

仅供组织内部学习参考!

试卷中的答案部分有误,文字描述有些也不够完整,仅供参考,祝考试顺利!



厦门大学《C 程序设计》课程试卷

信息学院 软件工程系 2021 级 软件工程专业

学年学期: 21-22/1 主考教师: A 卷答案

说明: 将答案按顺序写在学校统一印制的答题卷上, 写在本卷或自备纸上者一律不得分。

一、单选题 (共 20 分, 每小题 1 分)

答: CACDB CBDAB CBDBA BADCD

1. 下列关于 C 语言程序的说法正确的是 C。
A. 运行时从文件中第一个函数开始执行
B. 语句是 C 语言程序的基本构成单位
C. 可以将一个语句分多行书写 ☒
D. 源代码的命名不区分大小写
2. 下列 4 组 C 语言程序候选变量名中 (以分号间隔), 全部合法的是 A。
A. print; _3d; db8; main ☒
B. I'm; one_half; Float; 3pai
C. str_1; Cpp; pow; while
D. Pxq; Main; a-d; 111
3. 下面 4 个选项中 (以分号间隔), 均为合法浮点数的选项是 C。
A. 160.; 0.12; E3
B. 123; 2e4.2; .e5
C. -.8; .3e+4; 0.0 ☒
D. -e3; 0.234; 1e3
4. 运行 `int x=5; scanf("%d", &x); char c=getchar();`, 输入 u 并按回车键, 则 D。
A. c 被赋值为回车符
B. 回车后输入数字方可正确赋值 x
C. x 的值不确定
D. x 为 5, c 为 117 (u 的 ASCII 码) ☒
5. 设整型变量 a=2, 执行语句 B 后, 浮点型变量 b 的值不为 0.5。
A. b=1.0/a
B. b=(float)(1/a) ☒
C. b=1/(float)a
D. b=1/(a*1.0)
6. 设声明 `const int x=90;`, 以下表达式能正确得到值为 1 的是 C。
A. x-=89
B. -2<=-x<1
C. 85<x<89 ☒
D. x
7. 以下关于 for 循环, 正确的说法是 B。
A. 只能用于循环次数已经确定的情况
B. 可包含多条语句, 但必须用花括号 ☒
C. 不能用 break 语句跳出循环体
D. 先执行循环体, 后判断循环条件
8. 以下关于函数头中的形式参量, 不正确的说法是 D。
A. 不同函数中可以声明同名的形式参量
B. 函数的形式参量不可以声明为 static
C. 函数的形式参量可以声明为 const
D. 函数声明形式参量时可以赋默认值 ☒
9. 有 `int i=5, s=0;`, 执行 `while(i--){ if (i%2) continue; else s+=i; }` 后, s 值为 A。

A. 6 ✓

B. 9

C. 10

D. 15

10. 有声明语句 `double a=0, b=0, x=4;` , 则以下选项中正确的是 B 。

A. `switch((int)x/2.0) {case 0.0: a++; break; case 1.0: b++; }`

B. `switch((int)x%2) { case 0: a++; break; case 1: b++; }` ✓

C. `switch(x%2) { case 0: a++; break; case 1: b++; break; }`

D. `switch((int)(x)/2) {case 1-1: a++; break; case 4%3: b++; }`

11. 声明 `int mat[10][30]` 后, 以下关于两个元素的内存地址说法不正确的是 C 。

A. `mat[3][7]` 与 `mat[3][8]` 的地址相邻

B. `mat[3][7]` 与 `mat[4][7]` 的地址不相邻

C. `mat[3][30]` 与 `mat[4]` 的地址不相同 ✓

D. `mat[4][0]` 与 `mat[120]` 的地址相同

12. 关于字符串声明, 以下说法正确的是 B 。

A. `char b[10]={ "xmu" }`; 等价于 `char b[10]; b[]={ "xmu" }`;

B. `char *a="xmu"`; 等价于 `char *a; a="xmu"`; ✓

C. `char *c="xmu"`; 等价于 `char c[]="xmu"`;

D. `char d[]="xmu"`; 等价于 `char d[3]="xmu"`;

13. 以下说法不正确的是 D 。

A. 同一文件内的不同函数可以包含同名的 `goto` 标记

B. 声明变量为 `extern` 时, 不可以对变量进行初始化

C. 声明变量为 `static` 时, 在其外部不可以声明该变量为 `extern` 的

D. 声明全局变量为 `extern` 时, 不可同时声明其为 `const` 的 ✓

14. 若执行 `fopen` 函数时文件无法访问, 则函数的返回值是 B 。

A. 文件指针

B. `NULL (0)` ✓

C. `EOF (-1)`

D. "No Access"

15. 设一个 64 位程序中有 `int a=0, b=2, c=4; struct { int n; int *m; } s[3]={ {100,&a},{200,&b},{300,&c}}`, `*p=s;` , 以下值为 2 的表达式是 A 。

