北京理工大学珠海学院

2013～2014 学年第二学期《数据结构》期中试卷

诚信声明	专业:
考场是严肃的，作弊是可耻的，对作弊人的处分是严厉的。我承诺遵守考场纪律，不存在抄袭及其它违纪行为。 考生（承诺人）签字：	班级:
	学号:

适用年级专业：13 级软件工程　**试卷说明**：闭卷，考试时间 90 分钟

题号	一	二	三	四	总分
得分					

一、单项选择题（每小题2分，共30分）【得分：　　】

1. (1)用两个栈可以模拟一个队列，(2)用两个队列可以模拟一个栈。（　　）

A. (1)对，(2)对　　B. (1)对，(2)错

C. (1)错，(2)对　　D. (1)错，(2)错

2. 下面两段程序的时间复杂性是（　　）。

```
(1) for(i=2;i<=n;++i)
        for(j=2;j<=i-1;++j)
        {    ++x;
             a[i][j]=x;
        }
(2) i=1;
    while(i<=n)
        i=i*2;
```

A. $O(n^2),O(\log_2 n)$　　B. $O(n^2),O(n^2)$

C. $O(n),O(n)$　　D. $O(n),O(\log_2 n)$

3、带头结点的循环单链表为空的判定条件是（　　）。

A. head==NULL　　B. head->next==NULL

C. head->next==head　　D. head!=NULL

4. 在一个具有n个单元的顺序栈中，假设栈底是存储地址的低端，现以top作为栈顶指针，当进行取栈顶元素操作时，top的变化是(　　)。

A. top=top+1　　B. top=top-1　　C. top不变　　D. top不确定

5. 一个栈的入栈序列为a，b，c，d，e则出栈序列可能的是(　　)。

A. a，b，c，d，e　　B、e，a，b，c，d

C. d，e，a，b，c　　D、a，b，e，c，d

6. 三维数组A[4][5][6]以行优先顺序存储，设基地址为100，每个元素的长度为3个字节，元素A[3][3][3]的存储地址是（　　）。

A. 433　　B. 225　　C. 325　　D. 333

7. 已知输入序列为abcd，经过输出受限的双端队列后能得到的输出序列是（　　）。

A. cadb　　B. dacb　　C. dcab　　D. 以上都不对

8. 链表不具有的特点是（　　）。

A. 可随机访问任一元素　　B. 插入删除不需要移动元素

C. 不必事先估计存储空间　　D. 所需空间与线性表长度成正比

9. 在一个单链表中，已知q是p的前趋结点，若q和p之间插入结点s，则执行（　　）。

A．s->next=p->next; p->next=s;　　B．p->next=s->next; s->next=p;

C．q->next=s; s->next=p;　　D．p->next=s; s->next=q;

10. 若某线性表中最常用的操作是取第i个元素和找第i个元素的前趋元素，则采用（　　）存储方式最节省时间。

A. 顺序表　　B. 单链表　　C. 双向链表　　D. 循环链表

11. 在数据结构中，从逻辑上可以把数据结构分成（　　）。

A. 动态和静态结构　　B. 紧凑接和非紧凑结构

C. 线性与非线性结构　　D. 内部结构和外部结构

12. 循环链表主要优点是（　　）。

A. 不再需要头指针了

B. 已知某个结点的位置后，能够容易找到它的直接前趋

C. 在进行插入、删除运算时，能更好地保证链表不断开

D. 从表中任一结点出发都能扫描到整个链表

13. 删除双链表中间某个节点时，需要修改（　　）个指针域。

A. 1　　B. 2　　C. 3　　D. 4

14. 中缀表达式a*(b+c)-d的后缀表达式是（　　）。

A. abcd*+-　　B. abc*+d-　　C. abc+*d-　　D. -+*abcd

15. 利用栈求表达式的值时，设立操作数栈OPND，假设OPND只有两个存储单元，在下列表达式中，不发生溢出的是（　　）。

A. A-B*(C-D)　　B. (A-B)*C-D　　C. (A-B*C)-D　　D. (A-B)*(C-D)

二、填空题（每空 2 分，共 20 分 ）【得分：　　】

1. 若一个算法中的语句频度之和为 T（n）=3n+n*$\log_2 n$，则算法的时间复杂度为 _____。

2. 假设为循环队列分配的向量空间为 Q[20]（下标从 0 开始），若队列的长度和队头指针值分别为 13 和 17，则当前队尾指针的值为_____。

3. 表长为 N 的顺序表，当在任何位置上插入或删除一个元素的概率相等时，插入一个元素所需移动元素的平均次数为_____;删除一个元素需要移动的元素个数为_____。

4. 将一个下三角矩阵 A[1..50,1..50]按行优先存入一维数组 B[1..n]中，A 中元素 A[30,20]在 B 数组中的位置为_____。

5. 模式串〃ababaababa〃采用 KMP 算法的 next 数组为_____;修正 nextval 数组为_____。

6. 设 S=〃I am a Student〃，T=〃good〃，
则 ConCat(T,SubStr(S,7,8)))=_____。

7. 下列稀疏矩阵对应的三元组为_____、_____。

$$\begin{bmatrix} 0 & 0 & 0 & 3 \\ 0 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

