



本书附赠由**虎奔教育**提供的学习卡一张

2015年考试专用
根据教育部最新大纲编写

National Computer Rank Examination
全国计算机等级考试
无纸化真考三合一

试卷

二级C语言

■ 全国计算机等级考试命题研究室 虎奔教育教研中心 编著



手机版学习软件

题库试题，一网打尽，覆盖**99%**最新真考题库
按关键字快速查找指定试题，随时随地查看解析，**你懂的**



虎奔科举网

一学就懂，学完就会的课程
10分钟1个知识点，所学即所考
全职老师在线答疑，不懂就问
量身制定学习计划，定时催促学习进度

赠 428元
等考大礼包

手机软件
PC版软件
虎奔科举网体验班和优惠券
随身学

清华大学出版社

前 言

全国计算机等级考试（以下简称等级考试）由教育部考试中心组织，是目前报考人数较多、影响较大的全国性考试。随着教育信息化步伐的加快，等级考试也逐渐取消了笔试，全部采取无纸化的考试形式。然而，这样的变化也给广大老师的授课与考生的备考带来一定难度。

为了适应考试的变化，同时帮助广大老师和考生更好地把握新的考试内容，高效地通过计算机等级考试，本书编写组认真研究无纸化考试的考试形式和最新考试大纲，组织具有多年教学、命题、策划等经验的各方专业人士，仔细分析众多全国计算机等级考试以及其他教育产品的优点，精心策划了本套无纸化真考试卷。同时，以软件、网校、手机和现场培训等多种形式为考生提供服务。本书具有以下三大特点。

1. 百分百真题库

本书所有试题均为真实考试原题，试题类型包括选择题和上机操作题，知识点完全覆盖最新真题库，并逐年不断更新，以真题为核心组织全书的内容，同时提供考前预测试题。

2. 无纸化真考环境

在不使用计算机的情况下，本书可以完全模拟真考环境，学完本书可使考生熟悉真考中的题型分布和做题方法。

3. 全解析详细易懂

试卷中的选择题和上机题都配有答案和详细的解析。选择题解析简洁易懂，操作题解析步骤详尽，方便考生对知识点查缺补漏，综合备考。

王希更、路谨铭、李媛、王小平、张永刚、石永煊、刘爱格、戚海英、李鹏、刘欣苗等参与了本书的编写工作。

编 者

目 录

全国计算机等级考试无纸化真考试卷（1）·····	（共 8 页）
全国计算机等级考试无纸化真考试卷（2）·····	（共 8 页）
全国计算机等级考试无纸化真考试卷（3）·····	（共 8 页）
全国计算机等级考试无纸化真考试卷（4）·····	（共 8 页）
全国计算机等级考试无纸化真考试卷（5）·····	（共 8 页）
全国计算机等级考试无纸化真考试卷（6）·····	（共 8 页）
全国计算机等级考试无纸化真考试卷（7）·····	（共 8 页）
全国计算机等级考试无纸化真考试卷（8）·····	（共 8 页）
全国计算机等级考试无纸化真考试卷（9）·····	（共 8 页）
全国计算机等级考试无纸化真考试卷（10）·····	（共 8 页）
全国计算机等级考试参考答案及解析·····	（共 34 页）

前 言

全国计算机等级考试（以下简称等级考试）由教育部考试中心组织，是目前报考人数较多、影响较大的全国性考试。随着教育信息化步伐的加快，等级考试也逐渐取消了笔试，全部采取无纸化的考试形式。然而，这样的变化也给广大老师的授课与考生的备考带来一定难度。

为了适应考试的变化，同时帮助广大老师和考生更好地把握新的考试内容，高效地通过计算机等级考试，本书编写组认真研究无纸化考试的考试形式和最新考试大纲，组织具有多年教学、命题、策划等经验的各方专业人士，仔细分析众多全国计算机等级考试以及其他教育产品的优点，精心策划了本套无纸化真考试卷。同时，以软件、网校、手机和现场培训等多种形式为考生提供服务。本书具有以下三大特点。

1. 百分百真题库

本书所有试题均为真实考试原题，试题类型包括选择题和上机操作题，知识点完全覆盖最新真题库，并逐年不断更新，以真题为核心组织全书的内容，同时提供考前预测试题。

2. 无纸化真考环境

在不使用计算机的情况下，本书可以完全模拟真考环境，学完本书可使考生熟悉真考中的题型分布和做题方法。

3. 全解析详细易懂

试卷中的选择题和上机题都配有答案和详细的解析。选择题解析简洁易懂，操作题解析步骤详尽，方便考生对知识点查缺补漏，综合备考。

王希更、路谨铭、李媛、王小平、张永刚、石永煊、刘爱格、戚海英、李鹏、刘欣苗等参与了本书的编写工作。

编 者

目 录

全国计算机等级考试无纸化真考试卷（1）	（共 8 页）
全国计算机等级考试无纸化真考试卷（2）	（共 8 页）
全国计算机等级考试无纸化真考试卷（3）	（共 8 页）
全国计算机等级考试无纸化真考试卷（4）	（共 8 页）
全国计算机等级考试无纸化真考试卷（5）	（共 8 页）
全国计算机等级考试无纸化真考试卷（6）	（共 8 页）
全国计算机等级考试无纸化真考试卷（7）	（共 8 页）
全国计算机等级考试无纸化真考试卷（8）	（共 8 页）
全国计算机等级考试无纸化真考试卷（9）	（共 8 页）
全国计算机等级考试无纸化真考试卷（10）	（共 8 页）
全国计算机等级考试参考答案及解析	（共 34 页）

全国计算机等级考试无纸化真考试卷(1)

二级C语言

(考试时间120分钟, 满分100分)

一、选择题(每题1分, 共40分)

下列各题A)、B)、C)、D)四个选项中, 只有一个选项是正确的。

- (1) 程序流程图中带有箭头的线段表示的是()。
A) 图元关系 B) 数据流 C) 控制流 D) 调用关系
- (2) 结构化程序设计的基本原则不包括()。
A) 多态性 B) 自顶向下 C) 模块化 D) 逐步求精
- (3) 软件设计中模块划分应遵循的准则是()。
A) 低内聚低耦合 B) 高内聚低耦合
C) 低内聚高耦合 D) 高内聚高耦合
- (4) 在软件开发中, 需求分析阶段产生的主要文档是()。
A) 可行性分析报告 B) 软件需求规格说明书
C) 概要设计说明书 D) 集成测试计划
- (5) 算法的有穷性是指()。
A) 算法程序的运行时间是有限的
B) 算法程序所处理的数据量是有限的
C) 算法程序的长度是有限的
D) 算法只能被有限的用户使用
- (6) 对长度为n的线性表排序, 在最坏情况下, 比较次数不是 $n(n-1)/2$ 的排序方法是()。
A) 快速排序 B) 冒泡排序 C) 直接插入排序 D) 堆排序
- (7) 下列关于栈的叙述正确的是()。
A) 栈按“先进先出”组织数据 B) 栈按“先进后出”组织数据
C) 只能在栈底插入数据 D) 不能删除数据
- (8) 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) 概念设计阶段 C) 逻辑设计阶段 D) 物理设计阶段
- (9) 有三个关系R、S和T如下:

R		
B	C	D
a	0	k1
b	1	n1

S		
B	C	D
f	3	h2
a	0	k1
n	2	x1

T		
B	C	D
a	0	k1

由关系R和S通过运算得到关系T, 所使用的运算为()。

- A) 并 B) 自然连接 C) 笛卡儿积 D) 交
- (10) 设有表示学生选课的一张表, 学生S(学号, 姓名, 性别, 年龄, 身份证号), 课程C(课号, 课名), 选课SC(学号, 课号, 成绩), 则表SC的关键字(键或码)为()。
A) 课号, 成绩 B) 学号, 成绩
C) 学号, 课号 D) 学号, 姓名, 成绩
- (11) 下列叙述中错误的是()。
A) 一个C语言程序只能实现一种算法
B) C程序可以由多个程序文件组成
C) C程序可以由一个或多个函数组成
D) 一个C函数可以单独作为一个C程序文件存在
- (12) 以下选项中, 能用作数据常量的是()。
A) 115L B) 0118 C) 1.5e1.5 D) o115

