

## 习 题 三 栈和队列

### 3.1 单项选择题

1. 一个栈的入栈序列 a, b, c, d, e, 则栈的不可能的输出序列是 **C**。

- A. edcba    B. decba    C. dceab    D. abcde

2. 若已知一个栈的入栈序列是 1, 2, 3, ..., n, 其输出序列为 p1, p2, p3, ..., pn, 若 p1=n, 则 pi 为 **C**。

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

3. 栈结构通常采用的两种存储结构是 **A**。

- A. 顺序存储结构和链式存储结构  
B. 散列方式和索引方式  
C. 链表存储结构和数组  
D. 线性存储结构和非线性存储结构

4. 判定一个栈 ST (最多元素为 m0) 为空的条件是 **B**。

- A. ST  $\rightarrow$  top  $\neq$  0    B. ST  $\rightarrow$  top = 0  
C. ST  $\rightarrow$  top  $\neq$  m0    D. ST  $\rightarrow$  top = m0

5. 判定一个栈 ST (最多元素为 m0) 为栈满的条件是 **D**。

- A. ST  $\rightarrow$  top  $\neq$  0    B. ST  $\rightarrow$  top = 0  
C. ST  $\rightarrow$  top = m0    D. ST  $\rightarrow$  top = m0

6. 栈的特点是 **B**, 队列的特点是 **A**。

- A. 先进先出    B. 先进后出

7. 一个队列的入列序列是 1, 2, 3, 4, 则队列的输出序列是 **B**。

- A. 4, 3, 2, 1    B. 1, 2, 3, 4  
C. 1, 4, 3, 2    D. 3, 2, 4, 1

8. 判定一个循环队列 QU (最多元素为  $m0$ ,  $m0+1 = \text{Maxsize}$ ) 为满队列的条件是 **A**。

- A.  $((\text{QU} \rightarrow \text{rear} - \text{QU} \rightarrow \text{front}) + \text{Maxsize}) \% \text{Maxsize} = m0$
- B.  $\text{QU} \rightarrow \text{rear} - \text{QU} \rightarrow \text{front} - 1 = m0$
- C.  $\text{QU} \rightarrow \text{front} = \text{QU} \rightarrow \text{rear}$
- D.  $\text{QU} \rightarrow \text{front} = \text{QU} \rightarrow \text{rear} + 1$

9. 判定一个循环队列 QU 为空的条件是 **A**。

- A.  $\text{QU} \rightarrow \text{front} = \text{QU} \rightarrow \text{rear}$
- B.  $\text{QU} \rightarrow \text{front} \neq \text{QU} \rightarrow \text{rear}$
- C.  $\text{QU} \rightarrow \text{front} = (\text{QU} \rightarrow \text{rear} + 1) \% \text{maxsize}$
- D.  $\text{QU} \rightarrow \text{front} \neq (\text{QU} \rightarrow \text{rear} + 1) \% \text{maxsize}$

10. 判定一个循环队列 QU 为满队列的条件是 **C**。

- A.  $\text{QU} \rightarrow \text{front} = \text{QU} \rightarrow \text{rear}$
- B.  $\text{QU} \rightarrow \text{front} \neq \text{QU} \rightarrow \text{rear}$
- C.  $\text{QU} \rightarrow \text{front} = (\text{QU} \rightarrow \text{rear} + 1) \% \text{maxsize}$
- D.  $\text{QU} \rightarrow \text{front} \neq (\text{QU} \rightarrow \text{rear} + 1) \% \text{maxsize}$

11. 循环队列用数组  $A[0, m-1]$  存放其元素值, 已知其头尾指针分别是 front 和 rear, 则当前队列中的元素个数是 **A**。

- A.  $(\text{rear} - \text{front} + m) \% m$
- B.  $\text{rear} - \text{front} + 1$
- C.  $\text{rear} - \text{front} - 1$
- D.  $\text{rear} - \text{front}$

12. 循环队列存储在数组  $A[0..m-1]$  中, 则入队时 rear 应该变化为 (**A**)。

- A.  $\text{rear} = (\text{rear} + 1) \% m$ ;
- B.  $\text{rear}++$ ;
- C.  $\text{rear} = (\text{rear} + 1) \% (m + 1)$ ;
- D.  $\text{rear} = (\text{rear} + 1) \% (m - 1)$ ;

