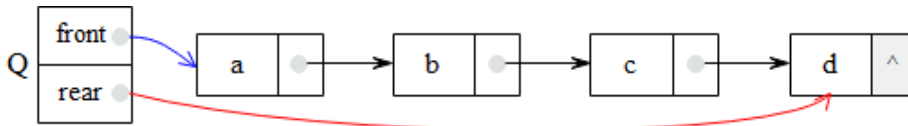


得分

一、单项选择题 (X 题, 每题 X 分, 共 X 分)

评分标准: 每题回答正确得 1.5 分, 错误不得分!

1、在下列链队列 Q 中, 元素 a 出队的操作序列为 ( B )。



- A.  $p=Q.front \rightarrow next; p \rightarrow next=Q.front \rightarrow next;$       B.  $p=Q.front; Q.front=p \rightarrow next;$   
 C.  $p=Q.rear \rightarrow next; p \rightarrow next=Q.rear \rightarrow next;$       D.  $p=Q \rightarrow next; Q \rightarrow next=p \rightarrow next;$

2、设有一个栈, 元素的进栈次序为 A, B, C, D, E, 下列是不可能的出栈序列 ( C )。

- A. A, B, C, D, E      B. B, C, D, E, A      C. E, A, B, C, D      D. E, D, C, B, A

3、在一个具有 n 个单元的顺序栈中, 假定以地址低端 (即 0 单元) 作为栈底, 以 top 作为栈顶指针, 当做出栈处理时, top 变化为 ( C )。

- A. top 不变      B.  $top=0$       C.  $top--$       D.  $top++$

4、向一个栈顶指针为 hs 的无头结点的链栈中插入一个 s 结点时, 应执行 ( B )。

- A.  $hs \rightarrow next=s;$       B.  $s \rightarrow next=hs; hs=s;$   
 C.  $s \rightarrow next=hs \rightarrow next; hs \rightarrow next=s;$       D.  $s \rightarrow next=hs; hs=hs \rightarrow next;$

5、在具有 n 个单元的顺序存储的循环队列中, 假定 front 和 rear 分别为队头指针和队尾指针, 则判断队满的条件为 ( D )。

- A.  $rear \% n == front$       B.  $(front+1) \% n == rear$   
 C.  $rear \% n - 1 == front$       D.  $(rear+1) \% n == front$

6、在具有 n 个单元的顺序存储的循环队列中, 假定 front 和 rear 分别为队头指针和队尾指针, 则判断队空的条件为 ( C )。

- A.  $rear \% n == front$       B.  $front+1 = rear$       C.  $rear == front$       D.  $(rear+1) \% n == front$

7、设指针变量 front 表示链式队列的队头指针, 指针变量 rear 表示链式队列的队尾指针, 指针变量 s 指向将要入队列的结点 X, 则入队列的操作序列为 ( C )。

- A.  $front \rightarrow next=s; front=s;$       B.  $s \rightarrow next=rear; rear=s;$   
 C.  $rear \rightarrow next=s; rear=s;$       D.  $s \rightarrow next=front; front=s;$