A. `*(++p)->m` ✓

B. `*(++p->m)`

C. `++*p->m`

D. `*++p->m`

16. 有预定义 `#define SQR(X) X*X`, 则 `y=2*SQR(5+2)` 的值是 B 。

A. 2

B. 22 ✓

C. 34

D. 98

17. 以下函数形式参量声明中, 合法的是 A 。

A. `int f(int p[][9]);` ✓

B. `int f(int p[9][]);`

C. `int f(int p[][]);`

D. `int f(int [3][3]p);`

18. 判断两个字符串 `a` 和 `b` 是否相等, 应使用 D 。

A. a == b B. a = b C. strcmp(a,b) D. !strcmp(a,b) ✓

19. 在声明 int *p,q; int a[40]={}; 后紧接着执行以下赋值语句, 不正确的是 C。

A. q=*a; B. p=&q; C. *p=a; ✓ D. q=a[0];

20. 以下说法不正确的语句是 D。

A. 循环语句的循环体可包含循环语句 B. 结构体 s 中不可包含类型为 s 的成员
C. 函数 f 可在运行时调用函数 f 自身 D. 宏 A 定义时在实体可包含该宏 A ✓

二、填空题 (共 20 分, 每个空位 1 分)

1. 程序的三种基本结构为顺序结构、(1) 和 (2)。

答: (1) 分支结构, (2) 循环结构。 顺序可以调换。

2. 连接程序将 (3) 后得到的目标模块连接起来, 再与函数库连接为一个整体, 生成 (4)。

答: (3) 编译, (4) 可执行程序

3. 全局变量在 (5) 范围内可以使用其变量名指代该变量 (被同名局部变量覆盖的范围除外), 其占用的内存空间在 (6) 时释放。

答: (5) 声明之后至本文件结束, (6) 程序运行结束。

4. 下列程序段 char s=8; do ++s; while(s>0); 的循环体, 共执行 (7) 次。

答: (7) 120。

5. 设变量 float n, 将 n 四舍五入保留小数点后 2 位的表达式是 (8)。

答: (8) ((int)(n*100+0.5))/100.0 或者 floor(n*100+0.5)/100.0, round(n*100)/100.0。

6. 设声明浮点型变量 x=1, y=3, 则在运行语句 y/=++x&& y<3?3:5; 之后, y 的值为 (9)。

答: (9) 0.6。

7. 请写出以下各声明语句中变量 p 的含义:

int *p[4]; (10); int (*p)[4]; (11);

int *p(); (12); int (*p[4])(); (13)。

答: (10) 指针数组; (11) 指向数组的指针; (12) 返回指针的函数; (13) 指向函数的指针。

8. 函数调用语句 `fgets(buf,n,fp);` 从 `fp` 指向的文件中读入 (14) 存入 `buf` 字符数组中。该函数若读入成功返回 `buf`，读入失败则返回为 (15)。

⑦
答：(14) 一行且不超过 `n` 个字符，(15) `NULL`。

9. 赋值操作符、关系运算符和算术运算符的优先级从高到低排序结果是 (16)。

答：算术运算符、关系运算符、赋值运算符。

10. 在 C 语言程序中，有 2 种情况需要为变量加上 `static` 关键字，包括：(17)；(18)。

答：(17) 声明局部变量为静态，避免退出函数时销毁变量；(18) 将全局变量声明为私有，声明其不被外部变量使用。

11. 设声明 `char s[] = "\\t\\0876\\0678";` 则 `sizeof(s)` 的值为 (19)，而 `strlen(s)` 的值为 (20)。

答：(19) 9；(20) 2。

三、分析题（共 30 分，每小题 6 分）

1. (2+1+1+2) 请填空，使其根据用户输入的初速度、加速度和时间计算位移。其公式为：

$$s = v_0 t + \frac{1}{2} a t^2。$$

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

```
int main(void) {
```

```
    double v0 = 0, a = 0, t = 0, s = 0;
```

```
    printf("Enter the Initial Velocity, Acceleration, and Time: ");
```

```
    (1)
```

```
    if ( (2) ) {
```

```
        fprintf(stderr, "Time cannot be negative.\n");
```

```
    (3)
```

```
    }
```

```
    (4) // the main equation
```

```
    printf("The total shift is %.3lf m.\n", s);
```

```
    return 0;
```

```
}
```


参考答案：(1) scanf("%lf%lf%lf",&v0,&a,&t); 次序不可错 (2) t<0 (3) return -1;

