广东工业大学

2017年攻读硕士学位研究生入学考试试题

考试科目（代码）名称：(831)数据结构与c语言　满分150分

（考生注意：答卷封面需填写自己的准考证编号，答完后连同本试题一并交回！）

一、单项选择题（共50分，25小题，每题2分）

1. 计算机算法指的是_____
A.计算方法　B. 排序方法　C.解决某一问题的有限运算序列　D.调度方法

2. 深度为6的二叉树最多有_____个结点
A.32　B.31　C.64　D.63

3.在具有 n 个结点的BST中查找数据元素时， 最坏情况下的时间复杂度是_____
A.O(n)　B. O(n^2)　C. O($\log_2 n$)　D. O($n\log_2 n$)

4.已知某二叉树的中序遍历序列是DEABC，前序遍历序列是BAEDC，则其后序遍历序列是_____，
A.DEACB　B. DEBAC　C. BAECD　D. BDAEC

5.采用顺序查找方法查找长度为n的线性表时，每个元素的平均查找长度为_____
A. n　B. n/2　C. (n+1)/2　D. (n-1)/2

6. 在所有的排序方法中，关键字比较的次数与记录的初始排列次序无关的是_____
A.希尔排序　B. 冒泡排序　C. 插入排序　D.选择排序

7. 在非空m阶B树上，除根节点以外的所有其它非终端节点_____
A.至少有m-1 棵子树　B.至多有m-1棵子树　C.至少有m棵子树　D.至多m棵子树

8.n个结点的线索二叉树中 线索数目是_____个
A.n　B.n+1　C.n+2　D.n-1

9. 对于含有 n 个顶点 e 条边的带权无向连通图 利用普里姆算法构造的最小生成树_____
A. 只有一棵　B. 有一棵或多棵　C.一定有多棵　D. 可能不存在

10 在平衡二叉树中，结点的平衡因子的绝对值不能超过_____
A.1　B. 0　C. 2　D. 3

11. 关于程序模块化，以下叙述错误的是（ ）。
A. 程序模块化可以提高程序编制的效率 B. 程序模块化可以提高代码复用率
C. 程序模块化可以提高程序运行的效率 D. 程序模块化可以提高调试程序的效率

12. 结构化程序的三种基本控制结构是（ ）。
A.顺序、选择和循环　B.顺序、选择和函数
C.层次、网状和关系　D.调用、返回和跳转

13. 下列选项中不属于C语言基本数据类型的是（ ）。
A. 单精度型　B.字符型　C.双长整型　D. 指针类型

14. 以下变量定义合法的是的是（ ）。
A. int 2n=1;　B. float define=1.234;
C. char _hello= '\0' ;　D. double x+1=2.5;

15.为了避免在嵌套的if-else语句中产生的二义性，C语言规定与else子句配对的是（ ）。
A. 与其在同一行上的if子句

B. 与其之后最近的不与其他 else 匹配的 if 子句
C. 与其缩排位置相同的 if 子句
D. 与其之前最近的不与其他 else 匹配的 if 子句

16.下列程序段循环执行的次数是（ ）。

```
int x=5; while (x=-10) printf("%d",x);
```

A. 无限次　B.有语法错，不能执行　C.一次也不执行　D. 执行 1 次

17. 对于 C 语言的 for 循环来讲，下列正确的说法为（ ）。

A.第一表达式不能放到循环前执行；　B.第二表达式不能放到循环前执行；
C.第三表达式能放到循环体前执行；　D.第三表达式不能放到循环体中执行。

18.以下关于数组的描述正确的是（　）。

A. 数组的大小是固定的，但可以有不同的类型的数组元素
B. 数组的大小是可变的，所有数组元素的类型必须相同
C. 数组的大小是固定的，所有数组元素的类型必须相同
D. 数组的大小是可变的，可以有不同的类型的数组元素

19.设有如下程序段：

```
int a[1]={0};int b[ ]={9};
char c[3]={"E", "F"};char d="12";
```

