



第三章栈和队列 自测卷答案

姓名_____ 班级_____

题号	一	二	三	四	五	六	总分
题分	15	10	20	20	20	15	100
得分							

一、填空题（每空 1 分，共 15 分）

1. 一个栈的输入序列是：1，2，3 则不可能的栈输出序列是 3,1,2。
2. 栈是一种特殊的线性表，允许插入和删除运算的一端称为 栈顶。不允许插入和删除运算的一端称为 栈底。
3. 在一个循环队列中，队首指针指向队首元素的 前一个 位置。
4. 【00 年统考题】带表头结点的空循环双向链表的长度等于 0。
5. 从循环队列中删除一个元素时，其操作是 先 移动队首指针，后 取出元素。
6. 队列 是被限定为只能在表的一端进行插入运算，在表的另一端进行删除运算的线性表。
7. 设有一个空栈，栈顶指针为 1000H(十六进制)，现有输入序列为 1，2，3，4，5，经过 PUSH,PUSH,POP,PUSH,POP,PUSH,PUSH 之后，输出序列是 2,3，而栈顶指针值是 100c H。(设栈为顺序栈，每个元素占 4 个字节)
8. 在具有 n 个单元的循环队列中，队满时共有 n-1 个元素。
9. 当两个栈共享一存储区时，栈利用一维数组 stack(1,n)表示，两栈顶指针为 top[1]与 top[2]，则当栈 1 空时，top[1]为 0，栈 2 空时，top[2]为 n+1，栈满时为 top【1】+1=top【2】

二、判断正误（判断下列概念的正确性，并作出简要的说明。）（每小题 1 分，共 10 分）

- (☒) 1. 栈与队列是一种特殊操作的线性表。
- (☒) 2. 栈和队列的存储方式，既可以是顺序方式，又可以是链式方式。
- (☒) 3. 栈是一种对所有插入、删除操作限于在表的一端进行的线性表，是一种后进先出型结构。
- (☒) 4. 栈和链表是两种不同的数据结构。
错，栈是逻辑结构的概念，是特殊线性表，而链表是存储结构概念，二者不是同类项。
- (☒) 5. 两个栈共享一片连续内存空间时，为提高内存利用率，减少溢出机会，应把两个栈的栈底分别设在这片内存空间的两端。
- (☒) 6. 栈和队列的存储方式既可是顺序方式，也可是链接方式。
- (☒) 7. 线性表的每个结点只能是一个简单类型，而链表的每个结点可以是一个复杂类型。
错，线性表是逻辑结构概念，可以顺序存储或链式存储，与元素数据类型无关。
- (☒) 8. 一个栈的输入序列是 12345，则栈的输出序列不可能是 12345。
错，有可能。
- (☒) 9. 队是一种插入与删除操作分别在表的两端进行的线性表，是一种先进后出型结构。
错，后半句不对。
- (☒) 10. 栈和队列是一种非线性数据结构。
错，他们都是线性逻辑结构，栈和队列其实是特殊的线性表，对运算的定义略有不同而已。

三、单项选择题（每小题 1 分，共 20 分）

(C) 1. 【李春葆】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

- A. i B. n-i C. n-i+1 D. 不确定

解释: 当 p1=n, 即 n 是最先出栈的, 根据栈的原理, n 必定是最后入栈的, 那么输入顺序必定是 1, 2, 3, ..., n, 则出栈的序列是 n, ..., 3, 2, 1。

(B) 2. 【李春葆】判定一个队列 QU (最多元素为 m0) 为满队列的条件是

- A. $QU \rightarrow rear - QU \rightarrow front == m0$
B. $QU \rightarrow rear - QU \rightarrow front - 1 == m0$
C. $QU \rightarrow front == QU \rightarrow rear$
D. $QU \rightarrow front == QU \rightarrow rear + 1$

(B) 3. 【00 年元月统考题】栈中元素的进出原则是

- A. 先进先出 B. 后进先出 C. 栈空则进 D. 栈满则出

(B) 4. 一个栈的输入序列为 123...n, 若输出序列的第一个元素是 n, 输出第 i (1<=i<=n) 个元素是

- A. 不确定 B. n-i+1 C. i D. n-i

(C) 5. 有六个元素 6, 5, 4, 3, 2, 1 的顺序进栈, 问下列哪一个不是合法的出栈序列?