(返回其它值也可以) (4) s = v0*t + 1.0 / 2 * a*t*t 不可出现 1/2

2. (1+1+2+1+1) 素数是指大于 1 且只能被 1 和自身整除的自然数。请填空完成以下程序，用于判断整数 n 是不是素数。当 n 是素数时返回 1，否则返回 0。

```
int is_prime(unsigned int n) {
    unsigned int i, k;
    k=____(1)____;
    for(____(2)____; i<k; i++)
        if(____(3)____) break;
    if (____(4)____) return 0;
    else ____ (5) ____;
}
```

参考答案：(1) n 或者 $n/2$; (2) $i=2$ (3) $n\%i$ (4) $n<2$ (5) return $i==k$ 或 return $i=k$ 的值。

或者 (4) if ($i!=k \parallel n<2$) , (5) return 1;

3. (2+2+2) 以下调用 fun(163)的输出是____(1)____。该程序的功能是____(2)____。

使用递归函数解决问题的优点在于代码简洁，缺点在于____(3)____。

```
void fun(unsigned char n) {
    int r = n % 2;
    if (n >= 2)
        fun(n / 2);
    putchar(r == 0 ? '0' : '1');
}
```

参考答案：(1) 10100011 ; (2) 输出 n 的二进制表示 ; (3) 时间和空间消耗比较大。

4. (1+2+1+2) 请填空完成程序，对给定区间 $[a, b]$ ，求在该区间内能被 n_1 或 n_2 整除的数。

例如：输入 9 15 3 7，输出 Answer: 9 12 14 15.; 输入 9 15 8 17，输出：No solution.。

```
#include <stdio.h>
int main() {
    int a,b,n1,n2,i,count=0;
    scanf("%d %d %d %d",&a,&b,&n1,&n2);
    for(i=a;____(1)____;i++)
        if(____(2)____) {
            count++;
        }
}
```

```

        if(____(3)____) printf("Answer:");
        printf(" %d",i);
    }
    if (____(4)____) printf("No solution.\n");
    else printf(".\n");
    return 0;
}

```

参考答案：(1) i<=b ; (2) i%c==0 || i%d==0 ; (3) count==1 ; (4) count==0 。

5. (2+1+1+1+1) 请填空以下程序，采用向链表末尾添加节点的方法，生成一个存储一组实数的链表。

```

#include <stdlib.h>
____(1)____ // 声明链表节点 node 结构体类型
____(2)____ // 声明 link_list 类型为指向 node 的指针
link_list create_linklist_from_tail() {
    link_list head = NULL, current = NULL, prev = NULL;
    double x;
    puts("Enter a set of numbers (end with 'q');");
    while (scanf("%lf", &x) == 1) {
        ____ (3) ____
        current->data = x;
        current->next = NULL;
        if (____ (4) ____)
            head = current;
        else
            ____ (5) ____
        prev = current;
    }
    return head;
}

```

参考答案：(1) struct node { double data; struct node *next; }; ; (2) typedef struct node *link_list; ; (3) current = (link_list *)malloc(sizeof(struct node)); 其中 link_list 部分可以等价写为 struct node *，但 struct 不能省略；(4) head==NULL ; (5) prev->next = current; 。

四、设计题（共 30 分，每小题 10 分，答案写在答卷纸上，并标明题号）

1. 编写程序实现：输入一组年份 years，按照闰年整体排在平年前，闰年和平年内部分别由小到大排序（排序算法不限）。要求写 2 个函数：判断闰年函数 int is_leap_year(int year);和排序函数 void sort(int years[], int n);。对 1900,1930,1940,1967,1988 排序的结果为 1940,1988,1900,1930,1967。

评分要点：

1、（2分）正确实现函数头和函数体框架、返回语句；

2、（4分）正确实现排序 sort，算法不限；

3、（4分）正确书写 is_leap_year 函数；

参考答案：

```
int is_leap_year(int year) {
    if (((year%100!=0) && (year%4==0)) || (year % 400==0))
        return 1;
    return 0;
}

void sort(int years[ ], int n) {
    int temp;
    for (int i=0; i<n; i++)
        for (int j=n-1; j>i; j--)
            if(is_leap_year(years[j])==is_leap_year(years[j-1]) &&
                years[j]<years[j-1]
                || is_leap_year(years[j]) && !is_leap_year(years[j-1])) {
                temp=years[j];
                years[j]=years[j-1];
                years[j-1]=temp;
            }
}
```

排序算法也可以是插入排序：

```
void inssort(int a[], int n) {
    int i, j, k;
    for (i = 0; i < n; i++) {
        k = a[i];
        j = i - 1;
        while ((j >= 0) && (a[j] > k)) {
            a[j + 1] = a[j];
            j--;
        }
        a[j + 1] = k;
    }
}
```



```

    }
    a[j + 1] = k;
}
}

