已知一算术表达式的后缀形式为ABC\*+DE/-，其前缀形式为( \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ )。

A. -A+B\*C/DE

B.  -A+B\*CD/E

C. - +\*ABC/DE

D. - +A\*BC/DE

**正确答案:**D

循环队列存储在数组A[0..m]中，则入队时队尾的操作为 \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 。

A. rear=rear+1          B. rear=(rear+1)%(m-1)

C. rear=(rear+1)%m       D. rear=(rear+1)%(m+1)

**正确答案:**D

一个栈的输入序列为1,2,3...n，若输出序列的第一个元素是n，则输出第i(1<=i<=n)个元素是 \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

A. 不确定

B. n-i+1

C. i

D. n-i

**正确答案:**B

设栈S和队列Q的初始状态均为空，元素a,b,c,d,e, f, g依次进入栈S。若每个元素出栈后立即进入队列Q, 且7个元素出队的顺序是b,e,f,d,c,a,g，则栈S的容量至少是 \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 。

A.1    B.2    C.3    D.4

**正确答案:**d

己知循环队列存储在一维数组A[0...n-1]中，且队列非空时front和rear分别指向队头元素和队尾元索。若初始时队列为空，且要求第1个进入队列的元素存储在A[0]处，则初始时front和rear的值分别是 \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 。

A.0,0   B.0,n-1   C.n-l,0    D.n-1,n-l

**正确答案:**B

**编程题：**

1. **质因数分解逆序输出（C++）**

**质因数分解(短除法)+栈**

**消消乐（C++）**

【问题描述】利用栈的基本操作，设计算法，对给定字符串s进行处理，依次消除相等的相邻字符对，最后使得其中任意的相邻字符不相等，其他字符的次序不变，如"stxyyxeddeff20"最后变为"st20"。

**Gettop（） 加规则 相等 pop，不等 push**

**奇数逆置**

**【**问题描述】

使用已学的各种数据结构及基本操作，设计算法void reverseOdd(CirQueue<int>& sq) ，对参数所给定的队列sq中的整数进行操作，将其中奇数的顺序进行逆置，偶数的顺序维持不变。例如，sq中队头至队尾的元素为：(14, 13, 17, 8, 4, 10, 11, 4, 15, 18, 19)，调用reverseOdd后，则sq的内容变为：(14, 19, 15, 8, 4, 10, 11, 4, 17, 18, 13)。

**注意不能访问quque中某个元素**

**常用方法**

**依次从原队列出队：**

**Array 数组记录even还是odd**

**偶数放置到队列Queue\_even**

**奇数放置到栈Stack\_odd**

**按照Array，碰到even，出队，碰到odd，出栈，依次放置到原队列**

**如果要使用原队列呢？**

**4. 栈实现队列**

【问题描述】

请使用栈类实现队列的类。

**解决思路：要使用到两个栈**

**假设分别为push\_Stack, pop\_Stack**

**进队，直接往push\_stack里放置；出队，要把push\_stack里最下面的元素弹出来，所以可以把push\_stack里元素逐一出栈压入到第2个栈pop\_stack中，然后弹出pop\_stack中的栈顶元素即可，做完该操作后，把pop\_stack中剩余元素逐一弹出并压入到push\_stack.**

**上述过程其实可以优化，比如，弹出pop\_stack栈顶元素后，可以暂时不回放到push\_stack, 如果下一个操作还是出队，就可以继续弹出pop\_stack栈顶，但是发生进队时，需要把pop\_stack里的所有元素放回到push\_stack中，所以上面过程可以优化：**

**进队不能直接往push\_stack里放，要判断一下pop\_stack里有无元素，如果没有，可以直接放，否则从pop\_stack里弹回所有元素到push\_stack中才可以进队；出队时，可以看pop\_stack是否有元素，如果有，直接取，没有，则要从push\_stack里取出所有元素压入到pop\_stack后，才能弹出栈顶元素，当出队元素。**

**其它班布置的题目：**

1. **简答题：**

**删除除队列中最后一个值为item的元素：**

调用栈或队列类的方法，设计算法删除队列中最后一个值为item的元素，即离队首最远的值为item的元素。

**假设队列q， 栈s**

**Q出队压入栈s，s出栈进队列Q，后面这个过程碰到第一item时不进入Q**

1. **升温等待天数（C++）**

题目内容：

设计算法，对气温数组a，生成一个升温等待天数数组b。即b对应位置的元素是你需要等待多少天，温度才会升至超过该日的温度。如果之后温度都不会升高，请在该位置用 0 来代替。

输入格式:

气温数组，包含若干个整数

输出格式：

整数组成的数组

输入样例：

23 24 25 21 19 22 26 23

输出样例：

1 1 4 2 1 1 0 0

**算法时间复杂度和空间复杂度分别为O(n^2)、O(1)或分别为O(n)、O(n)**

**如果栈顶存储的下标对应的数组元素较小，则将该下标出栈，并计算其与数组当前遍历的元素下标的差值，存入辅助数组的栈顶下标位置，将当前遍历的元素继续与栈顶元素比，如果栈顶还是小，继续重复前面的步骤，如果当前的大于栈顶元素，则将当前下标存入栈中。**

**vector<int> dailyTemp(vector<int> alst) {**

**int n = alst.size();**

**int j;**

**vector <int> blst (n);**

**SeqStack<tuple<int, int>> s; //第一个int表温度，第2个int表等待天数**

**if (n > 0) {**

**s.Push(make\_tuple(alst[0], 0));**

**}**

**for (int i = 1; i < n; i++) {**

**while (!s.Empty() && alst[i] > get<0>(s.GetTop()) ){**

**j = get<1>(s.Pop());**

**blst[j] = i - j;**

**}**

**s.Push(make\_tuple(alst[i], i));**

**}**

**return blst;**

**}**

**3.**

|  |  |
| --- | --- |
| **1** | **假设某公司于2021年1月、4月、9月分别购进A材料100吨，并在当年6月、11月分别卖出100吨。这5个月的材料成交价格如下。**  **时间 1月 4月 6月 9月 11月**  **单价 $10 $30 $20 $50 $30**  **请确定如果采用以下策略，一年总共盈利或亏损多少？假设这一年公司剩余的100吨材料不计算在内。**  **（1）采用先进先出的策略进行买卖，即6月卖出的材料是1月买入的；则该公司在该项目上是 \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 的（填盈利或亏损），盈利或亏损的金额是 \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 美元。**  **（2）采用后进先出的策略进行买卖，即6月卖出的材料是4月买入的。则该公司在该项目上是 \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 的（填盈利或亏损），盈利或亏损的金额是 \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 美元。**  **正确答案: 盈利 | 1000 | 亏损 | 3000** |