- (13) 按照C语言规定的用户标识符命名规则, 不能出现在标识符中的是()。
A) 下划线 B) 大写字母 C) 数字字符 D) 连接符
- (14) 设变量已正确定义并赋值, 以下正确的表达式是()。
A) $x=y+z+5, ++y$ B) $\text{int}(15.8\%5)$ C) $x=y*5=x+z$ D) $x=25\%5.0$
- (15) 设有定义“int x=2;”, 以下表达式中, 值不为6的是()。
A) $x++, 2*x$ B) $2*x, x+=2$ C) $x*=(1+x)$ D) $x*=x+1$

(16) 有以下程序

```
#include <stdio.h>
main()
{
    int x, y, z;
    x=y=1;
    z=x++, y++, ++y;
    printf("%d,%d,%d\n", x, y, z);
}
```

程序运行后的输出结果是()。

- A) 2,3,1 B) 2,3,2 C) 2,3,3 D) 2,2,1

(17) 有以下程序

```
#include <stdio.h>
main()
{
    char c1, c2;
    c1='A'+8-'4';
    c2='A'+8-'5';
    printf("%c,%c\n", c1, c2);
}
```

已知字母A的ASCII码为65, 程序运行后的输出结果是()。

- A) E,D B) D,69 C) E,68 D) 输出无定值
- (18) 若有定义“int x,y;”并已正确给变量赋值, 则以下选项中与表达式“(x-y)?(x++):(y++)”中的条件表达式“(x-y)”等价的是()。
A) $(x-y<0||x-y>0)$ B) $(x-y<0)$ C) $(x-y>0)$ D) $(x-y==0)$

(19) 有以下程序

```
#include <stdio.h>
main()
{
    int x=1, y=0;
    if (!x) y++;
    else if (x==0)
        if (x) y+=2;
        else y+=3;
    printf("%d\n", y);
}
```

程序运行后的输出结果是()。

- A) 3 B) 2 C) 1 D) 0
- (20) 若有定义“float x=1.5; int a=1,b=3,c=2;”, 则正确的switch语句是()。
A) switch(a+b) B) switch((int)x);

- ```
 { case 1: printf("**\n");
 case 2+1: printf("**\n"); }
C) switch(x)
 { case 1.0: printf("**\n");
 case 2.0: printf("**\n"); }
```
- (21) 有以下程序
- ```
#include <stdio.h>
main()
{ int y=9;
  for( ; y>0; y--)
    if(y%3==0) printf("%d", --y);
}
```
- 程序的运行结果是()。
- A) 852 B) 963 C) 741 D) 875421
- (22) 有以下程序
- ```
#include <stdio.h>
main()
{ int i,j,m=1;
 for(i=1;i<3;i++)
 { for(j=3;j>0;j--)
 { if(i*j>3) break;
 m*=i*j;
 }
 }
 printf("m=%d\n",m);
}
```
- 程序运行后的输出结果是( )。
- A) m=4                      B) m=2                      C) m=6                      D) m=5
- (23) 有以下程序
- ```
#include <stdio.h>
int fun (int x, int y)
{ if (x!=y) return( (x+y) /2 );
  else return( x );
}
main()
{ int a=4, b=5, c=6;
  printf( "%d\n", fun(2*a, fun( b, c )));
}
```
- 程序运行后的输出结果是()。
- A) 6 B) 3 C) 8 D) 12
- (24) 有以下程序

```
    { case 1: printf("**\n");
      case 2: printf("**\n"); }
D) switch(a+b)
    { case 1: printf("**\n");
      case c: printf("**\n"); }
```

```
#include <stdio.h>
int add( int a,intB){ return(a+b); }
main()
{ int k, (*f)( ), a=5,b=10;
  f=add;
  ...
}
```

则以下函数调用语句错误的是()。

- A) k=f(a,b); B) k=add(a,b); C) k=(f)(a,b); D) k= *f(a,b);

(25) 有以下程序

```
#include <stdio.h>
main()
{ int n,*p=NULL;
  *p=&n;
  printf("Input n:"); scanf("%d",&p); printf("output n:"); printf("%d\n",p);
}
```

该程序试图通过指针p为变量n读入数据并输出, 但程序有多处错误, 以下语句正确的是()。

- A) int n,*p=NULL; B) *p=&n; C) scanf("%d",&p) D) printf("%d\n",p);

(26) 下列定义数组的语句中, 正确的是()。

- A) int N 10 B) #define N=10; C) int x[N]; D) int x[0..10];
- C) int x[0..10]; D) int x[];

(27) 有以下程序

```
#include <stdio.h>
main()
{ int a[]={2,3,5,4},i;
  for(i=0;i<4;i++)
    switch(i%2)
    { case 0: switch(a[i]%2)
      { case 0:a[i]++;break;
        case 1:a[i]--;
      }break;
      case 1:a[i]=0;
    }
  for(i=0;i<4;i++) printf("%d ",a[i]); printf("\n");
}
```

程序运行后的输出结果是()。

- A) 3 0 4 0 B) 2 0 5 0 C) 3 3 4 4 D) 0 3 0 4

(28) 有以下程序

```
#include <stdio.h>
main()
{ int b[3][3]={0,1,2,0,1,2,0,1,2},i,j,t=1;
```

```

for(i=0; i<3; i++)
    for(j=i; j<=i; j++) t+=b[i][b[j][i]];
printf("%d\n", t);
}

```

程序运行后的输出结果是()。

- A) 3 B) 4 C) 1 D) 9

(29) 以下语句中存在语法错误的是()。

- A) char ss[6][20]; ss[1]= "right?"; B) char ss[][20]={ "right?";
C) char *ss[6]; ss[1]= "right?"; D) char *ss[]={ "right?";

(30) 以下不能将s所指字符串正确复制到t所指存储空间的是()。

- A) while(*t==*s){t++;s++;} B) for(i=0; t[i]=s[i]; i++);
C) do{t++=*s++;} while(*s); D) for(i=0, j=0; t[i++]=s[j++];);

(31) 有以下程序

```

#include <stdio.h>
void swap(char *x, char *y)
{
    char t;
    t=*x; *x=*y; *y=t;
}
main()
{
    char *s1="abc", *s2="123";
    swap(s1, s2); printf("%s,%s\n", s1, s2);
}

```

程序执行后的输出结果是()。

- A) 1bc,a23 B) abc,123 C) 123,abc D) 321,cba

(32) 有以下函数

```

int fun(char *x, char *y)
{
    int n=0;
    while((*x==*y) && *x!='\0') {x++; y++; n++;}
    return n;
}

```

函数的功能是()。

- A) 统计x和y所指字符串中最前面连续相同的字符个数
B) 查找x和y所指字符串中是否有'\0'
C) 将y所指字符串赋给x所指存储空间
D) 统计x和y所指字符串中相同的字符个数

(33) 有以下程序

```

#include <stdio.h>
int fun()
{
    static int x=1;
    x*=2;
    return x;
}

```

```

main()
{
    int i, s=1;
    for( i=1; i<=3; i++) s*=fun();
    printf("%d\n", s);
}

```

程序运行后的输出结果是()。

- A) 30 B) 10 C) 0 D) 64

(34) 在一个C源程序文件中所定义的全局变量, 其作用域为()。

- A) 所在文件的全部范围
B) 所在程序的全部范围
C) 所在函数的全部范围
D) 由具体定义位置和extern说明来决定范围

(35) 以下叙述中错误的是()。

- A) 用typedef定义新的类型名后, 原有类型名仍有效
B) 可以用typedef将已存在的类型用一个新的名字来代表
C) 可以通过typedef增加新的类型
D) 用typedef可以为各种类型起别名, 但不能为变量起别名