```

快速排序：

```

int partions(int l[], int low, int high) {
    int prvotkey = l[low];
    l[0] = l[low];
    while (low < high)
    {
        while (low < high && l[high] >= prvotkey)
            --high;
        l[low] = l[high];
        while (low < high && l[low] <= prvotkey)
            ++low;
        l[high] = l[low];
    }
    l[low] = l[0];
    return low;
}

void qsort(int l[], int low, int high) {
    int prvotloc;
    if (low < high)
    {
        prvotloc = partions(l, low, high);
        qsort(l, low, prvotloc-1);
        qsort(l, prvotloc+1, high);
    }
}

```

选择排序：

```

void selsort(int a[], int n) {
    int i, j, k, m;
    for (i = 0; i < n-1; i++)
    {
        k = i;
        for (j = i + 1; j < n; j++)
            if (a[k] > a[j])
                k = j;
    }
}

```



```

        if (k != i)
        {
            m = a[i];
            a[i] = a[k];
            a[k] = m;
        }
    }
}

```

2. 今年 3 月，有关部门联合对外发布通知，禁止小贷公司向大学生发放互联网消费贷款。部分贷款利用类似“借 1000 元日息 0.4 元”的话术掩盖了高利贷真相。贷款一般采用等额本息还款，即借款人每个月按照相等的金额来偿还贷款本息，每个月的贷款利息按照月初剩余贷款本金来计算。计算公式为：

$$a = \frac{Nr(1+r)^m}{(1+r)^m - 1}.$$

其中， a 为每月还款额， N 为贷款本金， $r = \frac{R}{12}$ 为月利率（ R 为年化利率）， m 为还款月数。请编写程序实现函数 `double interest(double N, double R, int m)`，如果 $N, R, m > 0$ ，计算并返回每月还款额 a ，参数不合法则返回 -1。

评分要点：

1、（4 分）正确实现函数头和函数体框架、返回语句；

2、（4 分）正确实现主要公式（循环部分）；

3、（2 分）对 N 、 R 、 m 是否为正数进行判定；

参考答案：

```

double interest(double N, double R, int m)
{
    double r=R/12;
    double a=0,b=1;
    if ( N <= 0 || R <=0 || m<=0 )
        return 0;
    for (int i=0;i<m;i++)
    {
        b*=(1+r);
        a=n*r*b/(b-1);
    }
    return a;
}

```

3. 实现完整的 C 语言程序，从文本文件 1.txt 中读取一组坐标值，遍历找出中距离最接近的一对所在的序号，以空格间隔输出到文本文件 2.txt 中。其中，文件 1.txt 内容为 $2N$ 个整数，其含义为 N 个直角坐标 (x_i, y_i) ，其中， $2 \leq N \leq 50, 0 \leq x_i, y_i \leq 1000$ 。坐标应使用结构体 struct point 定义。如有多组解，输出其中任意一种情况即可。注意勿使用 freopen() 函数实现文件操作。以下是文件的示例：

已有 1.txt 的内容	输出到 2.txt 的内容
3 2	1 3
1 5	
2 3	

评分要点：

1、（1分）完整的#include，main 函数框架；

2、（4分）文件操作：开、关、读写、判断文件末尾（或根据 if (fscanf(……)!=2) break; 判断输入结束）；

3、（4分）计算最短距离（定义最短距离初始值，遍历查找值更短则记录下值及其下标）；

4、（2分）point 的结构体定义；

参考答案：

```
#include <stdio.h>
struct point {
    double x;
    double y;
};
int main() {
    int n,i,j,pi,pj;
    double min_dist,cur_dist;
    struct point p[60];
    FILE *f1=fopen("1.txt","r");
    FILE *f2=fopen("2.txt","w");
    for(n=0; !feof(f1); n++)
        fscanf(f1,"%lf %lf",&p[n].x,&p[n].y);
    --n;
    pi=0;
    pj=1;
```



```

min_dist=(p[0].x-p[1].x)*(p[0].x-p[1].x)+(p[0].y-p[1].y)*(p[0].y-p[1].y);
for(i=0; i<n; i++) {
    for(j=i+1; j<n; j++) {
        cur_dist=(p[i].x-p[j].x)*(p[i].x-p[j].x)+(p[i].y-p[j].y)*(p[i].y-p[j].y);
        if (cur_dist<min_dist) {
            min_dist=cur_dist;
            pi=i;
            pj=j;
        }
    }
}
fprintf(f2,"%d %d\n",pi+1,pj+1);
fclose(f1);
fclose(f2);
return 0;
}

```