

西安电子科技大学(大学)

2018 年硕士研究生招生考试初试试题

考试科目代码及名称 902 数据结构与 C 语言程序设计

考试时间 2017 年 12 月 24 日下午 (3 小时)

答题要求: 所有答案(填空题按照标号写)必须写在答题纸上, 写在试题上一律作废, 准考证号写在指定位置!

一. 单项选择题 (每题 2 分, 共计 20 分)

1. 语句频度是指语句重复执行的次数, 下面代码段中 (n 为正整数), 注释为#的语句的频度为 ()。

```
x = 1;  
for(i=0; i<n; i++)  
    for(j=0; j<i; j++)  
        x = x + 1;           // #
```

A. $\frac{n(n-1)}{2}$ B. $\lfloor n \log n \rfloor$ C. $n(n-1)$ D. $\frac{n(n+1)}{2}$

2. 用顺序存储结构表示线性表时, ()。

- A. 按值访问表中元素的时间复杂度为 $O(1)$
- B. 按序号访问表中元素的时间复杂度为 $O(1)$
- C. 在表中任意位置插入元素的时间复杂度为 $O(1)$
- D. 在表中任意位置删除元素的时间复杂度为 $O(1)$

3. 用带头结点的单链表表示队列且设置队头指针指向头结点、队尾指针指向队尾元素结点。进行出队列运算时, ()。

- A. 仅修改队头指针, 无需修改队尾指针
- B. 仅修改队尾指针, 无需修改队头指针
- C. 无需修改队头指针, 可能需要修改队尾指针
- D. 无需修改队尾指针, 可能需要修改队头指针

4. 执行 KMP 模式匹配过程需要先计算模式串的 next 函数值。模式串“abcaabac”的 next 函数值为 ()。

- A. 01122331 B. 01234567
- C. 01112232 D. 01123122

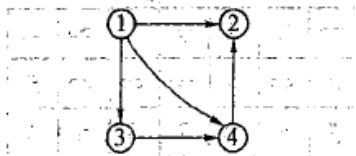
5. 在度为 3 的树中, 度为 2 的结点数是 4, 度为 1 的结点数是 2, 度为 0 的结点数是 7, 则度为 3 的结点数是 ()。

- A. 1 B. 2 C. 5 D. 6

6. 某字符集中各字符的使用频率为{0.01, 0.13, 0.38, 0.08, 0.15, 0.05, 0.20}, 采用哈夫曼编码构造各字符的编码, 则其字符编码长度的最小值和最大值分别为 ()。

- A. 1, 4 B. 2, 4 C. 1, 5 D. 2, 5

7. 以下关于图的存储结构的说法中, 正确的是 ()。



- A. 无向图适合采用邻接矩阵存储, 有向图适合采用邻接表存储
B. 有向图适合采用邻接矩阵存储, 无向图适合采用邻接表存储
C. 稀疏图适合采用邻接矩阵存储, 稠密图适合采用邻接表存储
D. 稠密图适合采用邻接矩阵存储, 稀疏图适合采用邻接表存储

8. 以下关于散列表(或哈希表)的说法中, 正确的是 ()。

- A. 用链地址法解决冲突构造散列表时, 查找最小元素的时间复杂度为 $O(1)$
B. 用线性探查法解决冲突构造散列表时, 查找最小元素的时间复杂度为 $O(1)$
C. 用链地址法解决冲突构造散列表时, 容易造成聚集现象
D. 用线性探查法解决冲突构造散列表时, 容易造成聚集现象

9. 设平衡二叉树的高度为 6, 则其结点数至少应为 ()。

- A. 19 B. 20 C. 21 D. 22

10. 以下排序方法中, 算法时间复杂度最高的是 ()。

- A. 简单选择排序 B. 快速排序 C. 堆排序 D. 归并排序

二. 简答题 (每题 5 分, 共计 20 分)

1. 设有两个栈 S1 和 S2 以及判栈空、入栈和出栈操作, 是否可以用这两个栈实现一个队列 Q, 若能, 请简要说明如何进行队列 Q 的入队列和出队列操作, 否则说明原因。

2. 设有一个上三角矩阵 $A_{n \times n}$, 对该矩阵按行压缩存储 (包括对角线元素) 至一维数组 B[] 中, 即矩阵元素 $A_{0,0}$ 存储在 B[1], $A_{0,1}$ 存储在 B[2], 依次类推, $A_{i,j}$ ($0 \leq i, j < n, i \leq j$) 存储在 B[k], 请给出 k 与 i、j 的关系 (即用 i、j 来表示 k 的算式)。

3. 在非空二叉树中, 若叶子结点的个数为 n_0 , 具有两个非空子树的结点个数为 n_2 , 请证明 $n_0 = n_2 + 1$ 。

4. 简要说明拓扑排序的方法以及如何利用拓扑排序结果判断有向图中是否存在回路。

三. 综合题 (每题 10 分, 共计 50 分)

1. 已知某二叉树的先序、中序遍历序列分别为 ABDGHCEF、GDHBAEFC

- (1) 请画出该二叉树, 并给出其层序遍历序列;
- (2) 请给出该二叉树采用顺序存储表示的示意图 (或表);
- (3) 画出由该二叉树转换所得的树 (或森林)。

2. 某图 (顶点依次表示为 a、b、...、g) 的邻接矩阵如下所示:

∞	8	5	9	∞	∞	∞
8	∞	5	∞	∞	∞	5
5	5	∞	7	6	5	4
9	∞	7	∞	3	∞	∞
∞	∞	6	3	∞	2	∞
∞	∞	5	∞	2	∞	6
∞	5	4	∞	∞	6	∞

- (1) 写出该图的邻接表表示;
- (2) 根据该图的邻接矩阵, 给出从顶点 a 出发进行广度优先遍历的生成树;
- (3) 用普里姆算法求解该图的最小生成树, 画出最小生成树并按照构成最小生成树时边加入的顺序写出边序列 (初始时顶点 a 在最小生成树中)。

3. 对于关键码序列 (41, 23, 60, 35, 12, 52, 49, 6, 87)

- (1) 给出根据该序列构造的二叉排序树 (二叉查找树);
- (2) 请分别计算在该二叉排序树上查找成功和查找不成功的平均查找长度 (列出算式即可);
- (3) 根据该序列构造平衡二叉树, 给出首次出现不平衡情况时的最小不平衡子树, 以及最后构造的平衡二叉树。

4. 设哈希函数 $H(\text{key}) = (3 \times \text{key}) \% 11$ 。试在 0~10 的散列地址空间中对关键码序列 (11, 30, 9, 35, 19, 2, 7, 23, 16) 构造散列表 (哈希表)。

- (1) 用线性探查法解决冲突, 给出所构造的散列表;
- (2) 给出查找关键码 18 的过程;
- (3) 在等概率情况下求查找成功时的平均查找长度 (列出算式即可)。

5. 对于关键码序列 (38, 27, 96, 25, 6, 90, 17, 84, 62, 49)

- (1) 以第一个元素为基准 (枢轴), 写出对该序列进行快速排序第一趟划分的结果, 以及再次对右子序列进行划分的结果;
- (2) 若对关键码序列 (或子序列) 进行划分的条件是 (子) 序列长度大于 1, 在对关键码序列进行快速排序的过程中, 共进行多少次划分?
- (3) 如果关键码序列长度为 n 且初始序列为降序, 则完成关键码升序排列的快速排序过程中共进行多少次划分 (划分条件同 (2))?