(36) 有以下程序

```

#include <stdio.h>
struct S
{
    int a, b; } data[2]={10,100,20,200 };
main()
{
    struct S p=data[1];
    printf("%d\n", ++(p.A));
}

```

程序运行后的输出结果是()。

- A) 21 B) 11 C) 20 D) 10

(37) 有以下程序

```

#include <stdio.h>
#define PT 3.5;
#define S(x) PT*x*x;
main()
{
    int a=1, b=2; printf("%4.1f\n", S(a+b)); }

```

程序运行后的输出结果是()。

- A) 31.5 B) 程序有错无输出结果
C) 7.5 D) 14.0

(38) 有以下程序

```

#include <stdio.h>
main()
{
    unsigned char a=2, b=4, c=5, d;
    d = a | b; d &= c; printf("%d\n", d); }

```

程序运行后的输出结果是()。

- A) 4 B) 3 C) 5 D) 6

(39) 有以下程序

```
#include <stdio.h>
#include <stdlib.h>
main()
{   int *a,*b,*c;
    a=b=c=(int *)malloc(sizeof(int));
    *a=1;*b=2,*c=3;
    a=b;
    printf("%d,%d,%d\n",*a,*b,*c);
}
```

程序运行后的输出结果是()。

- A) 1,1,3 B) 2,2,3 C) 1,2,3 D) 3,3,3

(40) 以下叙述中正确的是()。

- A) 当对文件的读(写)操作完成之后,必须将它关闭,否则可能导致数据丢失
B) 打开一个已存在的文件并进行了写操作后,原有文件中的全部数据必定被覆盖
C) 在一个程序中当对文件进行了写操作后,必须先关闭该文件然后再打开,才能读到第1个数据
D) C语言中的文件是流式文件,因此只能顺序存取数据

二、程序填空题

给定程序中,函数 fun 的功能是:找出 100 至 x ($x \leq 999$) 之间各位上的数字之和为 15 的所有整数,然后输出;符合条件的整数个数作为函数值返回。

例如,当 x 值为 500 时,各位数字之和为 15 的整数有:159、168、177、186、195、249、258、267、276、285、294、339、348、357、366、375、384、393、429、438、447、456、465、474、483、492,共有 26 个。

请在程序的下划线处填入正确的内容并把下划线删除,使程序得出正确的结果。

注意:源程序存放在考生文件夹下的 BLANK1.C 中。

不得增行或删行,也不得更改程序的结构!

给定源程序:

```
#include <stdio.h>
int fun(int x)
{   int n, s1, s2, s3, t;
    /*****found*****/
    n= 1;
    t=100;
    /*****found*****/
    while(t<= 2)
    {   s1=t%10; s2=(t/10)%10; s3=t/100;
        if(s1+s2+s3==15)
        {   printf("%d ",t);
            n++;
        }
    }
    /*****found*****/
    3;
    return n;
}
main()
{   int x=-1;
    while(x>999||x<0)
```

```
{   printf("Please input(0<x<=999): ");
    scanf("%d",&x); }
printf("\nThe result is: %d\n",fun(x));
}
```

三、程序改错题

给定程序 MOD11.C 中函数 fun 的功能是:先将 s 所指字符串中的字符按逆序存放到 t 所指字符串中,然后把 s 所指串中的字符按正序连接到 t 所指串的后面。

例如:当 s 所指的字符串为:“ABCDE”时,则 t 所指的字符串应为:“EDCBAABCDE”。

请改正程序中的错误,使它能得出正确的结果。

注意:不要改动 main 函数,不得增行或删行,也不得更改程序的结构!

给定源程序:

```
#include <stdio.h>
#include <string.h>
void fun(char *s, char *t)
{
    /*****found*****/
    int i;
    i=0;
    sl = strlen(s);
    for (; i<sl; i++)
    /*****found*****/
        t[i] = s[sl-i];
    for (i=0; i<sl; i++)
        t[sl+i] = s[i];
    t[2*sl] = '\0';
}
main()
{   char s[100], t[100];
    printf("\nPlease enter string s:"); scanf("%s", s);
    fun(s, t);
    printf("The result is: %s\n", t);
}
```

四、程序设计题

函数 fun 的功能是:将 a、b 中的两个两位正整数合并形成一个新的整数放在 c 中。合并的方式是:将 a 中的十位和个位数依次放在变量 c 的百位和个位上, b 中的十位和个位数依次放在变量 c 的千位和十位上。

例如,当 a=45, b=12, 调用该函数后, c=1425。

注意:部分源程序存在文件 PROG1.C 中。数据文件 IN.DAT 中的数据不得修改。

请勿改动主函数 main 和其他函数中的任何内容,仅在函数 fun 的花括号中填入你编写的若干语句。

给定源程序:

```
#include <stdio.h>
void fun(int a, int b, long *c)
{
}
main() /* 主函数 */
{   int a,b; long c;
    printf("Input a b:");
    scanf("%d%d", &a, &b);
    fun(a, b, &c);
    printf("The result is: %ld\n", c);
}
```

全国计算机等级考试无纸化真考试卷(2)

二级C语言

(考试时间120分钟, 满分100分)

一、选择题(每题1分, 共40分)

下列各题A)、B)、C)、D)四个选项中, 只有一个选项是正确的。

- (1) 一个栈的初始状态为空。现将元素1、2、3、4、5、A、B、C、D、E依次入栈, 然后再依次出栈, 则元素出栈的顺序是()。
A) 12345ABCDE B) EDCBA54321 C) ABCDE12345 D) 54321EDCBA
- (2) 下列叙述中正确的是()。
A) 循环队列有队头和队尾两个指针, 因此, 循环队列是非线性结构
B) 在循环队列中, 只需要队头指针就能反映队列中元素的动态变化情况
C) 在循环队列中, 只需要队尾指针就能反映队列中元素的动态变化情况
D) 循环队列中元素的个数由队头指针和队尾指针共同决定
- (3) 在长度为n的有序线性表中进行二分法查找, 最坏情况下需要比较的次数是()。
A) $O(n)$ B) $O(n^2)$ C) $O(\log_2 n)$ D) $O(n\log_2 n)$
- (4) 下列叙述中正确的是()。
A) 顺序存储结构的存储空间一定是连续的, 链式存储结构的存储空间不一定是连续的
B) 顺序存储结构只针对线性结构, 链式存储结构只针对非线性结构
C) 顺序存储结构能存储有序表, 链式存储结构不能存储有序表
D) 链式存储结构比顺序存储结构节省存储空间
- (5) 数据流图中带有箭头的线段表示的是()。
A) 控制流 B) 事件驱动 C) 模块调用 D) 数据流
- (6) 在软件开发中, 需求分析阶段可以使用的工具是()。
A) N-S图 B) DFD图 C) PAD图 D) 程序流程图
- (7) 在面向对象方法中, 不属于“对象”基本特点的是()。
A) 一致性 B) 分类性 C) 多态性 D) 标识唯一性
- (8) 一间宿舍可住多名学生, 则实体宿舍和学生之间的联系是()。
A) 一对一 B) 一对多 C) 多对一 D) 多对多
- (9) 在数据管理技术发展的三个阶段中, 数据共享最好的是()。
A) 人工管理阶段 B) 文件系统阶段 C) 数据库系统阶段 D) 三个阶段相同
- (10) 有三个关系R、S和T如下:

R		S		T		
A	B	B	C	A	B	C
m	1	1	3	m	1	3
n	2	3	5			

由关系R和S通过运算得到关系T, 则所使用的运算为()。

- A) 笛卡儿积 B) 交 C) 并 D) 自然连接
- (11) 以下叙述中错误的是()。
A) 使用三种基本结构构成的程序只能解决简单问题
B) 结构化程序由顺序、分支、循环三种基本结构组成
C) C语言是一种结构化程序设计语言
D) 结构化程序设计提倡模块化的设计方法
 - (12) 以下四个程序中, 完全正确的是()。

