

# 复旦大学信息科学与工程学院

## 《程序设计》期末考试试卷

A 卷 共 8 页

课程代码: INFO120001

考试形式: 闭卷

2008 年 1 月

(本试卷答卷时间为 120 分钟, 答案必须写在答题纸上, 做在草稿纸和试卷上无效)

专业\_\_\_\_\_ 学号\_\_\_\_\_ 姓名\_\_\_\_\_

题 号	一	二	三	四	总 分
满 分	20	42	18	20	100

( 装  
订  
线  
内  
不  
要  
答  
题  
)

### 一、单项选择题

(1) 下面没有“地址”的是\_\_\_\_\_。

- A. 变量      B. 函数      C. 宏      D. 指针

(2) 下列是 C 语言的关键字的是\_\_\_\_\_。

- A. main      B. sizeof

- C. include      D. printf

(3) 在以下一组运算符中, 优先级最高的运算符是

- A. <=      B. =      C. +      D. &&

(4) 下列说法正确的是\_\_\_\_\_。

- A. 一个 int 类型的变量占两个字节

- B. 一个结构体变量占的字节数, 是结构的每个成员变量字节数的总和

- C. 所有变量都是在执行时分配空间和赋初值

- D. 一个 long int 类型变量可能与一个 int 类型变量占用相同大小的内存

(5) 在 C 语言中, 如果在定义变量时没有对变量做显式初始化, 自动有初值“机器零”(每个二进制位都为 0) 的是\_\_\_\_\_。

- A. char 型的变量      B. 所有指针类型的变量

- C. 各种类型的全局变量      D. 函数的形参

(6) 如果 sizeof(int\*) 的值是 4, 那么 sizeof(char\*) 的值是\_\_\_\_\_。

- A. 1

- B. 2  
C. 4  
D. 不确定
- (7) 阅读下列函数, 函数功能为\_\_\_\_\_。
- ```
void exchange(int *p1, int *p2)
{
    int p;
    p = *p1;
    *p1 = *p2;
    *p2 = p;
}
```
- A. 交换\*p1 和\*p2 的值      B. 正确, 但无法改变\*p1 和\*p2 的值  
C. 交换\*p1 和\*p2 的地址      D. 可能造成系统故障

- (8) 下列定义中, 错误的是\_\_\_\_\_。
- A) char str[7] = "FORTRAN";  
B) char str[] = "FORTRAN";  
C) char \*str = "FORTRAN";  
D) char str[] = {'F', 'O', 'R', 'T', 'R', 'A', 'N', 0};
- (9) 设有定义 int \*p="abc", 下列描述正确的是\_\_\_\_\_。
- A) p 是变量, 指向的字符串"abc"是变量;  
B) p 是变量, 指向的字符串"abc"是常量;  
C) p 是常量, 指向的字符串"abc"是变量;  
D) p 是常量, 指向的字符串"abc"是常量;
- (10) 设有如下定义, 则对 pp 中的 name 成员的正确输入是\_\_\_\_\_。
- ```
struct s {char name[10]; int no; } pp;
```
- A) scanf("%s", pp->name);  
B) scanf("%s", pp.name);  
C) scanf("%s", &pp.name);  
D) 以上都不对

## 二、填空题

- (1) 请用文字解释下列定义中 p 的类型是什么?
- A) int (\*p)();  
B) int (\*p)[3];  
C) int \*p[4];
- (2) 设有变量定义: int a[2][2]={1, 2, 3}, \*p=\*a+1;  
计算表达式\*(p-1)\*(p+1)[1]的值。

(3)写出下面程序执行后的屏幕显示结果

```
#include <stdio.h>
void main()
{
    signed char c=44;
    signed char * p;
    p = &c;
    *p*=4;
    printf("%d", c);
}
```

(4)写出下面程序执行后的屏幕显示结果

```
#include <stdio.h>
void main()
{
    int i=0, j=0;
    for(i=1; ;i++)
        if (i-10)
            break;
        else
            ++j;
    printf("j=%d", j);
}
```

(5)写出下面程序执行后的屏幕显示结果

```
#include <stdio.h>
void main()
{
    char string[] = "nini&lala";
    int i;
    char *p;
    int nBytes = sizeof(string);

    printf("%d\n", nBytes);
    p = &string[nBytes-2];
    for (i=0; i < (nBytes-1); i++)
        putchar(*(p--));
}
```

(6)写出下面程序执行后的屏幕显示结果

```
#include <stdio.h>
int a=2,c=4;
void f(int a, int *x){
    int b=10;
    static int c=20;
    b+=a++;
    c+=a+b;
    *x=c+2;
}
void main(){
    f(3+c, &a);
    f(a,&c);
    printf("a=%d,c=%d\n",a,c);
}
```

(7)下面的程序执行到最后，打印出来的变量 count 的值是多少？

```
#include <stdio.h>
int hanoi ( int n,char A,char C,char B){
    static int i=1;
    if ( n == 1 ){
        printf("%d: move %c  to %c\n",i, A, C);
        i++;
    }
    else {
        hanoi ( n-1, A, B, C );
        printf( "%d: move %c  to %c\n",i, A, C);
        i++;
        hanoi ( n-1, B, C, A );
    }
    return i;
}
void main(){
    unsigned int count;
    count=hanoi(10,'A','C','B');
    printf("%d", count);
}
```

