) char \*a; a = "china";  
 C) char \*a; a = "china";  
 D) char a[10], \*p; p = a = "china"

(24) 阅读下列程序段, 则程序的输出结果为

```

#include "stdio.h"
#define M(X, Y)(X) * (Y)
#define N(X, Y)(X)/(Y)
main()
{ int a = 5, b = 6, c = 8, k;
 k = N(M(a, b), c);
 printf("%d\n", k); }

```

- A) 3    B) 5    C) 6    D) 8

(25) 若变量已正确定义, 以下 while 循环不断输出读入的字符, 当读入回车换行符时结束循环。正确的程序段是

- A) while (ch = getchar() == '\n') putchar(ch);  
 B) while (ch = getchar()) == '\n') putchar(ch);  
 C) while ((ch = getchar()) != '\n') putchar(ch);

D) while((ch = getchar()) != '\n') putchar(ch);

(26) 以下函数返回 a 所指数组中最小的值所在的下标值

```

fun(int *a, int n)
{ int i, j = 0, p;
 p = j;
 for(i = j; i < n; i++)
 if(a[i] < a[p]) _____;
 return(p); }

```

在下划线处应填入的是

- A) i = p    B) a[p] = a[i]    C) p = j    D) p = i

(27) 下面程序的运行结果是

```

#include <stdio.h>
main()
{ char a[] = "morning", t;
 int i, j = 0;
 for(i = 1; i < 7; i++) if(a[j] < a[i]) j = i;
 t = a[j]; a[j] = a[7];
 a[7] = a[j]; puts[a]; }

```

- A) mrgnir    B) mo    C) moring    D) morning

(28) 以下程序中函数 sort 的功能是对 a 数组中的数据进行由大到小的排序

```

void sort(int a[], int n)
{ int i, j, t;
 for(i = 0; i < n - 1; i++)
 for(j = i + 1; j < n; j++)
 if(a[i] < a[j]) { t = a[i]; a[i] = a[j]; a[j] = t; } }
main()
{ int aa[10] = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10}; i;
 sort(&aa[3], 5);
 for(i = 0; i < 10; i++) printf("%d, ", aa[i]);
 printf("\n"); }

```

程序运行后的输出结果是

- A) 1, 2, 3, 4, 5, 6, 7, 8, 9, 10,    B) 10, 9, 8, 7, 6, 5, 4, 3, 2, 1,  
 C) 1, 2, 3, 8, 7, 6, 5, 4, 9, 10,    D) 1, 2, 10, 9, 8, 7, 6, 5, 4, 3,

(29) 请阅读以下程序:

```

#include <stdio.h>
void fun(int s[])
{ static int j = 0;
 do
 s[j] += s[j + 1];

```

```

while(++j < 2); }
main()
{ int k, a[10] = {1, 2, 3, 4, 5};
 for(k = 1; k < 3; k++) fun(a);
 for(k = 0; k < 5; k++) printf("%d", a[k]); }

```

上面程序的输出是

- A) 34756      B) 23445      C) 35745      D) 12345

(30) 下面程序的输出结果是

```

#include <stdio.h>
#include <string.h>
main()
{ char *p1 = "abc", *p2 = "ABC", str[50] = "xyz";
 strcpy(str + 2, strcat(p1, p2));
 printf("%s\n", str); }

```

- A) xyzabcABC      B) zabcABC      C) xyabcABC      D) yzabcABC

(31) 下列程序的运行结果为

```

#include <stdio.h>
void abc(char *str)
{ int a, b;
 for(a = b = 0; str[a] != '\0'; a++)
 if(str[a] != 'c')
 str[b++] = str[a];
 str[b] = '\0'; }
void main()
{ char str[] = "abcdef";
 abc(str);
 printf("str[] = %s", str); }

```

- A) str[] = abcdef      B) str[] = abcdef  
C) str[] = a      D) str[] = ab

(32) 不合法的 main 函数命令行参数表示形式是

- A) main(int a, char \*c[])  
B) main(int arc, char \*\*arv)  
C) main(int argc, char \*argv)  
D) main(int argv, char \*argc[])

(33) 阅读下列程序, 则运行结果为

```

#include "stdio.h"
fun()
{ static int x = 5;
 x++;
 return x; }
main()

```

```

int i, x;
for (i = 0; i < 3; i++)
 x = fun();
printf("%d\n", x); }

```

- A) 5      B) 6      C) 7      D) 8

(34) 若有以下定义和语句

```

struct a
{ int n, m; };
struct a st[3] = { {1, 20}, {2, 19}, {3, 21} };
struct a *p = st;