A)

```
#include <stdio.h>
main();
{ /* programming */
printf("programming!\n"); }
```

B)

```
#include <stdio.h>
main()
{ /* programming */
printf("programming!\n"); }
```

C)

```
#include <stdio.h>
main()
{ /* programming */
printf("programming!\n"); }
```

D)

```
include <stdio.h>
main()
{ /* programming */
printf("programming!\n"); }
```

- (13) C源程序中不能表示的数制是()。
A) 十进制 B) 八进制 C) 二进制 D) 十六进制
- (14) 以下选项中, 能用作用户标识符的是()。
A) _0_ B) 8_8 C) void D) unsigned
- (15) 若有定义语句“int x=10;”, 则表达式“x-=x+x”的值为()。
A) 10 B) -20 C) 0 D) -10
- (16) 有以下程序

```
#include <stdio.h>
main()
{ int a=1,b=0;
printf("%d,",b=a+b);
printf("%d\n",a=2*b);
}
```

程序运行后的输出结果是()。

- A) 1,2 B) 1,0 C) 3,2 D) 0,0

- (17) 有以下程序

```
#include <stdio.h>
main()
{ int a1,a2; char c1,c2;
scanf("%d%c%d%c",&a1,&c1,&a2,&c2);
printf("%d,%c,%d,%c",a1,c1,a2,c2);
}
```

若想通过键盘输入, 使得a1的值为12, a2的值为34, c1的值为字符a, c2的值为字符b, 程序输出结果是12,a,34,b 则正确的输入格式是() (□代表空格字符)。

- A) 12□a□34□b<CR> B) 12a34b<CR>
C) 12,a,34,b<CR> D) 12□a34□b<CR>

- (18) 若变量已正确定义, 在“if(W) printf("%d\n",k);”中, 以下不可替代W的是()。

A) a==b+c B) ch=getchar() C) a<>b+c D) a++

- (19) 有以下程序段

```
#include <stdio.h>
main()
{
int a,b,c;
a=10; b=50; c=30;
if (a>b) a=b; b=c; c=a;
printf("a=%d b=%d c=%d\n", a, b, c);
}
```

程序的输出结果是()。

- A) a=10 b=50 c=10 B) a=10 b=50 c=30 C) a=10 b=30 c=10 D) a=50 b=30 c=50

- (20) 下列叙述中正确的是()。

A) 在switch语句中, 不一定使用break语句
B) 在switch语句中, 必须使用default

- C) break语句必须与switch语句中的case配对使用
D) break语句只能用于switch语句
- (21) 以下不构成无限循环的语句或语句组是()。
- A) n=0; while(1) { n++; }
B) n=0; do { ++n; } while(n<=0);
C) n=10; while(n); { n--; }
D) for(n=0,i=1; ;i++) n+=i;
- (22) 有以下程序
- ```
#include <stdio.h>
main()
{ int c=0,k;
 for(k=1;k<3;k++)
 switch(k)
 { default: c+=k;
 case 2: c++;break;
 case 4: c+=2;break;
 }
 printf("%d\n",c);
}
```
- 程序运行后的输出结果是( )。
- A) 3 B) 5 C) 7 D) 9
- (23) 有以下程序
- ```
#include <stdio.h>
int f(int x,int y)
{ return((y-x)*x); }
main()
{ int a=3,b=4,c=5,d;
  d=f(f(a,b),f(a,c));
  printf("%d\n",d);
}
```
- 程序运行后的输出结果是()。
- A) 8 B) 10 C) 9 D) 7
- (24) 若有定义语句“double a,*p=&a;”,以下叙述中错误的是()。
- A) 定义语句中,“*p=&a”是把变量a的地址作为初值赋给指针变量p
B) 定义语句中的*号是一个说明符
C) 定义语句中的p只能存放double类型变量的地址
D) 定义语句中的*号是一个间址运算符
- (25) 若有定义语句“double x,y,*px,*py;”,执行了“px=&x; py=&y;”之后,正确的输入语句是()。
- A) scanf("%lf %le", px, py);
B) scanf("%f %f" &x, &y);
C) scanf("%f %f", x, y);
D) scanf("%lf %lf", x, y);
- (26) 以下定义数组的语句中错误的是()。
- A) int num[]={ 1,2,3,4,5,6 };
B) int num[][3]={ {1,2},3,4,5,6 };
C) int num[2][4]={ {1,2},{3,4},{5,6} };
D) int num[][4]={1,2,3,4,5,6};
- (27) 有以下程序
- ```
#include <stdio.h>
void fun(int a, int b)
```

- ```
{ int t;
  t=a; a=b; b=t;
}
main()
{ int c[10]={1,2,3,4,5,6,7,8,9,0}, i;
  for(i=0; i<10; i+=2) fun(c[i], c[i+1]);
  for(i=0; i<10; i++) printf("%d,", c[i]);
  printf("\n");
}
```
- 程序的运行结果是()。
- A) 1,2,3,4,5,6,7,8,9,0, B) 2,1,4,3,6,5,8,7,0,9,
C) 0,9,8,7,6,5,4,3,2,1, D) 0,1,2,3,4,5,6,7,8,9,
- (28) 有以下程序
- ```
#include <stdio.h>
main()
{ int x[3][2]={0}, i;
 for(i=0; i<3; i++) scanf("%d",x[i]);
 printf("%3d%3d%3d\n",x[0][0],x[0][1],x[1][0]);
}
```
- 若运行时输入: 2 4 6<回车>, 则输出结果为( )。
- A) 2 0 0 B) 2 0 4 C) 2 4 0 D) 2 4 6
- (29) 有以下程序段
- ```
#include <stdio.h>
int j; float y; char name[50];
scanf("%2d%f%s", &j, &y, name);
```
- 当执行上述程序段,从键盘上输入55566 7777123后,y的值为()。
- A) 566.0 B) 55566.0 C) 7777.0 D) 566777.0
- (30) 下列语句组中,正确的是()。
- A) char *s;s="Olympic";
B) char s[7];s="Olympic";
C) char *s;s={"Olympic"};
D) char s[7];s={"Olympic"};
- (31) 有以下函数
- ```
int fun(char *s)
{ char *t=s;
 while(*t++);
 return(t-s);
}
```
- 该函数的功能是( )。
- A) 计算s所指字符串占用内存字节的个数 B) 比较两个字符串的大小  
C) 计算s所指字符串的长度 D) 将s所指字符串复制到字符串t中
- (32) 有以下程序(字符a的ASCII码值为97)
- ```
#include <stdio.h>
main()
{ char *s={"abc"};
  do
  { printf("%d", *s%10); ++s; }
```

密
封
线
内
不
要
答
题

```
while( *s );  
}
```

程序运行后的输出结果是()。

- A) abc B) 789 C) 7890 D) 979899

(33) 设有如下函数定义

```
#include <stdio.h>  
int fun(int k)  
{  
    if(k<1) return 0;  
    else if(k==1) return 1;  
    else return fun(k-1)+1;  
}
```

若执行调用语句“n=fun(3);”, 则函数fun总共被调用的次数是()。

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

(34) 有以下程序

```
#include <stdio.h>  
int f(int n);  
main()  
{  
    int a=3,s;  
    s=f(A);s=s+f(A); printf("%d\n",s);  
}  
int f(int n)  
{  
    static int a=1;  
    n+=a++;  
    return n;  
}
```

程序运行后的输出结果是()。

- A) 9 B) 8 C) 7 D) 10

(35) 设有定义:

```
struct complex  
{  
    int real,unreal; } data1={1,8},data2;  
则以下赋值语句中错误的是( )。
```