四. 算法题 (共计 10 分)

设图 G 采用邻接矩阵存储 (G 中边或弧的权值为整型), 请给出表示 G 存储方式的类型定义, 并编写对 G 进行深度优先遍历的算法 (语言不限), 输出所访问顶点的序号。

五. 程序填空题 (每空 2 分, 共计 10 分)

1. 下面程序的功能是调用函数 func() 计算字符串的长度, 请补充空缺处的代码。

```
#include <stdio.h>
int func(____(1)____)
{
    char *p = s;
    while (____(2)____)
        p++;
    return ____ (3) ____;
}

int main()
{
    printf("%d", func("abcdefg"));
    return 0;
}
```

2. 函数 isHeap 的功能是判断数组 a 的元素序列是否满足堆 (大根堆或小根堆) 的定义, 是则返回 1, 否则返回 0。请补充空缺处的代码。

```
#include <stdio.h>
#include <stdio.h>
int isHeap(int a[], int n, int (*cmp)(int, int))
{
    int f, lc;
    for(f=0, lc=2*f+1; lc<n; f=lc, ____ (4) ____){
        if (lc+1<n && cmp(a[lc+1], a[lc])) lc++;
        if (____ (5) ____) return 0; //不是堆
    }
    return 1;
}

int less(int x, int y)
{
    return x<y;
}

int large(int x, int y)
{
    return x>y;
}
```

```

int main()
{
    int a[] = {103, 97, 56, 38, 66, 23, 42, 12, 30, 52, 6, 20};
    int n = sizeof(a)/sizeof(int);

    if (isHeap(a, n, less))    printf("Min Heap. \n");
    else if (isHeap(a, n, large)) printf("Large Heap. \n");
    else                        printf("Not Heap. \n");
    return 0;
}

```

六. 程序阅读题（每题 5 分，共计 20 分）

1. 阅读以下代码，写出运行时的输出结果。

```

#include <stdio.h>
int main()
{
    int i, j, k;
    for(i=1; i<3; i++)
        for(j=1; j<4; j++)
            for(k=1; k<5; k++) {
                if (i!=k&&i!=j&&j!=k)
                    printf("%d %d %d\n", i, j, k);
            }
    return 0;
}

```

2. 阅读以下代码，写出运行时的输出结果。

```

#include <stdio.h>
void foo( int A[], int left, int right )
{
    if ( left == right ) return;
    for ( ; left<right; left++,right-- ) {
        int temp = A[left]; A[left] = A[right]; A[right] = temp;
    }
}

void shift(int A[], int n, int k)
{
    foo( A, 0, n-1 );
    foo( A, 0, k-1 );
    foo( A, k, n-1 );
}

int main()
{
    int a[] = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10};
    int i, n = sizeof(a)/sizeof(int);
}

```

```

    shift(a, n, 6);
    for(i=0; i<n; i++)
        printf("%d ", a[i]);

    printf("\n");
    return 0;
}

```

3. 阅读以下代码，写出程序运行时的输出结果。

```

#include <stdio.h>
#include <stdlib.h>
typedef struct Node{
    int data;
    struct Node *next;
}Node, *NodePtr;

NodePtr create(void)
{
    NodePtr head;
    head = (NodePtr) malloc(sizeof(Node));
    if (!head)
        return NULL;
    head->data = 1;    head->next = NULL;
    for(int i=1; i<5; i++) {
        NodePtr p = (NodePtr) malloc(sizeof(Node));
        if (!p)
            break;
        p->next = head->next;
        p->data = i*2;    head->next = p;
    }
    return head;
}

void output(NodePtr p)
{
    for(;p; p=p->next) {
        printf("%d ", p->data);
    }
    printf("\n");
}

int main()
{
    output(create());
    return 0;
}

```

4. 阅读以下代码，说明函数 foo 的功能。

```

void foo(int n, int (*a)[n])

```

```

{
    int i, j;

    for(i=0; i<n; i++)
        for(j=i; j<n; j++)
            {int tmp=a[i][j];
              a[i][j] = a[j][i];
              a[j][i] = tmp;
            }
}

```

七. 编程题 (每小题 10 分, 共计 20 分)

1. 编写一个函数, 将一个字符串中的重复字符去除, 即使得字符串中的所有字符都不相同, 且各字符的顺序保持不变并连续存储。

2. 文本数据文件 “exprec.dat” 中记录了某俱乐部会员在一段时间内的消费流水账, 每行一条消费记录 (包括 18 位身份证号和消费金额), 请编写一个程序, 尽可能快速地统计出消费总额最多的会员, 输出其身份证号和消费总额。(注: 该俱乐部会员数不超过 10000, 一名会员可能有多条消费记录, 若消费总额最多的会员有多名, 输出其中任意一位的身份证号和消费总额即可)

西安电子科技大学

2017 年硕士研究生招生考试初试试题

考试科目代码及名称 902 数据结构与 C 语言程序设计

考试时间 2016 年 12 月 25 日下午 (3 小时)

答题要求: 所有答案(填空题按照标号写)必须写在答题纸上, 写在试题上一律作废, 准考证号写在指定位置!

一. 单项选择题 (每题 2 分, 共计 20 分)

1. 语句频度指语句重复执行的次数, 下面代码段中 (n 为正整数), 注释为#的语句的频度为_①_。

```
x = 0;  
while ((x+1)*(x+1)<n)  
    x = x + 1;    // #
```

- A. $\left\lfloor \frac{n}{2} \right\rfloor$ B. $\lfloor \sqrt{n} \rfloor$ C. $\lfloor \log n \rfloor$ D. n

2. 线性表采用循环单向链表存储的特点是_②_。

- A. 可以随机存取表中的任何一个元素
B. 逻辑上相邻的元素在物理位置上必相邻
C. 访问表中任何一个元素都需要遍历整个链表
D. 在表中插入一个新元素时不移动其他元素

3. 对于 3 个互异的元素, 按照指定顺序入栈(栈初始和结束时为空), 通过合法的栈操作, 可得到_③_种不同的出栈序列。

- A. 1 B. 3 C. 5 D. 6

4. 设某 n 阶三对角矩阵 $A_{n \times n}$ 的示意图如下图所示。若将该三对角矩阵的非零元素按行存储在一维数组 $B[k](1 \leq k \leq 3 \cdot n - 2)$ 中, 则 k 与 i, j 的对应关系是_④_。

$$A_{n \times n} = \begin{bmatrix} a_{1,1} & a_{1,2} & & & & \\ a_{2,1} & a_{2,2} & a_{2,3} & & & \\ & a_{3,2} & a_{3,3} & a_{3,4} & & \\ & & \dots & \dots & \dots & \\ 0 & & & a_{i,j} & & \\ & & & \dots & \dots & \dots \\ & & & & a_{n,n-1} & a_{n,n} \end{bmatrix}$$