~  
装  
订  
线  
内  
不  
要  
答  
题  
~

(8) ~ (9) 勒让得多项式的递归定义如下，随后的函数用非递归方法求勒让得多项式  $p(n,x)$  的值，返回求值结果。

n=0;  
n=1;  
n>1.

```
double p(int n, double x){  
    double first, second, third;  
    int count;  
    if (n == 0) return 1.0;  
    if (n == 1) return x;  
    first = 1.0;  
    second = x;  
    for( _____ (8) ;count++) {  
        third = ((2.0*count-1.0)*x*second  
                  - (count-1.0)*first)/count;  
        _____ (9) ;  
        second = third;  
    }  
    return third;  
}
```

(10) ~ (12) 不带辅助表元链表颠倒函数。如需颠倒链表的头指针为  $h$ ，调用  $reverse(&h)$  即可将其颠倒。链表表元结构定义如下：

```
struct intNode { //整数链表表元类型  
    int value ; //存放整数  
    struct intNode *next ; //存放后继表元的指针  
};  
void reverse (struct intNode **hpt){  
    struct intNode *p, *v1, *v2;  
    v1 = NULL; //开始颠倒时，已颠倒部分为空  
    v2 = *hpt; //v2 指向链表的首表元  
    while ( _____ (10) ) {  
        p = v2->next;  
        v2->next = v1;  
        _____ (11) ;  
        v2 = p;  
    }  
    _____ (12) ;  
}
```

(13) ~ (14) 以下程序的功能是从键盘上输入若干个字符（以回车符作为结束）组成一个字符串存入一个字符数组

```
#include <stdio.h>
void main()
{
    char str[81];
    int i;
    for(i = 0; i < 80; i++)
    {
        str[i] = getchar();
        if (str[i] == '\n')
            _____ ;
    }
    _____ ;
}
```

### 三、改错

请用最少的修改让下面的程序变得正确。修改以行为单位，在答题纸上写上修改后完整的一行。

(1) 下面的程序要打印 0、1、2、3、4 这 5 个数，且每个数各占一行。

```
/*1*/ #include <stdio.h>;
/*2*/ #define PRINT(a) printf("%d", a);printf("\n")
/*3*/ void main()
/*4*/ {
/*5*/     int i;
/*6*/     for(i=0; i<5; )
/*7*/         PRINT(i++);
/*8*/ }
```

(2) 下面的程序要从键盘读入数据，给数组的每个元素赋值，然后判断数组首尾元素的值是否相等，如相等打印提示。

```
/*1*/ #include <stdio.h>
/*2*/ #define MAX 10
/*3*/ void main()
/*4*/ {
/*5*/     int size=MAX, i;
/*6*/     int f[size];
/*7*/
/*8*/     for (i=0; i<=size; i++)
/*9*/         scanf("%f", &f[i]);
/*10*/
/*11*/     if (f[0] == f[i])
/*12*/         printf("您输入的数组首尾元素相等\n");
/*13*/ }
```

(3) 下面的程序要读入正文文件的内容，统计文件中英文字母的个数，并输出。设程序要统计的正文文件名在程序启动时由输入指定。

```
/*1*/ #include <stdio.h>
/*2*/ void main(){
/*3*/     FILE *fp;
/*4*/     int count;
/*5*/     char ch;
/*6*/     char fname[40];
/*7*/     printf("输入文件名!\n");
/*8*/     scanf("%s%c", fname); //读入文件名和名后的回车符
/*9*/     if ((fp = fopen(fname, 'r')) == EOF){
/*10*/         printf("Can not open %s file.\n", fname);
/*11*/         return 1; // 程序非正常结束
/*12*/     }
/*13*/
/*14*/     count = 0;
/*15*/     while((ch = fgetc(fp)) != EOF) {
/*16*/         if('a'<ch<='z'||'A'<=ch<='Z')
/*17*/             count++;
/*18*/     }
/*19*/     fclose(fname);
/*20*/     // 输出处理结果
/*21*/     printf("文件%s有英文字母%d个.\n", fname, count);
/*22*/     return 0; // 程序执行正常结束
/*23*/ }
```

## 四、编程（计算机专业选 1, 3, 其他专业选 1, 2）

(1) 编写函数，对一个不带辅助表元的整数无序链表按从小到大次序排序。函数原型如下：

```
void sortLink(struct intNode ** hpt);
```

intNode 的定义如下：

```
struct intNode { //整数链表表元类型
    int value; //存放整数
    struct intNode *next; //存放后继表元的指针
};
```

例如，有一无序链表的头指针为 h，调用 sortLink(&h) 后，就对该链表的表元按 value 值从小到大次序排列了。

(2) 编写函数，把一个字符串的内容前后颠倒。函数原型如下：

```
void reverseStr(char * s);
```

(3) 从  $n$  个不同物体中任选  $r$  个物体的方案个数叫组合数。有这样两个计算公式：

$$\text{公式一: } C_n^r = \frac{n!}{r!(n-r)!}$$

$$\text{公式二: } C_n^r = \begin{cases} 1 & n = r \\ n & r = 1 \\ C_{n-1}^{r-1} + C_{n-1}^r & n > r \end{cases}$$

你认为哪个公式更适合用程序实现？为什么？然后编写一个用该公式计算组合数的函数，函数原型如下：

```
unsigned int Comb(unsigned int n, unsigned int r);
```