- A. 5 4 3 6 1 2 B. 4 5 3 1 2 6 C. 3 4 6 5 2 1 D. 2 3 4 1 5 6

6. 【94 初程 P75】从供选择的答案中, 选出应填入下面叙述____?____内的最确切的解答, 把相应编号写在答卷的对应栏内。

栈是一种线性表, 它的特点是__A__。设用一维数组 A[1,...,n]来表示一个栈, A[n]为栈底, 用整型变量 T 指示当前栈顶位置, A[T]为栈顶元素。往栈中推入 (PUSH) 一个新元素时, 变量 T 的值__B__; 从栈中弹出 (POP) 一个元素时, 变量 T 的值__C__。设栈空时, 有输入序列 a, b, c, 经过 PUSH, POP, PUSH, PUSH, POP 操作后, 从栈中弹出的元素的序列是__D__, 变量 T 的值是__E__。

供选择的答案:

- A: ① 先进先出 ② 后进先出 ③ 进优于出 ④ 出优于进 ⑤ 随机进出
B, C: ① 加 1 ② 减 1 ③ 不变 ④ 清 0 ⑤ 加 2 ⑥ 减 2
D: ① a,b ② b,c ③ c,a ④ b,a ⑤ c,b ⑥ a,c
E: ① n+1 ② n+2 ③ n ④ n-1 ⑤ n-2

答案: ABCDE=2, 2, 1, 6, 4

注意, 向地址的高端生长, 称为向上生成堆栈; 向地址低端生长叫向下生成堆栈, 本题中底部为 n, 向地址的低端递减生成, 称为向下生成堆栈。

7. 【91 初程 P77】从供选择的答案中, 选出应填入下面叙述____?____内的最确切的解答, 把相应编号写在答卷的对应栏内。

在做进栈运算时, 应先判别栈是否__A__; 在做退栈运算时, 应先判别栈是否__B__。当栈中元素为 n 个, 做进栈运算时发生上溢, 则说明该栈的最大容量为__C__。为了增加内存空间的利用率和减少溢出的可能性, 由两个栈共享一片连续的内存空间时, 应将两栈的__D__分别设在这片内存空间的两端, 这样, 只有当__E__时, 才产生上溢。

供选择的答案：

A, B: ①空 ② 满 ③ 上溢 ④ 下溢

C: ①n-1 ② n ③ n+1 ④ n/2

D: ① 长度 ②深度 ③ 栈顶 ④ 栈底

E: ①两个栈的栈顶同时到达栈空间的中心点 ②其中一个栈的栈顶到达栈空间的中心点

③两个栈的栈顶在达栈空间的某一位置相遇 ④两个栈均不空,且一个栈的栈顶到达另一个栈的栈底

答案: ABCDE=2, 1, 2, 4, 3

8. 【98 初程 P71】 从供选择的答案中,选出应填入下面叙述____?____内的最确切的解答,把相应编号写在答卷的对应栏内。

设有 4 个数据元素 a1、a2、a3 和 a4,对他们分别进行栈操作或队操作。在进栈或进队操作时,按 a1、a2、a3、a4 次序每次进入一个元素。假设栈或队的初始状态都是空。

现要进行的栈操作是进栈两次,出栈一次,再进栈两次,出栈一次;这时,第一次出栈得到的元素是__A__,第二次出栈得到的元素是__B__是;类似地,考虑对这四个数据元素进行的队操作是进队两次,出队一次,再进队两次,出队一次;这时,第一次出队得到的元素是__C__,第二次出队得到的元素是__D__。经操作后,最后在栈中或队中的元素还有__E__个。

供选择的答案:

A~D: ①a1 ②a2 ③ a3 ④a4

E: ①1 ②2 ③ 3 ④ 0

答: ABCDE=2, 4, 1, 2, 2

四、简答题（每小题 5 分，共 20 分）

1. 简述顺序存储队列的假溢出的避免方法及队列满和空的条件。

答案: 设顺序存储队列用一维数组 $q[m]$ 表示,其中 m 为队列中元素个数,队列中元素在向量中的下标从 0 到 $m-1$ 。设队头指针为 $front$,队尾指针是 $rear$,约定 $front$ 指向队头元素的前一位置, $rear$ 指向队尾元素。当 $front$ 等于 -1 时队空, $rear$ 等于 $m-1$ 时为队满。由于队列的性质(“删除”在队头而“插入”在队尾),所以当队尾指针 $rear$ 等于 $m-1$ 时,若 $front$ 不等于 -1,则队列中仍有空闲单元,所以队列并不是真满。这时若再有入队操作,会造成假“溢出”。其解决办法有二,一是将队列元素向前“平移”(占用 0 至 $rear-front-1$);二是将队列看成首尾相连,即循环队列 (0.. $m-1$)。在循环队列下,仍定义 $front=rear$ 时为队空,而判断队满则用两种办法,一是用“牺牲一个单元”,即 $rear+1=front$ (准确记是 $(rear+1) \% m = front$, m 是队列容量)时为队满。另一种解法是“设标记”方法,如设标记 tag , tag 等于 0 情况下,若删除时导致 $front=rear$ 为队空; $tag=1$ 情况下,若因插入导致 $front=rear$ 则为队满。