A. $k = 2i + j - 2$

B. $k = 2i - j + 1$

C. $k = 3i + j - 1$

D. $k = 3i - j + 2$

5. 以下关于字符串的叙述中, 正确的是_⑤_。

A. 长度为 0 的字符串称为空白串

B. 字符串中元素的逻辑关系是线性的

C. 串的模式匹配是指求出两个指定串的最长公共子串

D. 字符串采用链表存储更有利于其基本运算的实现

6. 某完全二叉树中的结点个数为 87, 可计算得知其叶子结点数目为_⑥_。

A. 24

B. 43

C. 44

D. 63

7. 在 55 个互异元素构成的有序表 $M[1..55]$ 中进行折半查找(或二分查找, 向下取整)。若需查找的元素等于 $M[19]$, 则在查找过程中参与比较的元素依次为 $M[28]$ 、_⑦_、 $M[19]$ 。

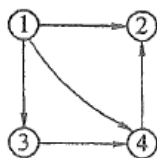
A. $M[30]$ 、 $M[15]$ 、 $M[20]$

B. $M[14]$ 、 $M[22]$ 、 $M[20]$

C. $M[15]$ 、 $M[22]$ 、 $M[18]$

D. $M[14]$ 、 $M[21]$ 、 $M[17]$

8. 某有向图 G 如下所示, 其拓扑序列是_⑧_。



A. ①②③④

B. ①③②④

C. ①③④②

D. ①④③②

9. 在平衡二叉查找树中, _⑨_。

A. 任意结点的左、右子树最多相差 1 层

B. 任意结点的左、右子树高度相同

C. 任意结点的左、右子树结点数目相同

D. 除叶子外的所有结点都有两个子结点

10. 在待排序的一组关键码序列 k_1, k_2, \dots, k_n 中, 若 k_i 和 k_j 相同, 且在排序前 k_i 领先于 k_j , 那么进行排序后, 如果能确保 k_i 和 k_j 的相对次序保持不变, 即 k_i 仍领先于 k_j , 则称此类排序为稳定的, 反之则称此类排序为不稳定的。_⑩_是稳定的排序方法。

A. 快速排序

B. 简单选择排序

C. 堆排序

D. 归并排序

二. 简答题 (每题 5 分, 共计 20 分)

1. 线性表常采用单链表存储, 简要说明头指针和头结点在单链表中的作用。

2. 简要说明什么是循环队列, 若循环队列的容量为 Maxsize, 且队头元素由 front 指示, 队尾元素由 rear 指示, 请给出表示队列长度 length 的算式。

3. 有一份电文中只使用下列字符: a、b、c、x、y、z, 它们的出现频率依次为 0.20、0.09、0.35、0.11、0.18、0.07, 试画出相应的哈夫曼树并为每个字符构造哈夫曼编码。

4. 设无向连通图 G 中含有 n 个顶点、e 条边, 请给出求解最小生成树的普里姆(Prim)算法和克鲁斯卡尔(Kruskal)算法的时间复杂度。当 G 为边稠密的图时, 选择其中的哪种算法更恰当?

三. 综合题 (每题 10 分, 共计 50 分)

1. 已知某二叉树的后序遍历序列为 3 6 5 4 2 8 7 1, 中序遍历序列为 3 2 4 6 5 1 7 8

(1) 画出这棵二叉树;

(2) 画出与该二叉树对应的中序线索二叉树;

(3) 设指针 p 指向中序线索二叉树中的某结点, 请说明如何找出该结点在中序遍历中的直接后继结点。

2. 某图的邻接矩阵如下所示:

∞	∞	∞	8	∞	4
∞	∞	∞	∞	3	∞
∞	∞	∞	6	∞	∞
∞	∞	∞	∞	9	∞
∞	∞	4	∞	∞	∞
∞	5	∞	∞	10	∞

(1) 画出该图, 用 a、b、...表示顶点;

(2) 写出该图的邻接表;

(3) 写出从顶点 a 出发的深度优先遍历序列和广度优先遍历序列。

3. 已知关键字序列为 (Kim, Eva, Jon, Tim, Roy, Dot, Amy, Jim, Kay, Tom, Ann, Guy)

(1) 画出根据该序列构造的二叉排序树 (二叉查找树);

(2) 分别给出查找 Jim 和 Max 的过程;

(3) 请分别计算查找成功和查找不成功时的平均查找长度 (列出算式)。

4. 根据关键字序列 (21, 33, 45, 15, 63, 87, 54, 49, 10) 构造散列表 (哈希表), 设散列表的长度为 13 (即散列表的地址范围为 0~12), 散列函数为 $H(k) = k \text{ MOD } 11$

(1) 画出用线性探查法解决冲突所构造的散列表;

(2) 给出查找关键字 10 和 26 的过程;

(3) 假定每个关键字的查找概率相同, 求查找成功时的平均查找长度。

5. 对于关键码序列 (58, 22, 69, 46, 95, 34, 88, 37, 73, 41)

(1) 以第一个元素为基准 (枢轴), 写出对该序列进行快速排序第一趟划分的结果,

以及再次对左子序列进行划分的结果;

(2) 以二叉树形式写出将其调整为大顶堆(或大根堆)的过程。

四. 算法题 (共计 10 分)

编写算法判断一个程序语句中的括号是否匹配, 假设语句已经保存在字符数组 statement 中, 括号字符为 '{'、'}'、'('、')'、'['、']'。(可以直接使用栈的基本操作, 需加注释, 语言不限)

五. 程序填空题 (每空 2 分, 共计 10 分)

1. 函数 reverse(int x) 将给定整数 x 逆置后返回 (假定 x 的值逆置后不溢出)。例如, -1234 逆置后得到 -4321。

```
int reverse(int x)
{
    int sign = 1;
    int ret = 0;

    if (x < 0) {x = -x; (1);}
    while (x) {
        ret = ret * 10 + (2);
        x = (3);
    }
    return ret * sign;
}
```

2. 函数 DelDupElem(int a[], int n) 在元素个数为 n 的数组 a 中删除重复出现的元素值, 最后返回数组中互异的元素个数 k。处理过程中, 对每个值仅保留首次出现者, 并使得保留下来的值按照原顺序存放在数组的前 k 个元素中。

```
int DelDupElem(int a[], int n)
{
    int i, j, k=0; //用 i 遍历数组 a 的元素, k 表示数组中互异的元素个数
    for(i=1, k=1; i<n; i++) {
        for(j=0; j<k&& (1); j++); //判断 a[i] 的值是否在前面出现过
        if (j==k) { //a[i] 的值在前面没出现过, 需保留
            (2) = a[i];
        }
    }
    return k;
}
```

六. 程序阅读题 (每题 4 分, 共计 20 分)

1. 阅读以下代码, 说明函数 fun1 的功能。

```
int fun1(char *message, char *word)
{
    int n, m, i, k, count = 0;
    char *tmp;

    n = strlen(message); m = strlen(word);
    tmp = (char *)malloc(m+1);

    for(i=0; i<n-m+1; i++) {
        for(k=0; k<m; k++)
            tmp[k] = message[i+k];
        tmp[m] = '\0';
        if (0==strcmp(tmp, word)) {
            i += m; count++;
        }
        else
            i++;
    }
    free(tmp);
    return count;
}
```