13. 栈和队列的共同点是 **C**。

- A. 都是先进后出
- B. 都是先进先出
- C. 只允许在端点处插入和删除元素
- D. 没有共同点

14、经过下列运算后，队头的元素是（**B**）。

InitQueue(qu); Enqueue(qu, 'a'); EnQueue(qu, 'b'); EnQueue(qu, 'c'); DeQueue(qu);

- A、 a
- B、 b
- C、 1
- D、 0

15、经过以下运算后， x 的值是（**A**）。

InitStack (s); Push(s, a); Push(s, b); Pop(s, x); GetTop(s,x)

- A、 a
- B、 b
- C、 1
- D、 0

16. 向一个栈顶指针为 HS 的链栈中插入一个 s 所指结点时，则执行 **C**。

(不带空的头结点)

- A. HS—>next=s;
- B. s—>next= HS—>next; HS—>next=s;
- C. s—>next= HS; HS=s;
- D. s—>next= HS; HS= HS—>next;

17. 从一个栈顶指针为 HS 的链栈中删除一个结点时，用 x 保存被删结点的值，则执行 **D**。

(不带空的头结点)

- A. x=HS; HS= HS—>next;
- B. x=HS—>data;
- C. HS= HS—>next; x=HS—>data;
- D. x=HS—>data; HS= HS—>next;

### 3.2 填空题（将正确的答案填在相应的空中）

1、向量、栈和队列都是\_\_**线性**\_\_结构，可以在向量的\_\_**任意**\_\_位置插入和删除元素；

对于栈只能在\_\_**栈顶**\_\_插入和删除元素；对于队列只能在\_\_**队尾**\_\_插入元素和\_\_**队头**\_\_删除元素。

2、向顺序栈中压入元素的操作是先插入再指针上移。

3、对顺序栈进行退栈时的操作是先下移指针，再删除。

4、在一个循环队列中，队首指针指向队首元素的当前位置。

5、从循环队列中删除一个元素时，其操作是先删除，再 front 指针循环队列意义下加 1。

6、在具有  $n$  个单元的循环队列中，队满时共有 $n-1$ 个元素。

7、一个栈的输入序列是 12345，则栈的输出序列 43512 是不可能。

8、一个栈的输入序列是 12345，则栈的输出序列 12345 是可能。

### 3.3 算法设计题:

1. 已知线性表中的元素以值递增有序排列，并以单链表作存储结构。试写一算法，删除表中所有大于  $x$  且小于  $y$  的元素（若表中存在这样的元素）同时释放被删除结点空间。
2. 试写一算法，实现单链表的就地逆置。

## 习 题 四 串

### 4.1 单项选择题

1. 空串与空格串是相同的, 这种说法 ( **B** )。

- A. 正确                      B. 不正确

2. 串是一中特殊的线性表, 其特殊性体现在 (        )。//已删

- A. 可以顺序存储                      B. 数据元素是一个字符  
C. 可以链接存储                      D. 数据元素可以是多个字符

### 数据元素只能取字符集

3. 设有两个串 p 和 q, 求 q 在 p 中首次出现的位置的运算称作 ( **B** )。

- A. 连接                      B. 模式匹配  
C. 求子串                      D. 求串长

4. 设串 s1='ABCDEFGH', s2='PQRST', 函数 concat(R,x,y)返回 x 和 y 串的连接串, subs(t, s,i,j)返回串 s 的从序号 i 的字符开始的 j 个字符组成的子串, len(s)返回串 s 的长度,

则 concat (subs (&s3, s1,2,len (s2)), subs (&s4, s1,len (s2),2))的结果串是 ( **D** )。

- A. BCDEF                      B. BCDEFG  
C. BCPQRST                      D. BCDEFEF

5. 函数 strcmp('stcabuc','stbabuc')的返回值是 ( **B** ) 。

- A、 2  
B、 1  
C、 0  
D、 -1

6. 设串 S1='I AM', S2=' A STUDENT', 则 ConcatStr(S1,S2)=( **A** )。

- A、 'I AM A STUDENT'  
B、 'I AM'  
C、 'A STUDENT'  
D、 'IAMASTUDENT'