8、设输入序列 1、2、3、...、n 经过栈作用后, 输出序列中的第一个元素是 n, 则输出序列中的第 i 个输出元素是 ( C )。

- A. n-i      B. n-1-i      C. n+1-i      D. 不能确定

- 9、在一个链队列中，假定 front 和 rear 分别为队首和队尾指针，则删除一个结点的操作为( **A** )。
- A. front=front->next      B. rear=rear->next      C. rear=front->next      D. front=rear->next
- 10、一个栈的输入序列为 1 2 3，则下列序列中不可能是栈的输出序列的是 ( **C** )。
- A. 2 3 1      B. 3 2 1      C. 3 1 2      D. 1 2 3
- 11、栈和队列的共同特点是 ( **A** )。
- A. 只允许在端点处插入和删除元素      B. 都是先进后出  
C. 都是先进先出      D. 没有共同点
- 12、一个栈的入栈序列为 1, 2, 3, ..., n，其出栈序列是  $p_1, p_2, p_3, \dots, p_n$ 。若  $p_2=3$ ，则  $p_3$  可能取值的个数是 ( **C** )。
- A. n-3      B. n-2      C. n-1      D. 无法确定
- 13、用链接方式存储的队列，在进行插入运算时 ( **D** )。
- A. 仅修改头指针      B. 头、尾指针都要修改  
C. 仅修改尾指针      D. 头、尾指针可能都要修改
- 14、栈的插入和删除操作在 ( **A** ) 进行。
- A. 栈顶      B. 栈底      C. 任意位置      D. 指定位置
- 15、由两个栈共享一个向量空间的好处是 ( **B** )。
- A. 减少存取时间，降低下溢发生的机率      B. 节省存储空间，降低上溢发生的机率  
C. 减少存取时间，降低上溢发生的机率      D. 节省存储空间，降低下溢发生的机率
- 16、设栈 S 和队列 Q 的初始状态为空，元素 E1、E2、E3、E4、E5 和 E6 依次通过栈 S，一个元素出栈后即进入队列 Q，若 6 个元素出列的顺序为 E2、E4、E3、E6、E5 和 E1，则栈 S 的容量至少应该是 ( **C** )。
- A. 6      B. 4      C. 3      D. 2
- 17、设顺序循环队列 Q[0: M-1] 的头指针和尾指针分别为 F 和 R，头指针 F 总是指向队头元素的前一位置，尾指针 R 总是指向队尾元素的当前位置，则该循环队列中的元素个数为 ( **C** )。
- A. R-F      B. F-R      C. (R-F+M)%M      D. (F-R+M)%M
- 18、设用链表作为栈的存储结构则退栈操作 ( **B** )。
- A. 必须判别栈是否为满      B. 必须判别栈是否为空  
C. 判别栈元素的类型      D. 对栈不作任何判别
- 19、设输入序列是 1、2、3、……、n，经过栈的作用后输出序列的第一个元素是 n，则输出序列中第 i 个输出元素是 ( **C** )。

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

20、设指针变量  $top$  指向当前链式栈的栈顶，则删除栈顶元素的操作序列为 ( D )。

- A.  $top=top+1;$                       B.  $top=top-1;$   
C.  $top->next=top;$                       D.  $top=top->next;$

21、队列是一种 ( A ) 的线性表。

- A. 先进先出                      B. 先进后出                      C. 只能插入                      D. 只能删除

22、设输入序列为 1、2、3、4、5、6，则通过栈的作用后可以得到的输出序列为 ( B )。

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

得分

## 二、填空题 (X 题，每题 X 分，共 X 分)

评分标准：每空回答正确得 1 分，错误不得分，不完全正确酌情给分！

- 1、栈和队列都是一种 【1】特殊（操作受限） 的线性表，栈的插入和删除只能在 【2】栈顶 进行。
- 2、对于一个栈作进栈运算时，应先判别栈是否为 【3】栈满，作退栈运算时，应先判别栈是否为 【4】栈空，当栈中元素为  $m$  时，作进栈运算时发生上溢，则说明栈的可用最大容量为 【5】 $m$ 。为了增加内存空间的利用率和减少发生上溢的可能性，由两个栈共享一片连续的内存空间时，应将两栈的 【6】栈底 分别设在这片内存空间的两端，这样只有当 【7】两个栈的栈顶在栈空间的某一位置相遇 时才产生上溢。
- 3、设有一空栈，现有输入序列 1, 2, 3, 4, 5，经过 push, push, pop, push, pop, push, push 后，输出序列是 【8】2、3。
- 4、无论对于顺序存储还是链式存储的栈和队列来说，进行插入或删除运算的时间复杂度均相同为 【9】 $O(1)$ 。
- 5、队列的插入操作是在队列的 【10】尾部 进行，删除操作是在队列的 【11】首 进行。
- 6、当用长度为  $N$  的数组顺序存储一个栈时，假定用  $top==N$  表示栈空，则表示栈满的条件是 【12】 $top==0$ （要超出才为满）。
- 7、用具有  $n$  个元素的一维数组存储一个循环队列，则其队首指针总是指向队首元素的 【13】前一个位置，该循环队列的最大长度为 【14】 $n-1$ 。
- 8、当堆栈采用顺序存储结构时，栈顶元素的值可用 【15】 $S.stack[S.top];$  表示；当堆栈采用链接存储结构时，栈顶元素的值可用 【16】 $HS \rightarrow data;$  表示。