2. 阅读以下代码, 说明函数 fun2 的功能。

```
void place_largest(int array[], int n)
{
    int temp, j, maxIndex;
    maxIndex = n - 1;
    for(j = n - 2; j >= 0; --j)
        if (array[j] > array[maxIndex])
            maxIndex = j;

    if (maxIndex != n - 1) {
        temp = array[n - 1]; array[n - 1] = array[maxIndex];
        array[maxIndex] = temp;
    }
}

void fun2(int array[], int n)
{
    if (n > 1) {
        place_largest(array, n);
        fun2(array, n - 1);
    }
}
```

3. 阅读以下代码，写出程序运行时的输出结果。

```
#include <stdio.h>
int main()
{
    int v[2][5] = {{5, 4, 3, 2, 1}, {10, 20, 30, 40, 50}};
    int *s, (*p)[5];
    s = &v[1][1];
    printf("%d\n", *s);
    s++;
    printf("%d\n", *s);
    p = v;
    printf("%d\n", **p);
    p++;
    printf("%d\n", **p);
    return 0;
}
```

4. 阅读以下代码，写出程序运行时的输出结果。

```
#include <stdio.h>
typedef struct {
    int key;
    int next;
} SLNode;
int main()
{
    SLNode tmp, tb[] = {{0, 2}, {49, 3}, {27, 5}, {65, 4}, {93, 0}, {32, 1}};
    int i, p, s;
    int size = sizeof(tb) / sizeof(SLNode);
    p = tb[0].next;
    for(i=1; i<size; i++) {
        while (p<i) p = tb[p].next;
        s = tb[p].next;
        if (p!=i) {
            tmp = tb[p]; tb[p] = tb[i]; tb[i] = tmp;
            tb[i].next = p;
        }
        p = s;
    }

    for(i=0; i<size; i++)
        printf("(%d,%d) ", tb[i].key, tb[i].next);

    printf("\n");
    return 0;
}
```

5. 阅读以下代码, 说明程序的功能。

```
#include <stdio.h>
#include <math.h>
int main()
{
    FILE *fin, *fout;
    int a;
    fin = fopen("data.in", "r");
    fout = fopen("data.out", "w");
    if (!fin || !fout){
        printf("open file error!\n");
        return 0;
    }
    while(!feof(fin)) {
        if (fscanf(fin, "%d", &a) != 1)
            break;
        if (a > 0)
            fprintf(fout, "%12.2lf", sqrt(a));
    }
    fclose(fin); fclose(fout);
    return 0;
}
```

七. 编程题 (每小题 10 分, 共计 20 分)

1. 编写函数, 返回一个字符串逆转后构成的串, 如“examiner”逆转为“renimaxe”, 原字符串不修改。

2. 一个合法的身份证号码由 17 位地区、日期编号、顺序编号和 1 位校验码组成。校验码的计算规则如下: 首先对前 17 位数字加权求和, 权重分配为: {7, 9, 10, 5, 8, 4, 2, 1, 6, 3, 7, 9, 10, 5, 8, 4, 2}; 然后将计算的和对 11 取模得到值 Z; 按照以下关系对应 Z 值与校验码 M 的值:

Z	0	1	2	3	4	5	6	7	8	9	10
M	1	0	X	9	8	7	6	5	4	3	2

编写程序, 从某文本数据文件中读入若干身份证号码, 根据以上规则验证每个身份证号码的有效性, 输出所有非法的身份证号码。注: 在该数据文件中, 每行一个身份证号。

西安电子科技大学

2016 年硕士研究生招生考试初试试题

考试科目代码及名称 902 数据结构与 C 语言程序设计

考试时间 2015 年 12 月 27 日下午 (3 小时)

答题要求: 所有答案 (填空题按照标号写) 必须写在答题纸上, 写在试题上一律作废, 准考证号写在指定位置!

一. 单项选择题 (每题 2 分, 共计 20 分)

1. 循环队列存储在数组 $A[0..m]$ 中, 用 $front$ 和 $rear$ 分别表示队头和队尾, 则入队时的操作为_____。

A. $rear = rear + 1$

B. $rear = (rear - front) \bmod (m + 1)$

C. $rear = (rear - front) \bmod m$

D. $rear = (rear + 1) \bmod (m + 1)$

2. 双向链表中, 在 p 所指向的结点前插入一个新结点 s 的操作是_____。

A. $p \rightarrow prior = s; s \rightarrow next = p; p \rightarrow prior \rightarrow next = s; s \rightarrow prior = s;$

B. $p \rightarrow prior = s; p \rightarrow prior \rightarrow next = s; s \rightarrow next = p; s \rightarrow prior = p \rightarrow prior;$

C. $s \rightarrow next = p; s \rightarrow prior = p \rightarrow prior; p \rightarrow prior \rightarrow next = s; p \rightarrow prior = s;$

D. $s \rightarrow prior = p \rightarrow prior; s \rightarrow next = s; p \rightarrow prior = s; p \rightarrow prior \rightarrow next = s;$

3. 若已知一个栈的入栈序列是 $1, 2, 3, \dots, n$, 其输出序列为 $p_1, p_2, p_3, \dots, p_n$ 。若 p_1 是 n , 则 p_i 是_____。

A. $n - i + 1$

B. $n - i$

C. i

D. 不确定

4. 若一个栈以向量 $V[1..n]$ 存储, 初始栈顶指针 top 为 $n + 1$, 则将元素 x 入栈的正确操作是_____。

A. $top = top + 1; V[top] = x;$

B. $top = top - 1; V[top] = x;$

C. $V[top] = x; top = top + 1;$

D. $V[top] = x; top = top - 1$

5. 已知某二叉树的中序序列为 CBDAEFI, 先序序列为 ABCDEFI, 则该二叉树的高

度为_____。

A. 2

B. 3

C. 4

D. 5

6. 用邻接表表示图进行广度优先遍历时, 通常是采用_____来实现算法的。

A. 栈

B. 队列

C. 树

D. 图

7. 若一棵二叉树具有 10 个度为 2 的结点, 5 个度为 1 的结点, 则度为 0 的结点个数是_____。

A. 9

B. 10

C. 11

D. 12

8. 下列图中, _____的邻接矩阵是对称矩阵。

A. 有向图

B. AOV 网

C. 无向图

D. AOE 网

9. 既希望较快的查找又便于线性表动态变化的查找方法是_____。

A. 顺序查找

B. 折半查找

C. 索引顺序查找

D. 哈希法查找

10. 对一待排序列进行直接插入排序和简单选择排序, 若序列中有两个元素的排序码相同, 则_____保证这两个元素在排序前后的相对位置不变。

A. 直接插入排序和简单选择排序都可以

B. 直接插入排序和简单选择排序都不能

C. 简单选择排序可以

D. 直接插入排序可以

二. 简答题 (每题 5 分, 共计 20 分)