- A) data2.real=data1.unreal; B) data2=data1;
C) data2.real=data1.real; D) data2=(2,6);

(36) 有以下程序

```
#include <stdio.h>  
struct S{ int n; int a[20]; };  
void f(struct S *p)  
{  
    int i,j,t;  
    for(i=0; i<p->n-1;i++)  
        for(j=i+1; j<p->n;j++)  
            if (p->a[i] > p->a[j]) { t= p->a[i]; p->a[i]= p->a[j]; p->a[j]= t; }  
}  
main()  
{  
    int i; struct S s={10, {2,3,1,6,8,7,5,4,10,9}};  
    f(&s);  
    for(i=0; i<s.n;i++) printf("%d,", s.a[i]);  
}
```

程序运行后的输出结果是()。

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

(37) 有以下程序

```
#include <stdio.h>  
#include <string.h>  
typedef struct { char name[9]; char sex; int score[2]; } STU;  
STU f (STU A)  
{  
    STU b={"Zhao", 'm', 85, 90};  
    int i;  
    strcpy( a.name, b.name );  
    a.sex = b.sex;  
    for( i=0; i<2; i++ ) a.score[i] = b.score[i];  
    return a;  
}  
main()  
{  
    STU c= { "Qian", 'f', 95, 92 }, d;  
    d = f(c);  
    printf ("%s,%c,%d,%d, ", d.name, d.sex, d.score[0], d.score[1]);  
    printf ("%s,%c,%d,%d\n", c.name, c.sex, c.score[0], c.score[1]);  
}
```

程序运行后的输出结果是()。

- A) Zhao,m,85,90, Qian,f,95,92 B) Zhao,m,85,90, Zhao,m,85,90
C) Qian,f,95,92, Qian,f,95,92 D) Qian,f,95,92, Zhao,m,85,90

(38) 以下关于宏的叙述中正确的是()。

- A) 宏替换没有数据类型限制 B) 宏定义必须位于源程序中所有语句之前
C) 宏名必须用大写字母表示 D) 宏调用比函数调用耗费时间

(39) 设有以下语句

```
int a=1, b=2, c;  
c=a^(b<<2);  
执行后, c的值为( )。
```

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

(40) 有以下程序

```
#include <stdio.h>  
main()  
{  
    FILE *fp; int a[10]={1,2,3}, i, n;  
    fp = fopen("d1.dat", "w");  
    for(i=0; i<3;i++) fprintf(fp, "%d", a[i]);  
    fprintf(fp, "\n");  
    fclose(fp);  
    fp = fopen("d1.dat", "r");  
    fscanf(fp, "%d", &n);  
    fclose(fp);  
}
```

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

程序的运行结果是()。

A) 123 B) 12300 C) 1 D) 321

二、程序填空题

给定程序中，函数 fun 的功能是：找出 100 ~ 999 之间（含 100 和 999）所有整数中各位上数字之和为 x（x 为一正整数）的整数，然后输出；符合条件的整数个数作为函数值返回。

例如，当 x 值为 5 时，100 ~ 999 之间各位上数字之和为 5 的整数有：104、113、122、131、140、203、212、221、230、302、311、320、401、410、500。共有 15 个。当 x 值为 27 时，各位数字之和为 27 的整数是 999。只有 1 个。

请在程序的下划线处填入正确的内容并把下划线删除，使程序得出正确的结果。

注意：源程序存放在考生文件夹下的 BLANK1.C 中。

不得增行或删行，也不得更改程序的结构！

给定源程序：

```
#include <stdio.h>
int fun(int x)
{   int n, s1, s2, s3, t;
    n=0;
    t=100;
    /*****found*****/
    while(t<= 1){
    /*****found*****/
        s1=t%10; s2=( 2 )%10; s3=t/100;
    /*****found*****/
        if(s1+s2+s3== 3 )
        {   printf("%d ",t);
            n++;
        }
        t++;
    }
    return n;
}
main()
{   int x=-1;
    while(x<0)
    {   printf("Please input(x<0): ");
        scanf("%d",&x); }
    printf("\nThe result is: %d\n",fun(x));
}
```

三、程序改错题

给定程序 MOD11.C 中函数 fun 的功能是：从低位开始取出长整型变量 s 中偶数位上的数，依次构成一个新数放在 t 中。高位仍在高位，低位仍在低位。

例如，当 s 中的数为 7654321 时，t 中的数为 642。

请改正程序中的错误，使它得出正确的结果。

注意：不要改动 main 函数，不得增行或删行，也不得更改程序的结构！

给定源程序：

```
#include <stdio.h>
/*****found*****/
void fun (long s, long t)
{   long sl=10;
    s /= 10;
    *t = s % 10;
    /*****found*****/
    while ( s < 0)
    {   s = s/100;
```

```
*t = s%10*sl + *t;
sl = sl*10;
```

```
    }
}
main()
{   long s, t;
    printf("\nPlease enter s:"); scanf("%ld", &s);
    fun(s, &t);
    printf("The result is: %ld\n", t);
}
```

四、程序设计题

学生的记录由学号和成绩组成，N 名学生的数据已在主函数中放入结构体数组 s 中，请编写函数 fun，它的功能是：按分数的高低排列学生的记录，高分在前。

注意：部分源程序存在文件 PROG1.C 中。

请勿改动主函数 main 和其他函数中的任何内容，仅在函数 fun 的花括号中填入你编写的若干语句。

给定源程序：

```
#include <stdio.h>
#define N 16
typedef struct
{   char num[10];
    int s;
} STREC;
void fun( STREC a[])
{
    STREC tmp;
    int i, j;
    for(i = 0; i < N; i++)
        for(j = i+1; j < N; j++)
            /* 请按题目要求完成以下代码 */

}
main()
{   STREC s[N]={{"GA005",85},{"GA003",76},
                {"GA002",69},{"GA004",85},{"GA001",91},
                {"GA007",72},{"GA008",64},{"GA006",87},
                {"GA015",85},{"GA013",91},{"GA012",64},
                {"GA014",91}, {"GA011",66},{"GA017",64},
                {"GA018",64},{"GA016",72}};
    int i; FILE *out;
    fun( s );
    printf("The data after sorted :\n");
    for(i=0;i<N; i++)
    {   if( (i)%4==0 )printf("\n");
        printf("%s %4d ",s[i].num,s[i].s);
    }
    printf("\n");
    out = fopen("c:\\test\\out.dat","w");
    for(i=0;i<N; i++)
    {   if( (i)%4==0 && i ) fprintf(out, "\n");
        fprintf(out, "%4d ",s[i].s);
    }
    fprintf(out, "\n");
    fclose(out);
}
```

全国计算机等级考试无纸化真考试卷(3)

二级C语言

(考试时间120分钟, 满分100分)

一、选择题(每题1分, 共40分)

下列各题A)、B)、C)、D)四个选项中, 只有一个选项是正确的。

- (1) 下列叙述中正确的是()。
 - A) 栈是“先进先出”的线性表
 - B) 队列是“先进后出”的线性表
 - C) 循环队列是非线性结构的线性表
 - D) 有序线性表既可以采用顺序存储结构, 也可以采用链式存储结构
- (2) 支持子程序调用的数据结构是()。
 - A) 栈
 - B) 树
 - C) 队列
 - D) 二叉树
- (3) 某二叉树有5个度为2的结点, 则该二叉树中的叶子结点数是()。
 - A) 10
 - B) 8
 - C) 6
 - D) 4
- (4) 下列排序方法中, 最坏情况下比较次数最少的是()。
 - A) 冒泡排序
 - B) 简单选择排序
 - C) 直接插入排序
 - D) 堆排序
- (5) 软件按功能可以分为应用软件、系统软件和支撑软件(工具软件), 下面属于应用软件的是()。
 - A) 编译程序
 - B) 操作系统
 - C) 教务管理系统
 - D) 汇编程序
- (6) 下面叙述中错误的是()。
 - A) 软件测试的目的是发现错误并改正错误
 - B) 对被调试的程序进行“错误定位”是程序调试的必要步骤
 - C) 程序调试通常也称为Debug
 - D) 软件测试应严格执行测试计划, 排除测试的随意性
- (7) 耦合性和内聚性是对模块独立性度量的两个标准, 下列叙述中正确的是()。
 - A) 提高耦合性降低内聚性有利于提高模块的独立性
 - B) 降低耦合性提高内聚性有利于提高模块的独立性
 - C) 耦合性是指一个模块内部各个元素间彼此结合的紧密程度
 - D) 内聚性是指模块间互相连接的紧密程度
- (8) 下列叙述中错误的是()。
 - A) 在数据库系统中, 数据的物理结构必须与逻辑结构一致
 - B) 数据库技术的根本目标是要解决数据的共享问题
 - C) 数据库设计是指在已有数据库管理系统的基础上建立数据库
 - D) 数据库系统需要操作系统的支持
- (9) 有两个关系R, S如下:

