

考试科目代码及名称: 839 数据结构(含 C 语言程序设计)

招生专业(领域)名称: 计算机技术

考生注意:

无论以下试题中是否有答题位置, 均应将答案做在考场另发的答题纸上(写明题号)。

C 语言程序设计部分(共 70 分)

一、回答问题(本大题共 5 小题, 每小题 2 分, 共 10 分)

1. 在 UNIX 或 LINUX 文件系统中, 库函数 creat 的函数原型: `int creat(char *name, int perms);` 中参数 perms 的含义是什么?
2. 关键字 static 修饰局部变量、全局变量、函数的含义分别是什么?
3. 为什么说要避免使用全局变量?
4. 要完成 $n \times n$ 矩阵的转置, 定义函数: `void Transpose(int* m, int n)` 时由于从参数 m 中无法得到矩阵大小信息而使得使用 `m[i][j]` 不能正确访问第 i 行第 j 列的元素。那么如何正确访问第 i 行第 j 列的元素呢?
5. 对于一个元素个数为 n 的未排序数组, 如果使用顺序搜索, 则算法的复杂度可表示为 $O(n)$; 如果使用二分搜索, 虽然搜索的复杂度为 $O(\lg n)$, 但二分搜索的前提是数组已经是排序的, 而好的排序方法复杂度为 $O(n \lg n)$, 使得总的二分搜索复杂度为 $O(n \lg n) + O(\lg n)$ 。显然 $O(n \lg n) + O(\lg n) > O(n)$ 。那我们为什么还要经常用二分搜索呢?

二、阅读分析程序并写出程序运行结果(10 分)

```
#include <stdio.h>
int max(int a, int b, int c) { a = a > b ? a : b; return a > c ? a : c; }
int f(int* p, int c1, int c2, int c3)
{ if (c1 < 0 || c2 < 0 || c3 < 0) return 0;
  return max(f(p, c1-1, c2, c3), f(p, c1, c2-1, c3), f(p, c1, c2, c3-1)) + p[c1+c2*2+c3*3];
}
int main() { int p[] = {6, 10, 14, 2, 8, 8, 18, 5, 17}; printf("%d\n", f(p, 1, 2, 1)); return 0; }
```

三、回答问题和按要求写函数定义(第1-2小题各5分、第3小题10分, 共20分)

1. 二分查找法中如果定义指向 first 和 last 的指针, 计算指向 first 和 last 中间元素的指针 middle, 代码 `int *first, *last, *middle; middle = (first+last)/2;` 是否正确? 如果不正确, 请说明错误原因, 并写出正确的语句。
2. 二分查找法中如果定义数组下标 first 和 last, 计算 first 和 last 中间元素的下标 middle, 代码 `int first, last, middle; middle = (first+last)/2;` 是否正确? 如果不正确, 请说明错误原因, 并写出正确的语句。
3. 请写出二分查找算法的函数代码。

函数原型: `void* bsearch(void *key, void *base, size_t num, size_t width,`

`int (*compare)(const void *elem1, const void *elem2));`

其中 key 为要查找元素的地址, base 为数组的开始地址, num 为数组的元素个数, width 为每个元素占的字节数, compare 为数组排序时使用的比较函数。

四、按要求写函数定义 (10分)

麦当劳出售麦香鸡有三种包装 6 块装、9 块装和 20 块装三种。因此总共购买 15 块是可能的 (6 块装 1 包+9 块装 1 包)，但买 16 块是不可能的。定义函数 MCNuggets(n) 判断总共购买 n 块是否可能。n 为非负整数。

即：能否找出三个非负整数 a, b, c 满足： $6a+9b+20c = n$

例如 MCNuggets(15) 为 1 表示可以买到 (取 a=b=1, c=0)，而 MCNuggets(16) 为 0 表示不可能买到。

函数原型： `int MCNuggets(int n);`

函数返回 1 表示可能，0 表示不可能

五、按要求写程序 (20 分)