1. 由二叉树的前序遍历和后序遍历结果能否唯一确定一棵二叉树? 解释你的论断。

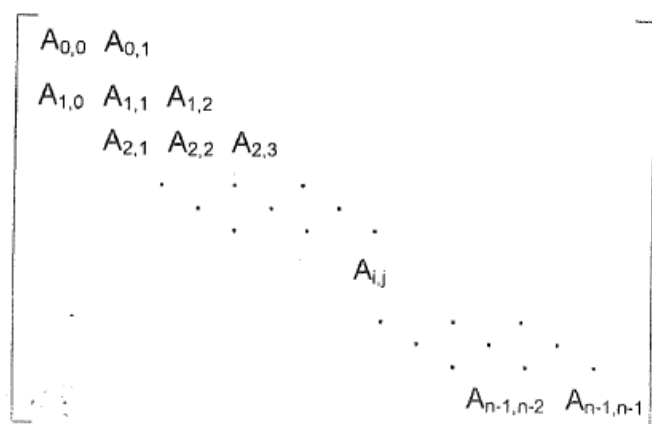
2. 在采用线性探测法处理冲突的哈希表中, 所有同义词在表中是否一定相邻? 请说明原因。

3. 假设用于通信的电文仅由 A、B、C、D、E、F 这 6 个字母组成，这些字母在电文中出现的频率分别为 0.20、0.04、0.15、0.10、0.37、0.14。分别列出 A、B、C、D、E、F 的哈夫曼编码 (Huffman)。

4. 设有 n 阶的三对角矩阵如下图所示，将该三对角矩阵按行优先压缩存储在数组 $B[1..M]$ 中，如果每个元素占两个存储单元，请问：

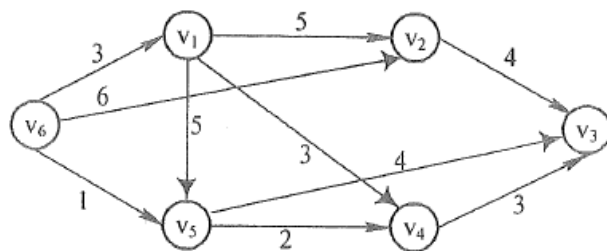
(1) M 的值至少为多少？ $A_{3,3}$ 存储在数组 B 的哪个元素中？

(2) 若 $A_{0,0}$ (即 $B[1]$) 的起始地址为 $base$ ，那么元素 A_{ij} ($0 \leq i \leq n-1, i-1 \leq j \leq i+1$) 的地址是什么？



三. 综合题 (每题 10 分, 共计 50 分)

1. 某带权图 G 如下所示：



(1) 分别写出该有向图所有可能的拓扑序列。

(2) 将该图看作无向图，用普里姆 (Prim) 算法求其最小生成树 (写出每一步选中的边即可)。

2. 已知某二叉树的后序遍历序列为 EFGCDBA，中序遍历序列为 ECGFADB，试画出这棵二叉树，并画出与该二叉树对应的树 (或森林)。

3. 已知关键字序列为 22, 41, 53, 08, 46, 30, 01, 31, 66

(1) 请给出根据该序列构造的二叉排序树, 要求写出建树的过程;

(2) 请分别求查找成功和查找不成功时的平均查找长度。

4. 设非递减有序表 A、B 的长度分别为 n 和 k, 已知 $0 < k < n$, 且 $A=(a_1, a_2, \dots, a_n)$ 、 $B=(b_1, b_2, \dots, b_k)$, 若对 A 和 B 进行二路归并处理:

(1) 简要说明二路归并的处理过程;

(2) 在什么情况下, 元素间的比较次数最少? 给出在此情况下的归并结果;

(3) 若归并结果为 $a_1, b_1, a_2, b_2, \dots, a_{k-1}, b_{k-1}, a_k, b_k, a_{k+1}, \dots, a_n$, 元素间的比较次数是多少?

5. 设关键字序列为 (Jan, Feb, Mar, Apr, May, June, July, Aug, Sep, Oct, Nov, Dec), 哈希函数 $\text{Hash}(\text{key}) = x \% 13$, 其中 x 表示关键字 key 的前两个字符的 ASCII 码值之和 (例如, $\text{Hash}(\text{Jan}) = (74+97) \% 13 = 2$, $\text{Hash}(\text{Mar}) = (77+97) \% 13 = 5$)。试用拉链法 (或链地址法) 解决冲突构造哈希表, 并计算等概率查找成功和查找不成功的平均查找长度 (写出算式)。提示: 各字符的 ASCII 码值如下表所示。

字符	A	D	F	J	M	N	O	S		a	c	e	o	p	u
ASCII 码值	65	68	70	74	77	78	79	83		97	99	101	111	112	117

四. 算法题 (共计 10 分)

设循环队列的容量为 M, 并定义以域变量 front 和 length 分别表示队头指针和队列长度, 请给出其存储结构的定义, 并写出元素入队列的算法。(给出存储结构定义, 简要说明算法思想。算法用类 C 语言或其他类高级语言描述均可)

五. 程序填空题 (每空 2 分, 共计 10 分)

1. 函数 $f(\text{char } *str, \text{char } del)$ 的功能是: 将非空字符串 str 分割成若干个子字符串并输出, del 表示分割时的标志字符。

例如若 str 的值为“33123333435”, del 的值为‘3’, 调用此函数后, 将输出三个子字符串, 分别为“12”、“4”和“5”。

```

void f(char *str, char del) {
    int i, j, len;
    len = strlen(str);
    i = 0;
    while( i < len ){
        while( _____ (1) _____ ) i++; /*忽略连续的标志字符*/
        /*寻找从str[i]开始直到标志字符出现的一个子字符串*/
        j = i + 1;
        while(str[j] != del && str[j] != '\0') j++;
        _____ (2) _____ = '\0'; /*给找到的字符序列置字符串结束标志*/
        printf("%s\t", &str[i]);
        _____ (3) _____;
    }
}

```

2. 函数 `palindrome(char s[])` 的功能是：判断字符串 `s` 是否为回文字符串，若是，则返回 0，否则返回 -1。若一个字符串顺读和倒读都一样时，称该字符串是回文字符串，例如：“LEVEL”是回文字符串，而“LEVAL”不是。

```

int palindrome(char s[]) {
    char *pi, *pj;
    pi = s;    pj = s + strlen(s) - 1;
    while ( pi < pj && _____ (4) _____ ) { pi++;    pj--; }
    if ( _____ (5) _____ ) return -1;
    else return 0;
}

```

六. 程序阅读题（每题 4 分，共计 20 分）

1. 运行下面程序，其输出是_____。

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

```

void main() {
    static int a[] = {2, 6, 10, 14, 18};
    static int *ptr[] = {&a[0], &a[1], &a[2], &a[3], &a[4]};
    int **p, i;
    for (i = 0; i < 5; i++) a[i] = a[i]/2 + a[i];
    p = ptr;
    printf ("%d\n", *((p+2)));
    printf ("%d\n", *((++p)));
}

```

2. 考虑如下定义, 该函数的作用是_____。

```

void func (int n, char x[][12]) {
    char temp[12];
    int j, item;
    for (item = 0; item < n-1; ++item)
        for (j = item+1; j < n; ++j)
            if (strcmp(x[item], x[j]) > 0) {
                strcpy(temp, x[item]);
                strcpy(x[item], x[j]);
                strcpy(x[j], temp);
            }
}

```

3. 函数 strchr 的功能是_____。

```

char *strchr(char *s, char ch) {
    char *p;
    p = strlen(s) + s;
    while (--p >= s) if (*p == ch) return p;
    return NULL;
}

```

4. 运行下面程序，其输出是_____。

```
#include <stdio.h>
#include <string.h>
void main( ) {
    char *p1, *p2, str[50] = "ABCDEFGH";
    p1 = "abcd";
    p2 = "efgh";
    strcpy(str+1, p2+1);
    strcpy(str+3, p1+3);
    printf("%s", str);
}
```