R		
A	B	C
a	3	2
b	0	1
c	2	1

S	
A	B
a	3
b	0
c	2

由关系R通过运算得到关系S, 则所使用的运算为()。

- A) 选择
 - B) 投影
 - C) 插入
 - D) 连接
- (10) 将E-R图转换为关系模式时, 实体和联系都可以表示为()。
- A) 属性
 - B) 键
 - C) 关系
 - D) 域
- (11) 以下叙述中错误的是()。
- A) C语言中的每条可执行语句和非执行语句最终都将被转换成二进制的机器指令
 - B) C程序经过编译、连接步骤之后才能形成一个真正可执行的二进制机器指令文件
 - C) 用C语言编写的程序称为源程序, 它以ASCII代码形式存放在一个文本文件中
 - D) C语言源程序经编译后生成后缀为.obj的目标程序

(12) 以下选项中, 合法的一组C语言数值常量是()。

- A) 12. 0Xa23 4.5e0
- B) 028 .5e-3 -0xf
- C) .177 4e1.5 0abc
- D) 0x8A 10,000 3.e5

(13) 以下选项中不合法的标识符是()。

- A) print
- B) FOR
- C) &a
- D) _00

(14) 若有代数式 $\sqrt{\ln^x + e^x}$ (其中e仅代表自然对数的底数, 不是变量), 则以下能够正确表示该代数式的C语言表达式是()。

- A) sqrt(fabs(pow(n,x)+pow(x,e)))
- B) sqrt(fabs(pow(n,x)+exp(x)))
- C) sqrt(fabs(n^x+e^x))
- D) sqrt(fabs(pow(x,n)+exp(x)))

(15) 若有定义“double a=22;int i=0,k=18;”, 则不符合C语言规定的赋值语句是()。

- A) i=a%11;
- B) i=(a+k)<=(i+k);
- C) a=a++,i++;
- D) i=!a;

(16) 有以下程序

```
#include <stdio.h>
main()
{
    int s,t,A=10; double B=6;
    s=sizeof(A); t=sizeof(B);
    printf("%d,%d\n",s,t);
}
```

在VC++ 6.0平台上编译运行, 程序运行后的输出结果是()。

- A) 4,8
- B) 4,4
- C) 2,4
- D) 10,6

(17) 有以下程序

```
#include <stdio.h>
main()
{
    char a,b,c,d;
    scanf("%c%c",&a,&b);
    c=getchar(); d=getchar();
    printf("%c%c%c%c\n",a,b,c,d);
}
```

当执行程序时, 按下列方式输入数据(从第1列开始, <CR>代表回车, 注意: 回车也是一个字符)

12<CR>

34<CR>

则输出结果是()。

- A) 12 34
- B) 12
- C) 1234
- D) 12 3

(18) 以下关于逻辑运算符两侧运算对象的叙述中正确的是()。

- A) 可以是任意合法的表达式
- B) 只能是整数0或非0整数
- C) 可以是结构体类型的数据
- D) 只能是整数0或1

(19) 有以下程序

```
#include <stdio.h>
main()
{
    int a=0,b=0,c=0,d=0;
    if(a=1) b=1;c=2;
    else d=3;
    printf("%d,%d,%d,%d\n",a,b,c,d);
}
```

程序输出()。

- A) 编译有错 B) 0,0,0,3 C) 1,1,2,0 D) 0,1,2,0

(20) 有以下程序

```
#include <stdio.h>
main()
{   int x=1, y=0, a=0, b=0;
    switch( x )
    {   case 1:
        switch(y)
        {   case 0: a++; break;
            case 1: b++; break;
        }
        case 2: a++; b++; break;
        case 3: a++; b++;
    }
    printf("a=%d, b=%d\n", a, b);
}
```

程序的运行结果是()。

- A) a=2, b=1 B) a=2, b=2 C) a=1, b=1 D) a=1, b=0

(21) 以下程序段中的变量已正确定义

```
for( i=0; i<4; i++,i++)
    for( k=1; k<3; k++) printf("*");
```

程序段的输出结果是()。

- A) * * B) **** C) * D) *****

(22) 有以下程序段

```
#include <stdio.h>
main()
{
    int i, n;
    for( i=0; i<8; i++)
    {   n = rand() % 5;
        switch(n)
        {   case 1:
            case 3: printf("%d \n", n); break;
            case 2:
            case 4: printf("%d \n", n); continue;
            case 0: exit(0);
        }
    }
    printf("%d \n", n);
}
```

以下关于程序段执行情况的叙述, 正确的是()。

- A) 当产生的随机数n为0时, 结束程序运行 B) 当产生的随机数n为4时, 结束循环操作

C) 当产生的随机数n为1和2时, 不做任何操作 D) for循环语句固定执行8次

(23) 有以下程序

```
#include <stdio.h>
int f(int x);
main()
{   int n=1, m;
    m=f(f(f(n))); printf("%d\n", m);
}
int f(int x)
{   return x*2; }
```

程序运行后的输出结果是()。

- A) 2 B) 8 C) 4 D) 1

(24) 以下叙述中错误的是()。

- A) 当在程序的开头包含头文件stdio.h时, 可以给指针变量赋NULL
B) 函数可以返回地址值
C) 改变函数形参的值, 不会改变对应实参的值
D) 可以给指针变量赋一个整数作为地址值

(25) 设已有定义“float x;”, 则以下对指针变量p进行定义且赋初值的语句中正确的是()。

- A) float *p=&x; B) int *p=(float)x; C) float p=&x; D) float *p=1024;

(26) 以下数组定义中错误的是()。

- A) int x[][3]={1,2,3},{4,5,6}; B) int x[][3]={0};
C) int x[2][3]={1,2},{3,4},{5,6}; D) int x[2][3]={1,2,3,4,5,6};

(27) 有以下程序

```
#include <stdio.h>
void fun( 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 k[10]={ 1,2,3,4,5,6,7,8,9,10};
    fun(k,5);
    for(i=2; i<8; i++) printf("%d", k[i]);
    printf("\n");
}
```

程序的运行结果是()。

- A) 321678 B) 876543 C) 1098765 D) 345678

(28) 有以下程序

```
#include <stdio.h>
#define N 4
void fun(int a[][N], int b[])
{   int i;
    for(i=0; i<N; i++) b[i] = a[i][i] - a[i][N-1-i];
}
```

```
main()
{
    int x[N][N]={{1, 2, 3, 4}, {5, 6, 7, 8}, {9,10,11,12}, {13,14,15,16}}, y[N], i;
    fun(x, y);
    for(i=0; i<N; i++) printf("%d,", y[i]); printf("\n");
}
```

程序运行后的输出结果是()。

A) -3,-1,1,3, B) -12,-3,0,0, C) 0,1,2,3, D) -3,-3,-3,-3,

(29) 设有定义“char s[81]; int i=0;”, 以下不能将一行(不超过80个字符)带有空格的字符串正确读入的语句或语句组是()。

A) while((s[i++] = getchar()) != '\n'); s[i] = '\0'; B) scanf("%s", s);
C) gets(s); D) do{ scanf("%c", &s[i]); }while(s[i++] != '\n'); s[i] = '\0';

(30) 设有定义“char p[]={'1', '2', '3'}, *q=p;”, 以下不能计算出一个char型数据所占字节数的表达式是()。

A) sizeof(p) B) sizeof(char) C) sizeof(*q) D) sizeof(p[0])

(31) 有以下程序

```
#include <stdio.h>
#include <string.h>
main()
{
    char str[][20]={"One*World", "One*Dream!"}, *p=str[1];
    printf("%d", strlen(p)); printf("%s\n", p);
}
```

程序运行后的输出结果是()。

A) 10,One*World B) 9,One*Dream! C) 9,One*World D) 10,One*Dream!

