

浙江大学 2017 - 2018 学年春夏学期

《程序设计专题》课程期末考试答案及评分标准

课程号: 211G0260, 开课学院: 计算机学院

考试试卷: ☒ A 卷、B 卷 (请在选定项上打 ☒)

考试形式: ☒ 闭、开卷 (请在选定项上打 ☒)，允许带 / 入场

考试日期: 2018 年 07 月 05 日, 考试时间: 120 分钟

试题一、单选题 (每小题 3 分, 共 30 分)

- | | | | | |
|------------|------------|------------|------------|-------------|
| 1 <u>A</u> | 2 <u>D</u> | 3 <u>C</u> | 4 <u>B</u> | 5 <u>D</u> |
| 6 <u>A</u> | 7 <u>C</u> | 8 <u>D</u> | 9 <u>C</u> | 10 <u>D</u> |

试题二、改错题 (共 20 分)

本题 2 分

- 1 `extern` 改为 `static`, 或该变量定义为静态全局变量

评分标准: 符合答案意图的说法或从全局变量作用域扩展到多个文件的角度的改法都得 2 分, 仅指出问题所在 1 分。

本题 2 分

- 2 不一样, 归并排序是 $O(N\log N)$

评分标准: 只要表达仅归并排序是 $O(N\log N)$ 得 2 分, 仅指出归并排序不一样得 1 分, 或其他合理表达得 1 分

本题 2 分

- 3 第 4 行, `m/n` 改为 `m%n`

评分标准: 符合答案改法或其他可能的递归函数正确版本 2 分, 仅指出错误得 1 分

- 4 本题 4 分

(1) 第 8 行改为 `p = (char*)malloc(sizeof(char)*LENGTH);`

(2) `while` 语句前插入 `q = p;`

(3) 第 15 行 `free(p)` 改为 `free(q)`

或者 `while` 语句中的所有 `p` 改为 `q`

评分标准: 符合答案改法 4 分, 否则: 对答案中 2 点给 3 分, 1 点给 2 分; 仅改第一点时将 `char *` 写为 `*char` 给 1 分

- 5 本题 5 分

(1) 第 2,3 行调换次序, `a1.h` 改为 `_A1_H`

(2) 第 8 行 `ENDIF` 改为 `endif`

评分标准: 符合答案或等效改法给 5 分, `_A1_H` 写为 `_A1_H` 不扣分。仅对 (1) 给 3 分, 若只写对第 2,3 行调换次序给 2 分, 只写对 `a1.h` 改为 `_A1_H` 给 1 分 (2)

对给 2 分。

6 本题 2 分

不能返回在函数 `getone()` 内部定义的局部数据对象的地址，因为局部数据对象在函数返回时就会消亡，其值不再有效

评分标准：原因不用写，只要指出错误就给 2 分，包括指出 `return s`；错误，或 `str[80]` 定义为静态局部变量，或全局变量都可以

7 本题 3 分 s

(1) 第 9-13 行，结构定义应移到第 3 行前

(2) 第 18 行，去掉一个 `%d`，`&s1.name` 去掉 `&`

评分标准：符合 (1)，即指明结构定义应该在函数外给 1 分；符合 (2)，只要写出去掉一个 `%d` 或 `&s1.name` 去掉 `&` 给 2 分。

试题三、问答题（共 10 分）

- 1 结构体变量作为函数参数的优点使可以减少函数参数的个数；缺点是：函数参数传递时需要将整个实参的结构体变量的内容复制到函数的形参，如果涉及到结构体内部存在较大的数据成员，会存在大量复制，影响效率；使用结构体指针可以避免结构体变量实参传递时的数据内容复制，只传递的是结构体变量地址，提高效率。

共 4 分，评分细则：

指出优点（重点突出：减少函数参数的个数）1 分；

指出缺点（重点突出：存在大量复制，影响效率）1 分；

指出使用结构体指针的优势（重点突出：避免复制，提高效率）2 分。

- 2 1) 外层循环变量初始化次数 1，比较次数 10001，自增次数 10000
内层循环变量初始化次数 10000，比较次数 6×10000 ，自增次数 5×10000
和赋值次数 5×10000