2. 【统考书 P60 4-11, 难于严题集 3.1①】设有编号为 1, 2, 3, 4 的四辆列车,顺序进入一个栈式结构的车站,具体写出这四辆列车开出车站的所有可能的顺序。

答案: 至少有 14 种。

① 全进之后再出情况,只有 1 种: 4, 3, 2, 1

② 进 3 个之后再出的情况,有 3 种, 3,4,2,1 3,2,4,1 3,2,1,4

③ 进 2 个之后再出的情况,有 5 种, 2,4,3,1 2,3,4,1 2,1, 3,4 2,1,4,3 2,1,3,4

④ 进 1 个之后再出的情况，有 5 种，1,4,3,2 1,3,2,4 1,3,4,2 1, 2,3,4 1,2,4,3

3. 【统考书 P60 4-13】顺序队的“假溢出”是怎样产生的？如何知道循环队列是空还是满？

答案：一般的一维数组队列的尾指针已经到了数组的上界，不能再有入队操作，但其实数组中还有空位置，这就叫“假溢出”。

采用循环队列是解决假溢出的途径。

另外，解决队满队空的办法有三：

- ① 设置一个布尔变量以区别队满还是队空；
- ② 浪费一个元素的空间，用于区别队满还是队空。
- ③ 使用一个计数器记录队列中元素个数（即队列长度）。

我们常采用法②，即队头指针、队尾指针中有一个指向实元素，而另一个指向空闲元素。

判断循环队列队空标志是： $f = rear$ 队满标志是： $f = (r + 1) \% N$

5. 有利用两个栈 s_1, s_2 模拟一个队列时，如何用栈的运算实现队列的插入，删除以及判队空运算。请简述这些运算的算法思想。

答案：栈的特点是后进先出，队列的特点是先进先出。初始时设栈 s_1 和栈 s_2 均为空。（1）用栈 s_1 和 s_2 模拟一个队列的输入：设 s_1 和 s_2 容量相等。分以下三种情况讨论：若 s_1 未滿，则元素入 s_1 栈；若 s_1 滿， s_2 空，则将 s_1 全部元素退栈，再压栈入 s_2 ，之后元素入 s_1 栈；若 s_1 滿， s_2 不空（已有出队列元素），则不能入队。（2）用栈 s_1 和 s_2 模拟队列出队（删除）：若栈 s_2 不空，退栈，即是队列的出队；若 s_2 为空且 s_1 不空，则将 s_1 栈中全部元素退栈，并依次压入 s_2 中， s_2 栈顶元素退栈，这就是相当于队列的出队。若栈 s_1 为空并且 s_2 也为空，队列空，不能出队。（3）判队空若栈 s_1 为空并且 s_2 也为空，才是队列空。

五、阅读理解（每小题 5 分，共 20 分。至少要写出思路）

1. 【严题集 3.7①】按照四则运算加、减、乘、除和幂运算（ \uparrow ）优先关系的惯例，并仿照教材例 3-2 的格式，画出对下列算术表达式求值时操作数栈和运算符栈的变化过程：

$$A - B \times C / D + E \uparrow F$$

序号	OPTR 栈	OPND 栈	当前字符													备注 (操作)
			A	-	B	*	C	/	D	+	E	^	F	#		
1	#		•												push(OPND, 'A')	
2	#	A		•											push(OPTR, ' - ')	
3	# -	A			•										push(OPND, 'B')	
4	# -	AB				•									push(OPTR, ' * ')	
5	# - *	AB					•								push(OPND, 'C')	
6	# - *	ABC						•							归约, 令 $T_1 = B * C$	
7	# -	AT_1						•							push(OPTR, ' / ')	
8	# -	AT_1							•						push(OPND, 'D')	
9	# - /	AT_1D								•					归约, 令 $T_2 = T_1 / D$	
10	# - /	AT_2								•					归约, 令 $T_3 = A - T_2$	
11	#	T_2								•					push(OPTR, ' + ')	
12	# +	T_3									•				push(OPND, 'E')	
13	# +	T_3E										•			push(OPTR, ' ^ ')	
14	# + ^	T_3E											•		push(OPND, 'F')	
15	# + ^	T_3EF												•	归约, 令 $T_4 = E ^ F$	
16	# +	T_3T_4												•	归约, 令 $T_5 = T_3 + T_4$	
17	#	T_5												•	return(T_5)	

2. 【严题集 3.12②】写出下列程序段的输出结果(队列中的元素类型 QElem Type 为 char)。

```
void main() {
    Queue Q; Init Queue (Q);
    Char x='e'; y='c';
    EnQueue (Q, 'h'); EnQueue (Q, 'r'); EnQueue (Q, 'y');
    DeQueue (Q, x); EnQueue (Q, x);
    DeQueue (Q, x); EnQueue (Q, 'a');
    while (!QueueEmpty(Q)) { DeQueue (Q, y); printf(y); };
    Printf(x);
}
```