(32) 有以下程序

```
#include <stdio.h>
main()
{
    char s[]="012xy\08s34f4w2";
    int i, n=0;
    for( i=0; s[i]!='\0'; i++)
        if(s[i] >= '0' && s[i] <= '9') n++;
    printf("%d\n", n);
}
```

程序运行后的输出结果是()。

A) 7 B) 0 C) 3 D) 8

(33) 有以下程序

```
#include <stdio.h>
fun(int x, int y)
{
    static int m=0, i=2;
    i+=m+1; m=i+x+y; return m;
}
main()
{
    int j=1, m=1, k;
    k=fun(j, m); printf("%d\n", k);
    k=fun(j, m); printf("%d\n", k);
}
```

执行后的输出结果是()。

A) 5, 11 B) 5, 5 C) 11, 11 D) 11, 5

(34) 在C语言中, 只有在使用时才占用内存单元的变量, 其存储类型是()。

A) auto和register B) extern和register C) auto和static D) static和register

(35) 下面结构体的定义语句中, 错误的是()。

A) struct ord {int x;int y;int z;} struct ord a; B) struct ord {int x;int y;int z;} struct ord a;
C) struct ord {int x;int y;int z;} a; D) struct {int x;int y;int z;} a;

(36) 有以下程序

```
#include <stdio.h>
typedef struct { int b, p; } A;
void f(A c) /* 注意: c是结构变量名 */
{
    int j;
    c.b += 1; c.p += 2;
}
main()
{
    int i;
    A a={1, 2};
    f(a);
    printf("%d,%d\n", a.b, a.p);
}
```

程序运行后的输出结果是()。

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

(37) 以下叙述中正确的是()。

A) 在C语言中, 预处理命令行都以“#”开头
B) 预处理命令行必须位于C源程序的起始位置
C) #include <stdio.h>必须放在C程序的开头
D) C语言的预处理不能实现宏定义和条件编译的功能

(38) 有以下程序

```
#include <stdio.h>
#include <stdlib.h>
int fun(int n)
{
    int *p;
    p = (int*)malloc(sizeof(int));
    *p = n; return *p;
}
main()
{
    int a;
    a = fun(10); printf("%d\n", a+fun(10));
}
```

程序的运行结果是()。

A) 20 B) 10 C) 0 D) 出错

(39) 有以下程序

```
#include <stdio.h>

main()
{   unsigned char  a=8, c;
    c = a>>3;
    printf("%d\n", c);
}
```

程序运行后的输出结果是()。

- A) 32 B) 16 C) 1 D) 0

(40) 读取二进制文件的函数调用形式为“fread(buffer,size,count,fp);”，其中buffer代表的是()。

- A) 一个内存块的首地址，代表读入数据存放的地址
B) 一个整型变量，代表待读取的数据的字节数
C) 一个文件指针，指向待读取的文件
D) 一个内存块的字节数

二、程序填空题

给定程序中，函数 fun 的功能是：将形参 n 中，各位上为偶数的数取出，并按原来从高位到低位的顺序组成一个新的数，并作为函数值返回。

例如，从主函数输入一个整数 27638496，函数返回值为 26846。

请在程序的下划线处填入正确的内容并把下划线删除，使程序得出正确的结果。

注意：源程序存放在考生文件夹下的 BLANK1.C 中。

不得增行或删行，也不得更改程序的结构！

给定源程序：

```
#include <stdio.h>
unsigned long fun(unsigned long n)
{   unsigned long x=0, s, i; int t;
    s=n;
    /*****found*****/
    i= 1 ;
    /*****found*****/
    while( 2 )
    {   t=s%10;
        if(t%2==0){
            /*****found*****/
            x=x+t*i; i= 3 ;
        }
        s=s/10;
    }
    return x;
}

main()
{   unsigned long n=-1;
    while(n>99999999||n<0)
    {   printf("Please input(0<n<100000000): ");
        scanf("%ld",&n); }
    printf("\nThe result is: %ld\n",fun(n));
}
```

三、程序改错题

给定程序 MOD11.C 中函数 fun 的功能是：输出 M 行 M 列整数方阵，然后求两条对角线上元素之和，返回此和数。

请改正程序中的错误，使它能得出正确的结果。

注意：不要改动 main 函数，不得增行或删行，也不得更改程序的结构！

给定源程序：

```
#include <stdio.h>
#define M 5
/*****found*****/
int fun(int n, int xx[M][M])
{   int i, j, sum=0;
    printf( "\nThe %d x %d matrix:\n", M, M );
    for( i = 0; i < M; i++ )
    {   for( j = 0; j < M; j++ )
        /*****found*****/
        printf( "%f ", xx[i][j] );
        printf("\n");
    }
    for( i = 0; i < n; i++ )
        sum += xx[i][i]+xx[i][ n-i-1 ];
    return( sum );
}

main()
{   int aa[M][M]={ {1,2,3,4,5},{4,3,2,1,0},
                    {6,7,8,9,0},{9,8,7,6,5},{3,4,5,6,7} };
    printf( "\nThe sum of all elements on 2 diagnals is %d.",fun( M, aa ));
}
```

四、程序设计题

函数 fun 的功能是：将 a、b 中的两个两位正整数合并形成一个新的整数放在 c 中。合并的方式是：将 a 中的十位和个位数依次放在变量 c 的千位和十位上，b 中的十位和个位数依次放在变量 c 的个位和百位上。

例如，当 a=45，b=12，调用该函数后，c=4251。

注意：部分源程序存在文件 PROG1.C 中。数据文件 IN.DAT 中的数据不得修改。

请勿改动主函数 main 和其他函数中的任何内容，仅在函数 fun 的花括号中填入你编写的若干语句。

给定源程序：

```
#include <stdio.h>
void fun(int a, int b, long *c)
{
}

main()
{   int a,b; long c;
    printf("Input a b:");
    scanf("%d%d",&a, &b);
    fun(a, b, &c);
    printf("The result is: %ld\n", c);
}
```

全国计算机等级考试无纸化真考试卷(4)

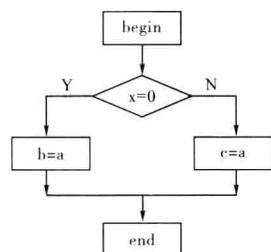
二级C语言

(考试时间120分钟, 满分100分)

一、选择题(每题1分, 共40分)