5. 运行下面程序，其输出是_____。

```
#include <stdio.h>
void main() {
    char str[] = "xyz", *ps = str;
    while(*ps) ps++;
    for(ps--; ps - str >= 0; ps--)
        printf("%s\n", ps);
}
```

七. 编程题（每题 10 分，共计 20 分）

1. 给定某个年、月、日，例如 2015 年 12 月 27 日。计算出这一天是该年的第几天。要求写出计算闰年的函数和计算天数的函数。

2. 某英汉词典文件中包含 N 个记录 (N>1)，每个记录有两个字段：一个是英文单词，另一个是相应的汉语解释。各个记录按英文单词的词典顺序排列，各英文单词并不重复。编写函数，在该英汉词典文件中查找某个英文单词的汉语解释。

西安电子科技大学

2015 年攻读硕士学位研究生入学考试试题

考试科目代码及名称 902 数据结构与 C 语言程序设计

考试时间 2014 年 12 月 28 日下午 (3 小时)

答题要求: 所有答案 (填空题按照标号写) 必须写在答题纸上, 写在试题上一律作废, 准考证号写在指定位置!

一. 单项选择题 (每题 2 分, 共计 20 分)

1. 按_____的不同, 通常可将数据结构分为线性结构和非线性结构两种。

- A. 物理结构 B. 逻辑结构 C. 存储结构 D. 链接结构

2. 算法的_____要求每一步的含义不能有歧义。

- A. 可行性 B. 有限性 C. 通用性 D. 确定性

3. 用栈来检查算术表达式中的括号是否匹配的方法是: 初始栈为空, 从左到右扫描表达式, 遇到“(”就将其入栈, 遇到”)”就执行出栈操作。检查算术表达式“ $((a+b)/(a+b)-c/a)/b$ ”时, _____, 因此该表达式中的括号不匹配。

- A. 栈为空却要进行出栈操作
B. 栈已满却要进行入栈操作
C. 表达式处理已结束, 栈中仍留有字符“(”
D. 表达式处理已结束, 栈中仍留有字符”)”

4. 某线性表最常用的运算是在表尾插入结点和删除尾结点, 则采用_____存储方式最节省时间。

- A. 单链表 B. 仅有头指针的单循环链表
C. 双链表 D. 仅有尾指针的单循环链表

5. 若 X 是二叉中序线索树中一个有左孩子的结点, 且 X 不为根, 则 X 的前驱为_____。

- A. X 的双亲 B. X 的右子树中最左的结点
C. X 的左子树中最右的结点 D. X 的左子树中最右的叶结点

6. 已知 x 是某完全二叉树的一个结点, 若 x 没有左孩子, 则 x 必然没有_____结点。

- A. 父亲 B. 右孩子 C. 左兄弟 D. 右兄弟

7. 在具有 N 个叶子节点的哈夫曼树中, 其节点总数为_____。

- A. $2N+2$ B. $2N-1$ C. $2N+1$ D. $2N$

8. 若某有向图中的顶点不能排成一个拓扑序列, 则可断定该图_____。

- A. 是完全图 B. 是强连通图 C. 有回路 D. 是有向无环图

9. 以下序列不是堆的是_____。

A. (100,85,98,77,80,60,82,40,20,10,66)

B. (100,98,85,82,80,77,66,60,40,20,10)

C. (10,20,40,60,66,77,80,82,85,98,100)

D. (100,85,40,77,80,60,66,98,82,10,20)

10. 快速排序在_____情况下排序效率最差。

- A. 待排序列的数据量太大 B. 待排序列中含有多个相同值
C. 待排序列的元素已基本有序 D. 待排序列的元素值都不相同

二. 简答题 (每题 5 分, 共计 20 分)

1. 假设以 S 和 X 分别表示入栈和出栈的操作:

(1) 对于入栈序列 1, 2, 3, 4, 5, 经过一个初始和终止时都为空的栈后, 可得到下面的哪些出栈序列? 请写出得到这些出栈序列的合法的操作序列。(例如, 得到出栈序列 1, 2, 3, 4, 5 的操作序列为 $SXSXSXSXSXSX$)

① 3, 1, 4, 2, 5 ② 4, 2, 5, 3, 1 ③ 2, 1, 4, 3, 5 ④ 4, 3, 5, 1, 2

(2) 指出判别给定出栈序列是否合法的一般规则。

2. 何为队列的上溢现象和假溢出现象? 解决它们有哪些办法?

3. 试找出分别满足下面条件的所有二叉树:

(1) 前序序列和中序序列相同;

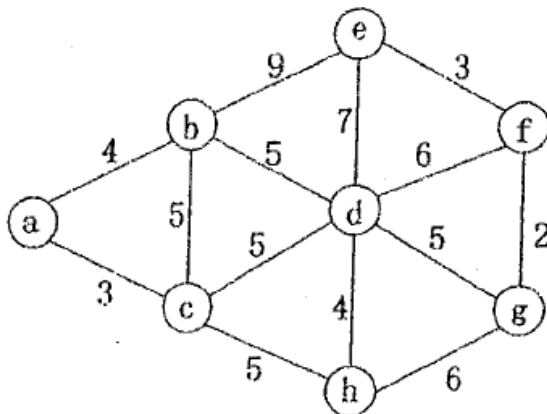
(2) 中序序列和后序序列相同;

(3) 前序序列和后序序列相同。

4. 任意一个有 N 个结点的二叉树, 已知它有 M 个叶子结点。证明非叶子结点中有 $M-1$ 个度为 2, 其余度为 1。

三. 综合题 (每题 10 分, 共计 50 分)

1. 某无向图如下图所示:



- (1) 按照元素字母的顺序写出其对应的邻接矩阵;
- (2) 按克鲁斯卡尔算法求其最小生成树 (给出构造步骤);
- (3) 从顶点 a 出发, 写出用普里姆算法求出的最小生成树 (给出构造步骤)。

2. 已知某二叉树的先序遍历序列为 ABCDEFGHIJ, 中序遍历序列为 BCEDAGHJIF, 则:

- (1) 画出这棵二叉树, 并画出与其对应的中序线索二叉树;
- (2) 画出与该二叉树对应的树或森林。

3. 已知关键字序列为 46, 88, 40, 25, 70, 58, 101, 11, 36

- (1) 给出根据该序列构造的二叉排序树, 要求写出建树的过程;
- (2) 分别给出查找 88 和 16 的查找过程;
- (3) 请分别求查找成功和查找不成功时的平均查找长度。

4. 对于关键字序列 (28, 07, 39, 10, 65, 14, 61, 17, 50, 21):

- (1) 以第一个元素为基准 (枢轴), 写出进行快速排序的第一趟划分的结果。
- (2) 将该序列调整成小顶堆 (小根堆), 以二叉树形式写出序列的初始排列情况和形成小顶堆之后的排列情况。

5. 设散列表的长度为 13, 散列函数为 $H(k) = k \text{ MOD } 13$, 给定的关键字序列为: 19, 14, 23, 01, 68, 20, 84, 27, 55, 11, 10, 79。分别画出用链地址法 (拉链法) 和线性探测再散列解决冲突时所构造的散列表, 并求出在等概率情况下, 这两种方法的查找成功和查找不成功的平均查找长度。