二代身份证号码共 18 位，其中最后一位是校验码。

校验码计算方法描述如下：

假设某身份证的前 17 位数字为 12345678912345678。

第 1 步：计算前 17 位各位数字与对应加权因子乘积的和 S。

| | | | | | | | | | | | | | | | | | |
|-----------|---|---|----|---|---|---|---|---|---|---|---|---|----|---|---|---|---|
| 前 17 位数字： | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| 加权因子： | 7 | 9 | 10 | 5 | 8 | 4 | 2 | 1 | 6 | 3 | 7 | 9 | 10 | 5 | 8 | 4 | 2 |

$S = 1*7+2*9+3*10+4*5+5*8+6*4+7*2+8*1+9*6+1*3+2*7+3*9+4*10+5*5+6*8+7*4+8*2 = 416$

第 2 步：计算 S 除 11 的余数 T： $T = 416 \% 11 = 9$ 。

第 3 步：将余数 T 转换为 1 位校验码。以下是转换表：

| | | | | | | | | | | | |
|-------|---|---|---|---|---|---|---|---|---|---|----|
| 余数 T: | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|-------|---|---|---|---|---|---|---|---|---|---|----|

| | | | | | | | | | | | |
|------|---|---|---|---|---|---|---|---|---|---|---|
| 校验码: | 1 | 0 | X | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 |
|------|---|---|---|---|---|---|---|---|---|---|---|

2 对应 X，所以我们会发现有人的身份证号码的最后一位是 X。

9 对应 3，所以例中身份证的校验码为 3，18 位身份证号码就应该是 123456789123456783。

写一个程序，输入一个 18 位身份证号码 (注意最后一位可能是大写字母 X)，检查这个号码是否合法。

若长度不对、前 17 位不是数字或校验码不正确就认为是不合法的号码 (本题不考虑其它的非法情况，如身份证中出生日期非法等)。

最后输出一行信息指出输入的身份证号码是否合法。

数据结构部分 (共 80 分)

六、分别写出判断一个顺序栈 $s[8]$ 为空、为满的条件, 并画出 $s[8]$ 分别为空、为满时的相应示意图。(8 分)

七、已知一个链队的头指针、尾指针分别为 $head$ 和 $tail$, 写出它的出队函数。(12 分)

八、一个串 s 采用顺序存储结构, 写一函数删除 s 中从第 i 个字符开始的共 j 个字符。(10 分)

九、已知一组数 $(9, 2, 8, 6, 1)$, 分别采用插入排序和合并排序进行排序, 写出排序过程中每遍处理后的结果。
(10 分)

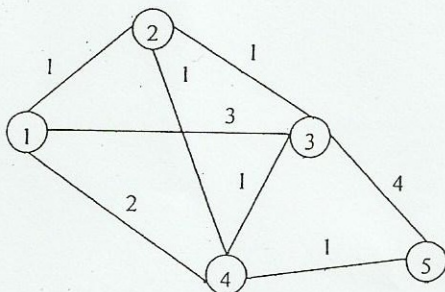
十、写出数组 $d[5][4][3][6]$ 的地址公式。(每个元素占 s 个存储单元)。(8 分)

十一、一组数如下图所示, 画出对这组数进行堆排序过程中所建立的初始堆。(10 分)

| | | | | | |
|---|---|---|---|---|----|
| 7 | 9 | 8 | 2 | 5 | 15 |
|---|---|---|---|---|----|

十二、(1) 写出下图从顶点 1 出发的一个广度优先搜索序列, 并画出相应的广度优先生成树。(5 分)

(2) 用 Prim 算法求出下图的一棵最小生成树, 请画出求解的过程。(5 分)



十三、已知一棵 5 次树 T 采用标准存储结构 (指向树根的指针为 t), 写一函数对 T 进行后序遍历。(12 分)