总计 $= 1 + 10001 + 10000 + 10000 + 6 \times 10000 + 5 \times 10000 + 5 \times 10000$
 $= 190002$

评分细则：

体现内外层循环变量初始化次数、比较次数、自增次数和赋值次数，就 2 分；或直接回答：190002，也给 2 分。

2) 改写后的代码：

```
for(row = 0; row < 5; row++) {  
    for(column = 0; column < 10000; column++) {  
        sum += table[row][column];  
    }  
}
```

评分细则：

优化代码：内外层循环变量交换，2 分

改写双层循环为单层循环，2 分

3) 把内外循环交换。

外层循环变量初始化次数 1，比较次数 6，自增次数 5
 内层循环变量初始化次数 5，比较次数 5*10001，自增次数 5*10000
 和赋值次数 5*10000
 总计=1+6+5+5+5*10001+5*10000+5*10000
 =150022

通过分析，此方法比原代码，节省 (190002-150022)/190002 即 21.04% 的时间。
 在多重循环设计时，把最忙的循环放在内层。

评分细则：体现优化后代码的执行次数的计算，1 分； 得出结论：1 分。

试题四、填空题（每小题 2 分，共 26 分）

1	<code>p != NULL, 或 p, 或 p != 0</code>	2	<code>p->data <= t->data, 或 p->data <= p->next->data</code>
3	<code>free(t)</code>	4	<code>a[i].con=0</code>
5	<code>j = i-1</code>	6	<code>a[i].num < a[j].num 或 a[i].num <= a[j].num</code>
7	<code>int event</code>	8	<code>MOUSEMOVE</code>
9	<code>InitGraphics()</code>	10	<code>registerMouseEvent(Painter)</code>
11	<code>--t 或 t--或 t=t-1 或 t-=1</code>	12	<code>++s 或 s++或 s=s+1 或 s+=1</code>
13	<code>Qsort(a, s + 1, high) 或 Qsort(a, t+1, high)</code>		

评分细则：

1. 符合一种参考答案，给满分（2 分），否则 0 分。大小写笔误不扣分。
2. 符合一种参考答案或者等价形式（例如将 `p->` 写作 `(*p).`），给满分（2 分），否则 0 分。大小写笔误不扣分。将 `p->` 和 `t->` 写作 `p.` 或者 `t.` 的，酌情给 1 分。
3. 符合参考答案，给满分（2 分），否则 0 分。大小写笔误不扣分。
4. 符合参考答案，给满分（2 分），否则 0 分。大小写笔误不扣分。
5. 符合参考答案，给满分（2 分），否则 0 分。大小写笔误不扣分。
如果写 `j=i`，酌情给 1 分。
6. 符合一种参考答案或者等价形式（例如将不等式反过来），给满分（2 分），否则 0 分。大小写笔误不扣分。缺少 `.num`，酌情给 1 分。

7. 符合参考答案，给满分（2分），否则0分。大小写拼写笔误不扣分。
8. 符合参考答案，给满分（2分），否则0分。写出 MOVE 酌情给1分。大小写拼写笔误不扣分。
9. 符合参考答案，给满分（2分），否则0分。大小写和少量拼写笔误不扣分。写出 InitWindow 等近似意思的酌情给1分。
10. 符合参考答案，给满分（2分），否则0分。答案中体现出“注册鼠标回调”意图的，酌情给1分。大小写拼写笔误不扣分。
11. 符合一种参考答案，给满分（2分），否则0分。
12. 符合一种参考答案，给满分（2分），否则0分。
13. 符合一种参考答案，给满分（2分），否则0分。答案中的函数调用参数不全对，但是至少有一个是对的，酌情给1分。

试题五、算法设计（共14分）

5-1. 参考答案（5分）

1)（2分）

递归式 $\text{sum}(n)=a[n-1]+\text{sum}(n-1)$

递归出口 $n=1$ 返回 $a[n-1]$ ，即 $a[0]$

2)（3分）

```
int sumArr(int a[],int n)
{
    if(n>1)
        return a[n-1] + sumArr(a,n-1);
    else
        return a[0];
}
```