7、若串 S='software',其前缀真子串的数目是 (A)。

- A、 7
- B、 10
- C、 9
- D、 8

8、空串是任何串的子串，则串 S='Computer'的子串个数是 (A)。

- A、 37
- B、 9
- C、 36
- D、 8

9、StrIndex ('DATASTRUCTURE', 1, 'STR') = (A)。

- A、 5
- B、 3
- C、 7
- D、 9

10、S1='good', S2='morning', 执行函数 SubStr(S2,4,LenStr(S1))后的结果为(A)。

- A、 'ning'
- B、 'good'
- C、 'go'
- D、 'morn'

11、设 "acbcaca" 是模式串， 则在进行 KMP 算法进行匹配时， 它的 next 数组元素依次为 (A)。

- A、 0111123
- B、 0112122
- C、 0111221
- D、 0111132
- E、 0111113

#### 4.2 填空题（将正确的答案填在相应的空中）

1. 串的两最最基本的存储方式是顺序存储和链式存储。

2. 两个串相等的充分必要条件是\_\_**长度相等且对应字符相等**\_\_。
3. 空串是\_\_**长度为 0 的串**\_\_，其长度等于\_\_**0**\_\_。
4. 空格串是\_\_**空格为字符的串**\_\_，其长度等于\_\_**空格个数**\_\_。
5. 设  $s = \text{'I\_AM\_A\_TEACHER'}$ , 其长度是 **14**。

#### 4.3 算法设计题:

1. 编写算法，从串  $s$  中删除所有和串  $t$  相同的子串。
2. 编写算法，实现串的基本操作  $\text{Replace}(\&S, T, V)$ 。

## 习 题 五      数 组

### 5.1 单项选择题（其中 $A[i..j]$ 表示下标从 $i$ 到 $j$ ）

2. 二维数组  $M$  的成员是 6 个字符（每个字符占一个存储单元，即一个字节）组成的串，行下标  $i$  的范围从 0 到 8，列下标  $j$  的范围从 1 到 10，则存放  $M$  至少需要\_\_①\_D\_个字节；

$M$  的第 8 列和第 5 行共占②A\_个字节。  $9+10-1 \times 6$

① A. 90      B. 180      C. 240      D. 540

② A. 108      B. 114      C. 54      D. 60

4. 数组  $A$  中，每个元素  $A$  的长度为 3 个字节，行下标  $i$  从 1 到 8，列下标  $j$  从 1 到 10，从首地址  $SA$  开始连续存放在存储器内，存放该数组至少需要的单元数是 C。

A. 80                  B. 100                  C. 240                  D. 270

5. 数组  $A$  中，每个元素  $A$  的长度为 3 个字节，行下标  $i$  从 1 到 8，列下标  $j$  从 1 到 10，从首地址  $SA$  开始连续存放在存储器内，该数组按行存放时，元素  $A[8][5]$  的起始地址为 C。

A.  $SA+141$       B.  $SA+144$       C.  $SA+222$       D.  $SA+225$

6. 数组  $A$  中，每个元素  $A$  的长度为 3 个字节，行下标  $i$  从 1 到 8，列下标  $j$  从 1 到 10，从首地址  $SA$  开始连续存放在存储器内，该数组按列存放时，元素  $A[5][8]$  的起始地址为 B。

A.  $SA+141$       B.  $SA+180$       C.  $SA+222$       D.  $SA+225$

7. 以行序优先顺序存储数组  $A[5][5]$ ；假定  $A[0][0]$  的地址为 1000，每个元素占 4 个字节，下标变量  $A[4][3]$  的地址是 A。

A、 1092

B、 1023

C、 1046

D、 1069

8. 数组  $A[1..6][1..5]$  以列序优先顺序存储，第一个元素  $A[1][1]$  的地址为 1000，每个元素占 2 个存储单元，则  $A[3][4]$  的地址是 A。

A、 1040



- B、 1026
- C、 1038
- D、 1046

9、数组  $A[0..5,0..6]$  的每个元素占 5 个字节，将其按列序为主序存储在起始地址为 1000 的内存单元中，则元素  $A[5][5]$  的地址是( A )。

- A、 1175
- B、 1180
- C、 1205
- D、 1210

10、对广义表  $E = ((a, ((b), c)), e, E)$  实施由取表头 `GetHead` 和取表尾 `GetTail` 组成的操作序列 ( A ) 得到的结果是  $c$ 。