- 9、栈顶的位置是随着 【17】进栈和退栈 操作而变化的。
- 10、设顺序循环队列  $Q[0:m-1]$  的队头指针和队尾指针分别为  $F$  和  $R$ ，其中队头指针  $F$  指向当前队头元素的前一个位置，队尾指针  $R$  指向当前队尾元素所在的位置，则出队列的语句为  $F =$  【18】 $(F+1) \% m$ ；。
- 11、下面程序段的功能实现数据  $x$  进栈，要求在下划线处填上正确的语句。
- ```
typedef struct {
    int s[100];
    int top;
} sqstack;

void push(sqstack &stack, int x) {
    if(stack.top==m-1) printf("overflow");
    else {
        【19】 stack.top++;
        【20】 stack.s[stack.top]=x (两者调换亦可);
    }
}
```
- 12、设输入序列为 1、2、3，则经过栈的作用后可以得到 【21】5 种不同的输出序列。
- 13、不论是顺序存储结构的栈还是链式存储结构的栈，其入栈和出栈操作的时间复杂度均为 【22】 $O(1)$ 。
- 14、设有一个顺序循环队列中有  $M$  个存储单元，则该循环队列中最多能够存储 【23】 $M-1$  个队列元素；当前实际存储 【24】 $(R-F+M)\%M$  个队列元素（设头指针  $F$  指向当前队头元素的前一个位置，尾指针指向当前队尾元素的位置）。
- 15、设有一个顺序共享栈  $S[0:n-1]$ ，其中第一个栈项指针  $top1$  的初值为 -1，第二个栈项指针  $top2$  的初值为  $n$ ，则判断共享栈满的条件是 【25】 $top1+1=top2$ 。
- 16、栈的插入和删除只能在栈的栈顶进行，后进栈的元素必定先出栈，所以又把栈称为 【26】**FILO（后进先出）** 表；队列的插入和删除运算分别在队列的两端进行，先进队列的元素必定先出队列，所以又把队列称为 【27】**FIFO（先进先出）** 表。
- 17、设  $F$  和  $R$  分别表示顺序循环队列的头指针和尾指针，则判断该循环队列为空的条件为 【28】 $F==R$ 。
- 18、设某顺序循环队列中有  $m$  个元素，且规定队头指针  $F$  指向队头元素的前一个位置，队尾指针  $R$  指向队尾元素的当前位置，则该循环队列中最多存储 【29】 $m-1$  队列元素。

得分

三、判断题（X 题，每题 X 分，共 X 分。正确填 ‘T’，错误 “F”。）

评分标准：每题回答正确得 2 分，错误不得分！

- 1、不论是入队列操作还是入栈操作，在顺序存储结构上都需要考虑“溢出”情况。（ T ）
- 2、入栈操作和入队列操作在链式存储结构上实现时不需要考虑栈溢出的情况。（ T ）

得分

#### 四、简答题（X 题，每题 X 分，共 X 分）

评分标准：每题回答完全正确得 5 分，其余按得分点给分！

- 1、假定有四个元素 A, B, C, D 依次进栈，进栈过程中允许出栈，试写出所有可能的出栈序列。

（评分标准：第 1 点 4 分，第 2 点 1 分！）

答：共有 14 种可能的出栈序列，即为：ABCD, ABDC, ACBD, ACDB, BACD, ADCB, BADC, BCAD, BCDA, BDCA, CBAD, CBDA, CDBA, DCBA。

- 2、什么是队列的上溢现象？一般有几种解决方法，试简述之。（评分标准：第 1 点 4 分，第 2 点 1 分！）

答：在队列的顺序存储结构中，设队头指针为 front，队尾指针为 rear，队列的容量（即存储的空间大小）为 maxnum。当有元素要加入队列（即入队）时，若 rear=maxnum，则会发生队列的上溢现象，此时就不能将该元素加入队列。对于队列，还有一种“假溢出”现象，队列中尚余有足够的空间，但元素却不能入队，一般是由于队列的存储结构或操作方式的选择不当所致，可以用循环队列解决。

一般地，要解决队列的上溢现象可有以下几种方法：

（1）可建立一个足够大的存储空间以避免溢出，但这样做往往会造成空间使用率低，浪费存储空间。

（2）要避免出现“假溢出”现象可用以下方法解决：

第一种：采用移动元素的方法。每当有一个新元素入队，就将队列中已有的元素向队头移动一个位置，假定空余空间足够。

第二种：每当删去一个队头元素，则可依次移动队列中的元素总是使 front 指针指向队列中的第一个位置。

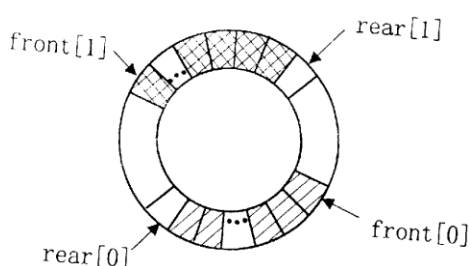
第三种：采用循环队列方式。将队头、队尾看作是一个首尾相接的循环队列，即用循环数组实现，此时队首仍在队尾之前，作插入和删除运算时仍遵循“先进先出”的原则。