四. 算法题 (共计 10 分)

设计实现在单链表中删除值相同的多余结点的算法。(给出存储结构定义、简要说明算法思想。算法用类 C 语言或其他高级语言描述即可)

五. 程序填空题 (每空 2 分, 共计 10 分)

1. 函数 `char *strrchr(char *s, char ch)` 的功能是在字符串 `s` 中寻找字符 `ch`。若 `ch` 出现在字符串 `s` 中, 则返回最后一次出现时的位置, 否则返回 `NULL`。

```
char *strrchr(char *s, char ch)
{
    char *p;
    p = ____ (1) ____; /*p 指向字符串 s 的结束标志*/
    while (--p >= s)
        if (____ (2) ____) return p;
    return NULL;
}
```

2. 设由 `hi-lo+1` 个元素构成的一个有序表存放在数组元素 `A[lo]~A[hi]` 中, 函数 `int Bsearch(int A[], int k, int lo, int hi)` 的功能是查找元素 `k` 是否在 `A[lo]~A[hi]` 之中, 若是, 函数返回相应数组元素的下标; 否则, 返回值 `-1`。

```
int Bsearch(int A[], int k, int lo, int hi)
{
    int mid;
    if (____ (1) ____) return -1;
    mid = (lo + hi)/2;
    if (A[mid] == k) return mid;
    if (A[mid] > k) return Bsearch(A, k, ____ (2) ____);
    else return Bsearch(A, k, ____ (3) ____);
}
```

六. 程序阅读题 (每题 4 分, 共计 20 分)

1. 下面函数 `fun` 的功能是_____。

```
long fun(long s)
{
    long t = 0, sl = 1;
    int d;
    while (s > 0) {
        d = s % 10;
```

```

        if (d % 2 == 0) {
            t = d * sl + t;
            sl *= 10;
        }
        s /= 10;
    }
    return (t);
}

```

2. 下面程序的功能是_____。

```

#include <stdio.h>
#include <string.h>
int main()
{
    char s[80], t[200];
    int i, slen;
    gets(s);
    slen = strlen(s);
    for (i = 0; i < slen; i++)
        t[i] = s[slen-i-1];
    for (i = 0; i < slen; i++)
        t[slen+i] = s[i];
    t[slen+i] = '\0';
    puts(t);
    return 0;
}

```

3. 考虑如下定义, 该函数将返回_____。

```

int func1(float array[], int big)
{
    int middle = 0, start;
    float item = array[middle];
    for(start = 1; start < big; start++)
        if(array[start] < item)
            item = array[middle = start];
    return middle;
}

```

4. 运行下面程序, 其输出是_____。

```
#include <stdio.h>
#define MCRA(m)      2 * m
#define MCRB(n,m)    2 * MCRA(n) + m
int main() {
    int i = 2, j = 3;
    printf("%d\n", MCRB(j, MCRA(i)));
    return 0;
}
```

5. 考虑以下函数定义。函数调用 `f("33123333435", '3')` 的输出结果是_____。

```
void f(char *str, char del) {
    int i, j, len;
    len = strlen(str);
    i = 0;
    while(i < len) {
        while(str[i] == del) i++;
        j = i + 1;
        while(str[i] != del && str[j] != '\0') j++;
        str[j] = '\0';
        printf("%s\t", &str[i]);
        i = j + 1;
    }
}
```

七. 编程题 (每小题 10 分, 共计 20 分)

1. 编写函数, 实现用直接插入排序法对指定数组的前 `count` 个元素从小到大排序。
2. 编写函数, 在给定的文本文件中, 统计某个单词出现的次数以及每次出现的行号。

西安电子科技大学

2014 年攻读硕士学位研究生入学考试试题

考试科目代码及名称 902 数据结构与 C 语言程序设计

考试时间 2014 年 1 月 5 日下午 (3 小时)

答题要求: 所有答案 (填空题按照标号写) 必须写在答题纸上, 写在试题上一律作废, 准考证号写在指定位置!

一. 单项选择题 (每小题 2 分, 共计 20 分)

1. 某线性表最常用的运算是在表尾插入元素, 在表头删除元素, 则采用_____最节省运算时间。

- A. 仅设置尾指针的单向循环链表
- B. 仅设置头指针的单向循环链表
- C. 仅设置尾指针的单向链表
- D. 仅设置头指针的单向链表

2. 若已知一个栈的入栈序列是 $1, 2, 3, \dots, n$, 其输出序列为 $p_1, p_2, p_3, \dots, p_n$, 若 p_n 是 n , 则 p_i 是_____。

- A. $n-i+1$
- B. $n-i$
- C. i
- D. 不确定

3. 循环队列存储在数组 $A[0..m]$ 中, 用 $front$ 和 $rear$ 分别表示队头和队尾, 则入队时的操作为_____。

- A. $rear = rear + 1$
- B. $rear = (rear - front) \bmod (m + 1)$
- C. $rear = (rear - front) \bmod m$
- D. $rear = (rear + 1) \bmod (m + 1)$

4. 模式串 "ababaabab" 的 next 数组为_____。

- A. (0, 1, 1, 2, 2, 4, 2, 2, 3)
- B. (0, 1, 1, 2, 2, 2, 3, 1, 2)
- C. (0, 1, 1, 2, 3, 4, 2, 3, 4)
- D. (0, 1, 2, 3, 2, 1, 3, 1, 1)

5. 若一棵二叉树, 高度为 h , 所有结点的度为 0 或为 2, 则此树最少有_____个结点。

- A. $h+1$
- B. $2h-1$
- C. $2h$
- D. $2h+1$

6. 在一棵高度为 k 的完全二叉树中, 至少有_____个结点。

- A. 2^k
- B. $2^k - 1$
- C. 2^{k-1}
- D. $2^{k-1} - 1$

7. 若 X 是二叉中序线索树中一个有左孩子的结点, 且 X 不为根, 则 X 的前驱为_____。

- A. X 的双亲
- B. X 的右子树中最左的结点
- C. X 的左子树中最右的结点
- D. X 的左子树中最右的叶结点

8. 用邻接表表示图进行广度优先遍历时, 通常是采用_____来实现算法的。

A. 栈 B. 队列 C. 树 D. 图

9. 具有 12 个关键字的有序表, 折半查找的平均查找长度是_____。

A. 5 B. 4 C. 3.1 D. 2.5

10. 以下属于稳定排序方法的是_____。

A. 希尔排序 B. 快速排序 C. 堆排序 D. 2-路归并排序

二. 简答题 (每题 5 分, 共计 20 分)

1. 将两个栈存入数组 $V[1..m]$ 应如何安排最好? 这时栈空、栈满的条件是什么?