以下叙述正确的是（　）。

A. a,b 的定义合法，c，d 的定义不合法
B. a,b,c,d 的定义都合法
C. a,b,c 的定义合法，d 的定义不合法
D. 只有 a 的定义合法

20.要求定义一个有 6 个元素的整型一维数组，以下选项错误的是（ ）。

A. `int a[2*3]={0};`　B. `int N=6, a[N]`
C. `#define N 3`　D.`int a[]={1,2,3,4,5,6}`
`int a[N+N];`

21. C 语言中，以下叙述的是（　）。

A. 函数不能嵌套定义但可以嵌套调用　B. 函数可以嵌套定义也可以嵌套调用
C. 函数可以嵌套定义但不能嵌套调用　D. 函数不能嵌套定义也不能嵌套调用

22. 下列 C 语言的函数声明语句中，正确的是（　）。

A. `char fun(int,int);`　B. `char fun(int p,q);`
C. `char fun(p,q);`　D. `char fun(int p;int q);`

23. 下列哪种数据不存放在动态存储区中（ ）。

A. 函数形参变量　B. 局部自动变量
C. 函数调用时的现场保护和返回地址　D. 局部静态变量

24. 有以下程序段：

```
int *p1,*p2,a[10]; p1=a;p2=&a[5];
```

则 p2-p1 的值为（ ）。

A. 5　B.10　C.12　D.无法确定

25.在说明一个结构体变量时系统分配给它的存储空间是（ ）。

A. 该结构体中第一个成员所需的存储空间
B. 该结构体中最后一个成员所需的存储空间
C. 该结构体中占用最大存储空间的成员所需的存储空间
D. 该结构体中所有成员所需存储空间的总和。

二.C程序分析题（共26分，3小题，每题8分或10分）

1．阅读以下程序回答问题（8分）

```
void f21(int n)
{ int i;
  if((i=n/8) != 0)
     f21(i);
  putchar(n%8+'0');
}

void main()
{ int number;
  printf("Please input:");
  number = - 66;
  if(number < 0)
  {   putchar('-');
      number = -number;
  }
  f21(number);
}
```

(1)函数f21的功能是什么？(4分)

(2)程序的输出结果是什么？(4分)

2.阅读以下程序回答问题。（8分）

```
#define M   7
void f22 (int a[], int x, int n)
 {int  i,k;
  for ( i=M-1; i>=n; i--)
     a[i]=a[i-1];
     a[n]=x  ;
 }
main()
 {int a[M]={21,22,23,24,28,29}, x, n, k;
  n=4;
  x=25;
  f22(a, x, n   );
  for (k=4; k<M-2; k++)  printf("%d  ", a[k]);
  printf("\n");
 }
```

(1)函数f22的功能是什么？(4分)

(2)程序的输出结果是什么？(4分)

3.阅读以下程序回答问题。（10分）

```
#define N 6
char s[N];int w[6];
f31(char s[])
{ int k,i;
   for(k='A',i=0;i<N;i++)
```

```
    {s[i]=k;
     k+=1;
    }
}
f32(char *s,int *w)
{int sum=0;
 int k,i;
 for(k=0,i=0;i<N;i++)
    { sum=s[i]-'A'+10;
      w[i]=sum;
     }
}
main()
{int i;
 f31(s);
 f32(s,w);
 printf("%c=%d ",s[4],w[4]);
}
```

(1)函数 f31 的功能是什么？(3 分)
(2)函数 f32 的功能是什么？ (3 分)
(3)程序的输出结果是什么？(4 分)

三.C 程序填空题（共 18 分，2 小题，每题 9 分，每空 3 分）

1. 求一个 6×6 的整型矩阵对角线元素之和。

```
#include <stdio.h>
int main()
{ int  ____ (1) ___,sum=0;
  int i,j;
  printf("enter data:\n");
  for (i=0;i<6;i++)
    for (j=0;j<6;j++)
      scanf("%3d", ____ (2) ___);
  for (i=0;i<6;i++)
      ____ (3) ___;
  printf("sum=%6d\n",sum);
  return 0;
}
```

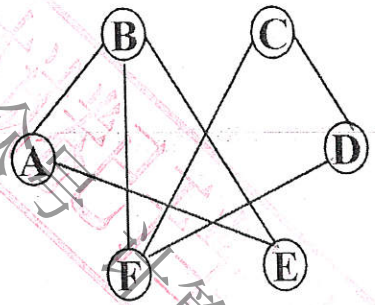
2. 将一个数组中的数值按逆序重新存放。例如，原来顺序为 9,7,4,2,1。要求改为 1,2,4,7,9。

```
#include <stdio.h>
#define N 5
 int main()
 { int a[N],i,temp;
   printf("enter array a:\n");
   for (i=0;i<N;i++)
     scanf("%d",&a[i]);
   printf("array a:\n");
```

```
    for (i=0;i<N;i++)
      printf("%4d",a[i]);
    for (i=0; _____(1)___;i++)
      { temp=a[i];
          _____(2)___;
        a[N-i-1]=temp;
      }
    printf("\nNow,array a:\n");
    for (i=0;i<N;i++)
      printf("%4d",_____(3)___);
    printf("\n");
    return 0;
}
```

四、数据结构解答题（共 20 分，3 小题，每题 6 分或 7 分）

1. 写出下图的邻接矩阵并画出该图的邻接表的存储结构图。（7 分）



2. 写出关键字序列{49 ，38，65，97， 76 ，13， 27， 56}快速排序的第一趟排序结果（6 分）

3. 设有一组关键字{19, 01, 23, 14, 55, 68, 11, 82, 36}，用哈希函数 H(key)=key%11，采用线性探测再散列的方法处理冲突，试在 0-14 的散列地址空间中对该序列构造哈希表，写出该哈希表，并求查找成功时的平均查找长度。（7 分）