评分标准及说明：

（1）递归式与递归出口 2分

递归式 $\text{sum}(n)=a[n-1]+\text{sum}(n-1)$ （1分）

递归出口 $n=1$ 返回 $a[n-1]$ ，即 $a[0]$ （1分）

用数学函数表示，对。

$\text{Fsum}(n)=\text{Fsum}(n-1)+a_n$

等价答案：

递归式 $\text{sum}(a,n)=a[0]+\text{sum}(a+1,n-1)$ （1分）

递归出口 $n=1$ 返回 $a[n-1]$ ，即 $a[0]$ （1分）

另外：

递归式 数组元素 $a[0]$ 到 $a[n-1]$ 或 $a[1]$ 到 $a[n]$ 都算正确，或 a_n 和 a_{n-1} 都对。

递归出口 $n=0$ 返回 0，正确，多递归调用 1 次。

(2) 写出递归函数 3 分

int sumArr(int a[],int n) 题目已给出

```
{
    if(n>1)
        return a[n-1] + sumArr(a,n-1);    (2 分)
    else
        return a[0];                    (1 分)
}
```

或

```
int sumArr(int a[],int n)
{
    if(n>0)
        return a[n-1] + sumArr(a,n-1);    (2 分)
    else
        return 0;                    (1 分)
}
```

或

```
int sumArr(int a[],int n)
{
    if(n)
        return a[n-1] + sumArr(a,n-1);    (2 分)
    else
        return 0 或 n;                (1 分)
}
```

或

```
int sumArr(int a[],int n)
{
    if(n>1)
        return a[0] + sumArr(a+1,n-1);    (2 分)
    else
        return a[0];                    (1 分)
}
```

5-2. 参考答案与评分标准

方法一、用归并法把 A[]和 B[]合并到 C[]，那么 $C[(2n-1)/2]$ 就是所求的 m；（只需要归并到第 $(2n-1)/2+1$ 个元素，而不需要归并所以元素）。时间复杂性是 $O(n)$ 。（这种解法，给 7 分）

方法二、(满分 9 解法) 该方法采用分治递归思想，程序可以递归非递归实现，时间复杂性为

$O(\log n)$

(1) 算法思想：定义递归函数：

```
int Median (int A[ ], int B[ ], int n);
```

递归计算中位数 m 。递归式如下：

1、当 $n=1$ 时，中位数 $m = \min(A[0], B[0])$ (递归出口 1)

2、当 $n=2$ 时，(求 4 个元素的中位数，递归出口 2)

```
if( A[0] < B[0] )    m = min ( A[1], B[0] )
```

```
else    m = min ( A[0], B[1] )
```

3、当 $n > 1$ 时 (此时递归)

```
    设 ma = A[ n/2 ];    mb = B[ n/2 ];
```

```
    if ( ma == mb ) return 中位数 m = ma; (出口 3，结束递归)
```

```
    else if ( ma > mb ) // 舍去中位数大的后半段、舍去中位数小的前半段，递
```

归

```
        return Median( A, &B[n/2+1],    n/2 );
```

```
    else //  ma < mb
```

```
        return Median ( &A[n/2+1],    B, n/2 );
```

方法三、任何通过超过 $O(n)$ 时间复杂性的排序方法都可以求得中位数，过程正确，给 5 分。

评分标准：满分 9 分

a、 方法二、分治法 (递归或非递归实现，复杂度 $O(\log n)$):

总分 (9 分) = 算法思想 (5 分) + 程序代码 (4 分)

b、 方法一、归并解法 (含其它 $O(n)$ 复杂度方法):

总分 (7 分) = 算法思想 (4 分) + 程序代码 (3 分)

c、 方法三、其它通过排序解法:

总分 (5 分) = 算法思想 (3 分) + 程序代码 (2 分)

d、 算法复杂性与算法思想一致，给 1 分 (含在算法思想分中)，否则没有分。

e、 没有写算法思想，但程序全对，则除程序分外，另外给一半算法思想分。

f、 有少量瑕疵酌情扣减。