下列各题A)、B)、C)、D)四个选项中, 只有一个选项是正确的。

- (1) 下列数据结构中, 属于非线性结构的是()。
A) 循环队列 B) 带链队列 C) 二叉树 D) 带链栈
- (2) 下列数据结构中, 能够按照“先进后出”原则存取数据的是()。
A) 循环队列 B) 栈 C) 队列 D) 二叉树
- (3) 对于循环队列, 下列叙述中正确的是()。
A) 队头指针是固定不变的
B) 队头指针一定大于队尾指针
C) 队头指针一定小于队尾指针
D) 队头指针可以大于队尾指针, 也可以小于队尾指针
- (4) 算法的空间复杂度是指()。
A) 算法在执行过程中所需要的计算机存储空间
B) 算法所处理的数据量
C) 算法程序中的语句或指令条数
D) 算法在执行过程中所需要的临时工作单元数
- (5) 软件设计中划分模块的一个准则是()。
A) 低内聚低耦合 B) 高内聚低耦合 C) 低内聚高耦合 D) 高内聚高耦合
- (6) 下列选项中不属于结构化程序设计原则的是()。
A) 可封装 B) 自顶向下 C) 模块化 D) 逐步求精
- (7) 软件详细设计产生的图如下:



该图是()。

- (8) 数据库管理系统是()。
A) 操作系统的一部分 B) 在操作系统支持下的系统软件
C) 一种编译系统 D) 一种操作系统
- (9) 在E-R图中, 用来表示实体联系的图形是()。
A) 椭圆形 B) 矩形 C) 菱形 D) 三角形
- (10) 有三个关系R, S和T如下:

R		
A	B	C
a	1	2
b	2	1
c	3	1

S		
A	B	C
d	3	2

T		
A	B	C
a	1	2
b	2	1
c	3	1
d	3	2

其中关系T由关系R和S通过某种操作得到, 该操作为()。

- A) 选择 B) 投影 C) 交 D) 并
- (11) 以下关于结构化程序设计的叙述中正确的是()。
A) 由三种基本结构构成的程序只能解决小规模的问题
B) 结构化程序使用goto语句会很便捷
C) 一个结构化程序必须同时由顺序、分支、循环三种结构组成
D) 在C语言中, 程序的模块化是利用函数实现的
- (12) 对于一个正常运行的C程序, 以下叙述中正确的是()。
A) 程序的执行总是从main函数开始
B) 程序的执行总是从程序的第一个函数开始, 在main函数结束
C) 程序的执行总是从main函数开始, 在程序的最后一个函数中结束
D) 程序的执行总是从程序的第一个函数开始, 在程序的最后一个函数中结束
- (13) 以下选项中能表示合法常量的是()。
A) "\007" B) 1.5E2.0 C) '\ D) 1, 200
- (14) 以下定义语句中正确的是()。
A) int a=b=0; B) char A=65+1, b='b';
C) float a=1, *b=&a, *c=&b; D) double a=0.0; b=1.1;
- (15) 若变量x、y已正确定义并赋值, 以下符合C语言语法的表达式是()。
A) ++x, y=x-- B) x+1=y C) x=x+10=x+y D) double(x)/10
- (16) 若变量已正确定义为int型, 要通过语句scanf("%d,%d,%d",&a,&b,&c); 给a赋值1、给b赋值2、给c赋值3, 以下输入形式中错误的是() (注: □代表一个空格符)。
A) 1□2□3 B) □□□1, 2, 3
C) 1, □□□2, □□□3 D) 1, 2, 3
- (17) 以下不能输出字符A的语句是() (注: 字符A的ASCII码值为65, 字符a的ASCII码值为97)。
A) printf("%c\n", 65); B) printf("%c\n", 'a'-32);
C) printf("%d\n", 'A'); D) printf("%c\n", 'B'-1);
- (18) 若a是数值类型, 则逻辑表达式(a==1)!(a!=1)的值是()。
A) a的值不确定 B) 0 C) 2 D) 1
- (19) 设有定义:
int a=1, b=2, c=3;
以下语句中执行效果与其他三个不同的是()。
A) if(a>b) c=a; a=b; b=c;
B) if(a>b) {c=a, a=b, b=c;}
C) if(a>b) c=a, a=b, b=c;
D) if(a>b) {c=a; a=b; b=c;}
- (20) 有以下程序
#include <stdio.h>
main()
{ int y=10;
while(y--); printf("y=%d\n", y);
}
程序执行后的输出结果是()。
A) y=0 B) y=-1 C) y=1 D) while 构成无限循环
- (21) 有以下程序
#include <stdio.h>
main()
{

密
封
线
内
不
要
答
题

```
int i,j;
for(i=1; i<4; i++)
{
    for(j=i; j<4; j++) printf("%d*%d=%d ",i,j,i*j);
    printf("\n");
}
```

程序运行后的输出结果是 ()。

- | | |
|--|--|
| A) 1*1=1 1*2=2 1*3=3
2*2=4 2*3=6
3*3=9 | B) 1*1=1 1*2=2 1*3=3
2*1=2 2*2=4
3*1=3 |
| C) 1*1=1
1*2=2 2*2=4
1*3=3 2*3=6 3*3=9 | D) 1*1=1
2*1=2 2*2=4
3*1=3 3*2=6 3*3=9 |

(22) 有以下程序

```
#include <stdio.h>
main()
{ int i=5;
  do
  { if (i%3==1)
    if (i%5==2)
    { printf("%d",i); break; }
    i++;
  } while(i!=0);
  printf("\n");
}
```

程序的运行结果是 ()。

- | | | | |
|-------|---------|-------|---------|
| A) *7 | B) *3*5 | C) *5 | D) *2*6 |
|-------|---------|-------|---------|

(23) 以下叙述中错误的是 ()。

- A) 函数形参的值也可以传回给对应的实参
- B) 函数调用可以作为一个独立的语句存在
- C) 若函数有返回值, 必须通过 return 语句返回
- D) C 程序必须由一个或一个以上的函数组成

(24) 有以下程序

```
#include <stdio.h>
main()
{
    int a=1,b=3,c=5;
    int *p1=&a,*p2=&b,*p=&c;
    *p=*p1*(p2);
    printf("%d\n",c);
}
```

执行后的输出结果是 ()。

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

(25) 有以下程序

```
#include <stdio.h>
void f(int *p,int *q);
main()
{
    int m=1,n=2,*r=&m;
    f(r,&n);
}
```

```
printf("%d,%d",m,n);
}
void f(int *p,int *q)
{
    p=p+1;
    *q=*q+1;
}
```

程序运行后的输出结果是 ()。

- | | | | |
|---------|---------|---------|---------|
| A) 1, 3 | B) 2, 3 | C) 1, 4 | D) 1, 2 |
|---------|---------|---------|---------|

(26) 若有定义语句:

```
int a[2][3], *p[3];
```

则以下语句中正确的是 ()。

- | | | | |
|-------------------|------------|---------|-------------|
| A) p[0]=&a[1][2]; | B) p[0]=a; | C) p=a; | D) p[1]=&a; |
|-------------------|------------|---------|-------------|

(27) 以下程序中函数f的功能是: 当flag为1时, 进行由小到大排序; 当flag为0时, 进行由大到小排序。

```
#include <stdio.h>
void f(int b[],int n,int flag)
{
    int i,j,t;
    for (i=0; i<n-1; i++)
        for (j=i+1; j<n; j++)
            if (flag?b[i]>b[j] : b[i]<b[j])
            { t=b[i]; b[i]=b[j]; b[j]=t; }
}
main()
{
    int a[10]={5,4,3,2,1,6,7,8,9,10};
    f(&a[2],5,0);
    f(a,5,1);
    for (i=0; i<10; i++)
        printf("%d,",a[i]);
}
```

程序运行后的输出结果是 ()。

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

(28) 有以下程序

```
#include <stdio.h>
main()
{
    int s[12]={1,2,3,4,4,3,2,1,1,1,2,3},c[5]={0};
    for(i=0; i<12; i++)
        c[s[i]]++;
    for(i=1; i<5; i++)
        printf("%d ",c[i]);
    printf("\n");
}
```

程序的运行结果是 ()。

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

(29) 设有定义:

```
char *c;
```

以下选项中能够使 c 正确指向一个字符串的是 ()。

- | | |
|-----------------------------------|---------------------|
| A) char str[]="string" ; c=str ; | B) scanf("%s", c) ; |
|-----------------------------------|---------------------|