五、算法填空题（共 18 分，3 小题，每题 6 分）

1. 顺序栈的类型定义为：

```
typedef struct {
  ElemType *elem;   // 存储空间的基址
  int top;          // 栈顶元素的下一个位置，简称栈顶位标
  int size;         // 当前分配的存储容量
  int increment;    // 扩容时，增加的存储容量
} SqStack;          // 顺序栈
```

算法 Pop_Sq(SqStack &S, ElemType &e) 实现顺序栈的出栈操作，请在空缺处填写合适的内容，使其成为完整的算法。

```
Status Pop_Sq(SqStack &S, ElemType &e)
/* 顺序栈 S 的栈顶元素出栈到 e，并返回 OK；*/
/* 若失败，则返回 ERROR。                  */
{
  if(S.top==0) return ___（1）____;
  ____（2）_____= S.elem[--S.top];
  return ____（3）_____;
}
```

2．顺序表类型定义如下：

```
typedef struct {
   ElemType *elem;
   int          length;
   int          size;
   int          increment;
} SqList;
```

算法 Inverse (SqList &L)实现顺序表的就地逆置，
即利用原表的存储空间将线性表(a1,a2,…,an)
逆置为(an,an-1,…,a1)。请在空缺处填写合适的内容，使其成为完整的算法。

```
void Inverse (SqList &L) {
   int i,n,x;
   ElemType ch;
   n=L.length-1;
   x = (n+1)/2;
   for (i=__（1）_____; i<=x-1; i++) {
      ch=____（2）_____;
      L.elem[i]=L.elem[n-i];
      L.elem[n-i]=___（3）_____;
   }
}
```

3．二叉链表类型定义：

```
typedef struct BiTNode {
   TElemType   data;
   struct BiTNode   *lchild, *rchild;
} BiTNode, *BiTree;
```

算法 binode (BiTree T)实现计算二叉树 T 中度为 2 的结点的数目。请在空缺处填写合适的内容，使其成为完整的算法。

```
int   binode (BiTree T)
/* 计算二叉树 T 中度为 2 的结点的数目 */
```

```
{
  if (NULL==T) return ____（1）_____;
  if (_____（2）_______) return 1;
  return binode (T->lchild) + binode  __（3）_____ ;
}
```

六、算法分析题（共 18 分，3 小题，每题 6 分）

1.二叉排序树的类型 BSTree 定义如下：

```
typedef struct {
    KeyType key;
    ... ...    // 其他数据域
} TElemType;
typedef struct BSTNode {
  TElemType   data;
  struct BSTNode   *lchild,*rchild;
}BSTNode, *BSTree;
```

算法 f1 定义如下：

```
void    f1(BSTree T, KeyType k, void(*visit)(TElemType))
/*  调用 visit(T->data)输出   */
{
  if (T!=NULL) {
    f1 (T->rchild, k, visit);
    if (T->data.key>=k) {
      visit(T->data);
      f1 (T->lchild, k, visit);
    }
  }
}
```

简述算法 f1 的功能。

2.单链表类型定义如下：

```
typedef struct LNode {
  ElemType        data;
  struct LNode *next;
} LNode, *LinkList;

Status f2(LinkList L, ElemType x)
{ LinkList   p,q;
  int k=0;
  p = L;   q = L->next;
  while (q!=NULL)
      if (q->data<x) {
      p->next = q->next;
      free(q);
```

```
            k++;
            q = p->next;
        } else {
            p = q;
            q = q->next;
        }
    return k;
}
```

简述算法 f2 的功能，若单链表 L 数据域的序列是（13, 45, 34, 10, 65, 7, 45, 15），请写出调用函数 f2(L，15)后的返回结果。

3.图的邻接表存储结构的类型定义如下：

```
#define UNVISITED   0
#define VISITED     1
#define INFINITY MAXINT // 计算机允许的整数最大值，即∞
typedef char VexType;
typedef enum {DG,DN,UDG,UDN} GraphKind; // 有向图,有向网,无向图,无向网
typedef struct AdjVexNode {
    int adjvex;   // 邻接顶点在顶点数组中的位序
    struct AdjVexNode *next; // 指向下一个邻接顶点（下一条边或弧）
    int info;     // 存储边（弧）相关信息，对于非带权图可不用
} AdjVexNode, *AdjVexNodeP; // 邻接链表的结点类型
typedef struct VexNode {
    VexType data;     // 顶点值，VexType 是顶点类型，由用户定义
    struct AdjVexNode *firstArc; // 邻接链表的头指针
} VexNode; // 顶点数组的元素类型
typedef struct {
    VexNode *vexs;   // 顶点数组，用于存储顶点信息
    int n, e;        // 顶点数和边（弧）数
    GraphKind kind; // 图的类型
    int *tags;       // 标志数组
} ALGraph;   // 邻接表类型

int f3(ALGraph G, int k)
{   int odeg;   odeg = 0;
    AdjVexNodeP p;
    if(k<0 || k>=G.n) return -1;
    p = G.vexs[k].firstArc;
    while(p!=NULL) {
        p = p->next;   odeg++;
    }
    return odeg;
}
```

简述算法 f3 的功能，并分析该算法的时间复杂度。