三、简答题（每题 6 分，共 12 分 ）【得分：　　】

1. 阅读下面算法，并回答下列问题：

（1）设队列 Q=（1，3，5，4，2），写出执行算法 fun 后的队列 Q;

（2）简述算法的功能。

```
void fun(Queue Q){
  ElemType e;
  if(!QueueEmpty(Q)){
     e=Dequeue(Q);
     fun(Q);
     Enqueue(Q,e);
  }
}
```

2. 对于堆栈，给出三个输入项 A，B，C，如果输入项序列为 ABC，试给出全部可能的输出序列，并写出每种序列对应的操作。例如：A 进 B 进 C 进 C 出 B 出 A 出，产生的序列为 CBA。

四、算法阅读题（第 1-5 题每题 6 分，第 6 题 8 分，共 38 分）

1. 算法 fun1 实现在顺序表第 i 个位置插入元素 e(整型)，请在_____处补充适当的语句（可以写多条语句）完成算法。

```
#define LINC    10    //线性表存储空间的分配增量
typedef struct{
   int      *elem;          //存储空间基址
   int      length;         //当前长度
   int      listsize;       //当前分配的存储容量
}SqList;

Status fun1(SqList &L,int i, int e){// 顺序表的插入
  if(i<1||i>L.length+1)  return ERROR;   //i 值不合法
  if(L.length>=L.listsize){     //当前存储空间已满,增加分配
    newb=(int*)realloc(L.elem,(L.listsize+LINC)*sizeof(int));
    if(!newb) exit(OVERFLOW);       //存储分配失败
    L.elem=newb;                    //新基址
    L.listsize+=LINC;               //增加存储容量
   }
```

_____

```
  ++L.length;                                  //表长增 1
```

```
  return OK;
} //fun1
```

2. 算法 fun2 实现在带头结点的单链表第 i 个位置的插入元素 e(整型)，请在_____处补充适当的语句（可以写多条语句）完成算法。

```
typedef struct LNode{
    int            data;       //数据域
    struct LNode  *next;       //指针域
}LNode, *LinkList;

Status fun2(LinkList &L,int i,int e){
  p=L;  j=0;
  while(p && j<i-1){p=p->next;++j;}     //寻找第 i-1 个结点
  if(!p||j>i-1) return ERROR;       //i 小于 1 或大于(表长+1)
  s=(LinkList)malloc(sizeof(LNode));    //生成新结点
```

_____

```
  return OK;
}//fun2
```

3. 算法 fun3 实现：逆位序输入 n 个元素的值，建立带头结点的单链线性表 L，请在_____处补充适当的语句（可以写多条语句）完成算法。

```
void fun3(LinkList &L,int n){
  L=(LinkList)malloc(sizeof(LNode));
  L->next=NULL;                  //先建立一个带头结点的空单链表
  for(i=n;i>0;--i){
    p=(LinkList) malloc(sizeof(LNode));  //生成新结点
    scanf(&p->data);                     //输入元素值
```

_____

```
  }
}//fun3
```

4. 算法fun4实现在循环队列的出队列操作，出队元素赋给e，请在_____处补充适当的语句完成算法。

```
#define MAXQSIZE  100   //最大队列长度
typedef struct{
   QElemType     *base;   //初始化的动态分配存储空间
   int  front;  //头指针,若队列不空,指向队列头元素.
   int  rear;   //尾指针,若队列不空,指向队列尾元素的下一个位置.
}SqQueue;

Status fun4(LinkQueue &Q,QElemType &e){
   if(_____) return ERROR;

    _____

    _____
    return OK;
}//fun4
```

5. 算法fun5的功能是，对以带头结点的单链表作为存储结构的两个递增有序表La和Lb（表中不存在值相同的数据元素）进行如下操作：依次检查Lb表中的元素，如果已在La中出现则删除之，否则将元素结点插入到La中。请在空缺处填入合适的内容，使其成为完整的算法。

```
void fun5(LinkList La,LinkList Lb){
  LinkList pre=La,q;
  LinkList pa=La->next,pb=Lb->next;
  free(Lb);
  while(pa && pb){
     if(pa->data<pb->data)
        {pre=pa;pa=pa->next;}
     else if(pa->data>pb->data){
        _____;
        pre=pb; pb=pb->next;
        _____;
     }
     else { q=pb; pb=pb->next; free(q); }
  }
  if(pb)
     _____;
}//fun5
```

6. 已知栈的基本操作函数：

```
int InitStack(SqStack &S); //构造空栈
int StackEmpty(SqStack S);//判断栈空
int Push(SqStack &S,ElemType e);//入栈
int Pop(SqStack &S,ElemType &e);//出栈
int GetTop(SqStack S, ElemType &e);//取栈顶元素
```

算法fun6实现检验exp[]括号是否匹配，利用上面的基本操作来实现，并在_____处补充适当的语句完成算法。

注：假设在表达式中([ ]( )) 或[([ ][ ])]等为正确的格式，

[( ]) 或([( )) 或 (( )])均为不正确的格式。

```
Status fun6(char exp[]){
  int state=1,i=0,ii=0;
  while(exp[i]!='\0'&& state){
    switch of exp[i]{
        case '(' : _____
        case '[' : _____
        case ')' :
                   _____
        case ']' :
                   _____
    }//switch
    i++;
  }
  if (StackEmpty(S)&&state)
     return OK;
  else
     return ERROR;
}//fun6
```