2. 设有三对角矩阵 $(a_{ij})_{n \times n}$, 将其三条对角线上的元素逐行的存于数组 $B(1:3n-2)$ 中, 使得 $B[k]=a_{ij}$, 求:

(1) 用 i, j 表示 k 的下标变换公式;

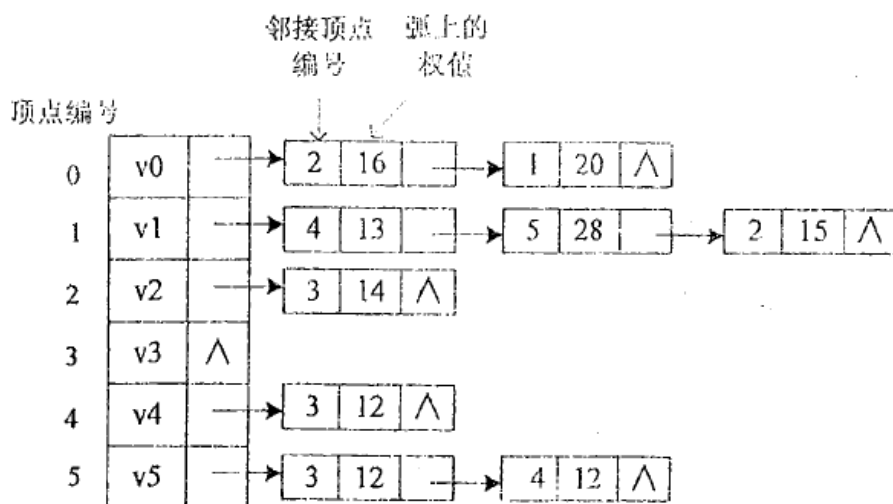
(2) 若 $n=10^3$, 每个元素占用 L 个单元, 则用 $B[k]$ 的方式比常规存储节省多少单元?

3. 求含有 n 个结点、采用顺序存储结构的完全二叉树中的序号最小的叶子结点的下标。要求写出简要步骤。

4. 假设序列由 n 个关键字不同的记录元素构成, 欲求前 k 个关键字值最大的元素, 用什么排序方法好? 为什么?

三. 综合题 (每题 10 分, 共计 40 分)

1. 某图的邻接链表如图所示。



(1) 分别写出该有向图所有可能的拓扑序列。

(2) 将该图看作无向图, 用普里姆 (Prim) 算法求其最小生成树 (写出每一步选中的边即可)。

2. 已知一棵二叉树的先序遍历序列为 ABDFCEGH, 中序遍历序列为 BFDAGEHC, 请构造此二叉树, 并画出与其对应的后序线索二叉树。

3. 已知关键字序列为 23, 31, 17, 27, 19, 11, 13, 91, 61, 41

(1) 请给出根据该序列构造的二叉排序树, 要求写出建树的过程;

(2) 请分别求出该二叉排序树查找成功和查找不成功时的平均查找长度。

(3) 设哈希表表长 $m=11$, 哈希函数 $\text{Hash}(\text{key}) = \text{key} \% 7$ 。请用线性探查法解决冲突构造哈希表, 并计算等概率查找成功的平均查找长度 (写出算式)。

4. 对于关键字序列 (28, 07, 39, 10, 65, 14, 61, 17, 50, 21):

(1) 以第一个元素为基准 (枢轴), 写出进行快速排序的第一趟划分的结果。

(2) 将该序列调整成小顶堆 (小根堆), 以二叉树形式写出序列的初始排列情况和形成小顶堆之后的排列情况。

四. 算法题 (每题 10 分, 20 分)

1. 设二叉树用二叉链表存储, 请写出链表的结点类型定义, 并编写算法判断某二叉树是否为二叉排序树。

2. 采用邻接表存储结构, 编写一个判别无向图中任意给定的两个顶点之间是否存在一条长度为 k 的简单路径的算法。

五. 程序填空题 (每空 2 分, 共计 10 分)

1. 下面函数 fun 的功能是将长整型数中的偶数依次取出, 构成一个新数返回, 例如, 当 s 中的数为: 87654321 时, 则返回的数为: 8642。请填空使程序完整、正确。

```
1    long fun(long s)
2    {    long t = 0, sl = 1;
3        int d;
4        while (s > 0) {
5            d = _____ (1) _____;
6            if (_____ (2) _____) {
7                t = _____ (3) _____;
8                sl *= 10;
9            }
10           s /= 10;
11       }
12       return (t);
13   }
```

2. 下面程序的功能是先将在字符串 s 中的字符按逆序存放到 t 串中, 然后把 s 中的字符按正序连接到 t 串的后面。例如, 当 s 中的字符串为: "ABCDE" 时则 t 中的字符串应为: "EDCBAABCDE"。请填空使程序完整、正确。

```

1    #include <stdio.h>
2    #include <string.h>
3    int main()
4    {    char s[80], t[200];
5         int i, slen;
6         gets(s);
7         slen = strlen(s);
8         for (i = 0; i < slen; i++)    t[i] = ____ (1) ____;
9         for (i = 0; i < slen; i++)    t[slen+i] = s[i];
10        ____ (2) ____;
11        puts(t);
12        return 0;
    }

```

六. 程序阅读题 (每小题 4 分, 共计 20 分)

1. 函数 strrchr 的功能是_____。

```

char *strrchr(char *s, char ch)
{
    char *p;
    p = strlen(s) + s;
    while (--p >= s)
        if (*p == ch) return p;
    return NULL;
}

```

2. 考虑如下定义, 该函数的作用是_____。

```

void func2(int n, char x[][12])
{
    char temp[12];
    int j, item;
    for(item = 0; item < n-1; ++item)
        for(j = item+1; j < n; ++j)
            if(strcmp(x[item], x[j]) > 0)
            {
                strcpy(temp, x[item]);
                strcpy(x[item], x[j]);
                strcpy(x[j], temp);
            }
}

```

3. 考虑如下定义，当输入 "*This is makeup test.*" 时，其输出是_____。

```
#include<stdio.h>
#include<string.h>
void main( )
{
    int i = 0, k = 0, n = 0;
    char str[80], *p = str;
    gets(p);
    for(; *p != '\0'; p++) {
        n++;
        if(*p == ' ')
            i = 0;
        else if(i == 0) {
            k++;
            i++;
        }
    }
    printf("cw=%d\ncc=%d\n", k, n);
}
```

4. 运行下面程序，其输出是_____。

```
#include <stdio.h>
void main( )
{
    static int a[ ]={2, 6, 10, 14, 18};
    static *ptr[ ]={&a[0], &a[1], &a[2], &a[3], &a[4]};
    int **p, i;
    for (i = 0; i < 5; i++)
        a[i] = a[i]/2 + a[i];
    p = ptr;
    printf ("%d\n", *(*p+2));
    printf ("%d\n", *(*++p));
}
```


5. 考虑以下函数定义。函数调用 `f("33123333435", '3')` 的输出结果是_____。

```
void f(char *str, char del)
```

```
{    int i, j, len;
```

```
    len = strlen(str);
```

```
    i = 0;
```

```
    while( i < len ){
```

```
        while( str[i] == del ) i++;
```

```
        j = i + 1;
```

```
        while(str[j] != del && str[j] != '\0') j++;
```

```
        str[j] = '\0';
```

```
        printf("%s\t", &str[i]);
```

```
        i = j+1;
```

```
    }
```

```
}
```

七. 编程题（每小题 10 分，共计 20 分）

1. 编写递归函数，判断输入的字符串，是否为回文(如：ABBA 就是回文)。
2. 编写函数，在给定的文本文件中，查找某个单词出现的行号及该行的内容。