- A、 `GetHead(GetTail(GetHead(GetTail(GetHead(E)))))`
- B、 `GetHead(GetTail(GetTail(E)))`
- C、 `GetHead(GetHead(GetTail(GetHead(GetHead(GetTail(GetHead(GetHead(E)))))))`
- D、 `GetHead(GetTail(GetHead(E)))`
- E、 `GetHead(GetTail(GetHead(GetTail(E))))`

11、若广义表满足  $\text{GetHead}(A) = \text{GetTail}(A)$ ，则  $A$  为 ( A )。

- A、  $(( ))$
- B、  $( )$
- C、  $(( ), ( ))$
- D、  $(( ), ( ), ( ))$
- E、  $(( ( )) )$

12、设广义表  $A = (a, (((b,c),(d,e)),f),g)$ ， $\text{Head}(p)$  为求广义表  $p$  的表头的函数， $\text{Tail}(p)$  为求广义表  $p$  的表尾的函数，那么从表  $A$  中取出子表  $(b,c)$  的操作为 ( A )。

- A、 `Head(Head(Head(Tail(A))))`
- B、 `Tail(Head(Head(Head(A))))`
- C、 `Head(Head(Head(Head(Tail(A)))))`
- D、 `Tail(Head(Head(Head(Head(A)))))`
- E、 `Head(Head(Head(Head(Head(Head(Tail(A)))))))`
- F、 `Tail(Head(Head(Head(Head(Head(Head(A)))))))`

13、广义表 $((a,b),(c),(d))$ 的表尾是 **A**。

- A、 $((c),(d))$
- B、 $(d)$
- C、 $d$
- D、 $(a,b)$

14、对广义表  $G=((a, (( ), b)), ((( ), (c,d)), ( )))$  执行  $\text{tail}(\text{head}(\text{head}(\text{tail}(G))))$  操作的结果是 **A**。

- A、 $((c,d))$
- B、 $( )$
- C、 $c$
- D、 $(c,d)$

### 5.2 填空题（将正确的答案填在相应的空中，其中 $A[i,j]$ 表示下标从 $i$ 到 $j$ ）

1. 已知二维数组  $A[m][n]$  采用行序为主方式存储，每个元素占  $k$  个存储单元，并且第一个元素的存储地址是  $\text{LOC}(A[0][0])$ ，则  $A[i][j]$  的地址是  **$\text{LOC}(A[0][0]) + (i*n+j)*k$** 。

2. 二维数组  $A[10][20]$  采用列序为主方式存储，每个元素占一个存储单元并且  $A[0][0]$  的存储地址是 200，则  $A[6][12]$  的地址是 **326**。

3. 二维数组  $A[10..20][5..10]$  采用行序为主方式存储，每个元素占 4 个存储单元，并且  $A[10][5]$  的存储地址是 1000，则  $A[18][9]$  的地址是 **1208**。

### 5.3 算法设计题:

1. 假设稀疏矩阵  $A$  和  $B$  均以三元组顺序表作为存储结构。试写出矩阵相加的算法，另设三元组表  $C$  存放结果矩阵。
2. 假设系数矩阵  $A$  和  $B$  均以三元组顺序表作为存储结构。试写出满足以下条件的矩阵相加的算法：假设三元组顺序表  $A$  的空间足够大，将矩阵  $B$  加到矩阵  $A$  上，不增加  $A$ 、 $B$  之外的附加空间，你的算法能否达到  $O(m+n)$  的时间复杂度？其中  $m$  和  $n$  分别为  $A$ 、 $B$  矩阵中非零元的数目。
3. 试编写一个以三元组形式输出用十字链表表示的稀疏矩阵中非零元素及其下标的算法。
4. 求下列广义表操作的结果：

(1)  $\text{GetTail}[\text{GetHead}[(a,b),(c,d)]]$ ;  **$((c,d))$**

(2)  $\text{GetTail}[\text{GetHead}[\text{GetTail}[(a, b), (c, d)]]]$  (d)

5. 利用广义表的  $\text{GetHead}$  和  $\text{GetTail}$  操作写出如上题的函数表达式, 把原子 banana 分别从下列广义表中分离出来.

(1)  $L_5 = (((\text{apple})), (\text{pear}), (\text{banana}), \text{orange});$  wwt

(2)  $L_7 = (\text{apple}, (\text{pear}, (\text{banana}), \text{orange}));$  twtt