得分

#### 五、程序填空题（X 题，每题 X 分，共 X 分）

评分标准：每空回答正确得 1 分，错误不得分，不完全正确酌情给分！

- 1、假设两个队列共享一个循环向量空间（如下图），其类型 Queue2 定义如下。对于 i=0 或 1，front[i]和 rear[i]分别为第 i 个队列的头指针和尾指针。请对以下算法填空，实现第 i 个队列的入队操作。请在下划线处填上正确的语句。（评分标准：第 1 点 3 分！第 2 点 2 分！）



```
typedef struct {
    DateType data[MaxSize];
    int front[2], rear[2];
} Queue2
```

//若第  $i$  个队列不满，则元素  $x$  入队列，并返回 1；否则返回 0

```
int EnQueue(Queue2*Q,int i,DateType x) {
    if(i<0||i>1)
        return 0;

    if(Q->rear[i]==Q->front[【1】 (i+1)%2 (或 1-i)])
        return 0;

    Q->data[【2】 Q->rear[i]]=x;
    Q->rear[i]=[【3】 (Q->rear[i]+1)%Maxsize];

    return 1;
}
```

得分

六、阅读程序题 (X 题，每题 X 分，共 X 分)

评分标准：每空回答正确得 1 分，错误不得分，不完全正确则酌情给分！

1、假定从键盘上输入一批整数，依次为：78 63 45 30 91 34 -1，请写出输出结果。(评

分标准：第 1 点 4 分，第 2 点 1 分！)

```
# include < iostream.h>
# include < stdlib.h >
const int stackmaxsize = 30;
typedef int elemtype;
struct stack {
    elemtype stack[stackmaxsize];
    int top;
};

# include "stack.h"
void main ( ) {
    stack a;
    initstack(a);
    int x;
```

```

cin >>x;
while (x!= -1) {
    push (a, x );
    cin >>x;
}

while (!stackempty (a))
    cout <<pop (a) <<" " ;
cout <<endl;
}

```

答：算法的输出结果为：34 91 30 45 63 78;

## 得分 七、编程题 (X 题, 每题 X 分, 共 X 分)

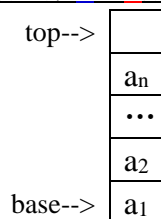
评分标准：每小题回答正确得 10 分，不完全正确则按得分点给分。

- 1、设顺序栈及其结点定义如下，请设计算法，将栈顶元素出栈并存入 e 中。实现算法的函数原型为：void **pop**(Stack &S, Elemtype &e); (评分标准：第 1 点 4 分，第 2 点 1 分!)

```

typedef struct {
    Elemtype *base; //栈底指针
    Elemtype *top;  //栈顶指针
} Stack;

```



答：void **pop**(Stack &S, Elemtype &e) {  
 if( S.top==S.base )  
 return ERROR;  
  
 S.top--;  
 e=\*s.top;  
}

- 2、假定用一个单循环链表来表示队列（也称为循环队列），该队列只设一个队尾指针，不设队首指针，试编写下列各种运算的算法：(评分标准：第 1 点 4 分，第 2 点 1 分!)

- (1) 向循环链队列插入一个元素值为 x 的结点；
- (2) 从循环链队列中删除一个结点。

答：本题是对一个循环链队列做插入和删除运算，假设不需要保留被删结点的值和不需要回收结点，算法描述如下：

(1) 插入（即入队）算法

```

void insert(LinkedList *rear, elemtype x) {
    //设循环链队列的队尾指针为 rear,x 为待插入的元素
    LinkedList *p;

```

```

p=(LinkedList *)malloc(sizeof(LinkedList));
if(rear==NULL) { //如为空队，建立循环链队列的第一个结点
    rear=p;
    rear->next=p;    //链接成循环链表
} else { //否则在队尾插入 p 结点
    p->next=rear->next;
    rear->next=p;
    rear=p;
}
}

```

(2) 删除（即出队）算法

```

void delete(LinkedList *rear) {
    //设循环链队列的队尾指针为 rear
    if(rear==NULL) //空队
        printf("underflow\n");
    if(rear->next==rear) //队中只有一个结点
        rear=NULL;
    else
        rear->next=rear->next->next; //rear->next 指向的结点为循环链队列的队头结点
}

```