```

则以下错误的引用是

- A) (p++)->n;      B) st[0].n;  
C) (\*p).n;      D) p = &st.m;

(35) 已知 int a = 1, b = 3 则 a<sup>b</sup> 的值为

- A) 3      B) 1      C) 2      D) 4

二、填空题 (每空 2 分, 共 30 分)

请将每一个空的正确答案写在答题卡【1】~【15】序号的横线上, 答在试卷上不得分。

(1) 数据的逻辑结构有线性结构和 【1】 两大类。

(2) 顺序存储方法是把逻辑上相邻的结点存储在物理位置 【2】 的存储单元中。

(3) 一个类可以从直接或间接的祖先中继承所有属性和方法。采用这个方法提高了软件的 【3】。

(4) 软件工程研究的内容主要包括: 【4】 技术和软件工程管理。

(5) 关系操作的特点是 【5】 操作。

(6) 现有如下定义: struct aa { int a; float b; char c; } \*p; 现需用 malloc 函数动态的申请一个 struct aa 类型大小的空间(由 p 指向), 则定义的语句为: 【6】。

(7) 下列程序的输出结果是 16.00, 请填写。

```

main()
{ int a = 9, b = 2;
 float x = 【7】, y = 1.1, z;
 z = a/2 + b * x/y + 1/2;
 printf("%5.2f\n", z); }

```

(8) 下面程序的功能是: 计算 1~10 之间的奇数之和与偶数之和, 请填写。

```

#include <stdio.h>
main()
{ int a, b, c, i;
 a = c = 0;
 for(i = 0; i <= 10; i += 2)
 { a += i;
 【8】; }

```



用方法可以是 abc. a 和 p\_abc 【12】 a。

(13) 以下程序是求矩阵 a, b 的和, 结果存入矩阵 c 中, 并按矩阵形式输出, 请填空。

```
#include <stdio.h>
main()
{
 int a[3][4] = { {3, -2, 7, 5}, {1, 0, 4, -3}, {6, 8, 0, 2} };
 int b[3][4] = { { -2, 0, 1, 4}, {5, -1, 7, 6}, {6, 8, 0, 2} };
 int i, j, c[3][4];
 for(i=0; i<3; i++)
 for(j=0; j<4; j++)
 c[i][j] = 【13】 ;
 for(i=0; i<3; i++)
 for(j=0; j<4; j++)
 printf("%3d", c[i][j]);
 printf("\n");
}
```

(14) 以下函数 rotate 的功能是: 将 a 所指 N 行 N 列的二维数组中的最后一行放到 b 所指二维数组的第 0 列中, 把 a 所指二维数组中的第 0 行放到 b 所指二维数组的最后一列中, b 所指二维数组中其他数据不变。

```
#define N 4
void rotate(int a[][N], int b[][N])
{
 int i, j;
 for(i=0; i<N; i++)
 b[i][N-1] = 【14】 ;
 【15】 = a[N-1][i];
}
```

```
c += b; }
printf("偶数之和 = %d\n", a);
printf("奇数之和 = %d\n", c - 11); }
(9) 下面程序的运行结果是 【9】。
```

```
#include <stdio.h>
#define SIZE 12
main()
{
 char s[SIZE]; int i; for(i=0; i<SIZE; i++) s[i] = 'A' + i + 321;
 sub(s, 7, SIZE - 1);
 for(i=0; i<SIZE; i++) printf("%c", s[i]);
 printf("\n"); }
sub(char * a, int t1, int t2)
{
 char ch;
 while (t1 < t2)
 ch = * (a + t1);
 * (a + t1) = * (a + t2);
 * (a + t2) = ch;
 t1++; t2--; }
void fun(char str[])
{
 int i, j, k;
 for(i=0, j = 【10】 ; i<j; i++, j--)
 k = str[i];
 str[i] = str[j];
 str[j] = k; }
(11) 阅读下面程序, 则程序的执行结果为 【11】。
```

(10) 下面函数的功能是将一个字符串的内容颠倒过来, 请填空。

```
#include "stdio.h"
fun(int k, int * p)
{
 int a, b;
 if(k == 1 || k == 2)
 * p = 1;
 else { fun(k - 1, &a);
 fun(k - 2, &b);
 * p = a + b; } }
main()
{
 int x;
 fun(6, &x);
 printf("%d\n", x); }
(12) 设有定义 " struct { int a; float b; char c; } abc, * p_abc = &abc; ", 则对结构体成员 a 的引
```