3. 【严题集 3.13②】利用一个栈，将队列中的元素逆置。

```
void reverseQueue(queue<int> &q) {
    stack<int> s;
    while (!q.empty()) {
        s.push(q.front());
        q.pop();
    }
    while (!s.empty()) {
        q.push(s.top());
    }
}
```

```

        s.pop();
    }
}

```

4. 简述以下算法的功能（栈和队列的元素类型均为 int）。

```

void algo3(Queue &Q){
Stack S; int d;
InitStack(S);
while(!QueueEmpty(Q)){
DeQueue (Q,d);  Push(S,d);
};
while(!StackEmpty(S)){
Pop(S,d); EnQueue (Q,d);
}
}

```

六、算法设计（每小题 5 分，共 15 分。至少要写出思路）

1. 【李春葆及严题集 3.19④】假设一个算术表达式中包含圆括弧、方括弧和花括弧三种类型的括弧，编写一个判别表达式中括弧是否正确配对的函数 `correct(exp,tag)`；其中：`exp` 为字符串类型的变量（可理解为每个字符占用一个数组元素），表示被判别的表达式，`tag` 为布尔型变量。

答：用堆栈 `st` 进行判定，将 `[`、`[` 或 `{` 入栈，当遇到 `]`、`]` 或 `}` 时，检查当前栈顶元素是否是对应的 `]`、`]` 或 `}`，若是则退栈，否则返回表示不配对。当整个算术表达式检查完毕时，若栈为空表示括号正确配对，否则不配对。

编程后的整个函数如下（李书 P31—32）

```

#define m0 100      /*m0 为算术表达式中最多字符个数*/
correct(exp,tag)
char exp[m0];
int tag;
{char st[m0];
  int top=0, i=1;
  tag=1;
  while (i<=m0 && tag)
  {if (exp[i] == '(' || exp[i] == '[' || exp[i] == '{') /*遇到'('、'['或'{'，则将其入栈*/
    {top++;
     st[top]=exp[i];
    }
    if (exp[i] == ')') /*遇到')'，若栈顶是'('，则继续处理，否则以不配对返回*/
      if(st[top] == '(') top--;
      else tag=0;
    if (exp[i] == ']') /*遇到']'，若栈顶是'['，则继续处理，否则以不配对返回*/
      if(st[top] == '[') top--;
      else tag=0;
    if (exp[i] == '}') /*遇到'}'，若栈顶是'{'，则继续处理，否则以不配对返回*/

```

```

        if(st[top]!='{') top--;
        else tag=0;
        i++;
    }
    if(top>0)tag=0; /*若栈不空，则不配对*/
}

```

严题集对应答案：

3.19

Status AllBrackets_Test(char *str)//判别表达式中三种括号是否匹配

```

{
    InitStack(s);
    for(p=str;*p;p++)
    {
        if(*p=='('||*p=='['||*p=='{') push(s,*p);
        else if(*p==')'||*p==']'||*p=='}')
        {
            if(StackEmpty(s)) return ERROR;
            pop(s,c);
            if(*p==')'&&c!='(') return ERROR;
            if(*p==']'&&c!='[') return ERROR;
            if(*p=='}'&&c!='{') return ERROR; //必须与当前栈顶括号匹配
        }
    }
    if(!StackEmpty(s)) return ERROR;
    return OK;
}
//AllBrackets_Test

```

2. 【统考书 P60 4-15】假设一个数组 `squ[m]` 存放循环队列的元素。若要使这 m 个分量都得到利用，则需另一个标志 `tag`，以 `tag` 为 0 或 1 来区分尾指针和头指针值相同时队列的状态是“空”还是“满”。试编写相应的入队和出队的算法。

解：这就是解决队满队空的三种办法之① 设置一个布尔变量以区别队满还是队空（其他两种见简答题）；

思路：一开始队空，设 `tag=0`，若从 `rear` 一端加到与 `front` 指针相同时，表示入队已满，则令 `tag=1`；

若从 `front` 一端加到与 `rear` 指针相同时，则令 `tag=0`，表示出队已空。

3. 【严题集 3.31③】试写一个算法判别读入的一个以 '@' 为结束符的字符序列是否是“回文”。

答：编程如下：

```

int Palindrome_Test()//判别输入的字符串是否回文序列,是则返回 1,否则返回 0
{

```

```
InitStack(S);InitQueue(Q);
while((c=getchar())!='@')
{
    Push(S,c);EnQueue(Q,c); //同时使用栈和队列两种结构
}
while(!StackEmpty(S))
{
    Pop(S,a);DeQueue(Q,b);
    if(a!=b) return ERROR;
}
return OK;
} //Palindrome_Test
```