1. 调试成功程序及说明

1、题目：编程实现书P32 ADT Stack 基本操作9个，用顺序存储结构实现；

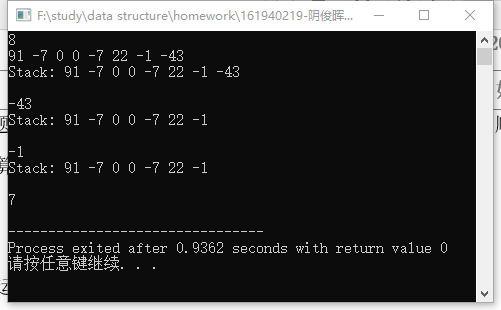
算法思想：

出栈：先返回top所指元素，再让top减1

入栈：先让top加1，再在top所指位置增加元素

返回栈顶元素：直接返回top所指元素

运行结果：



结果分析：

T(n) = O(1)

S(n) = O(1)

附源程序。

2、题目：编程实现书P48 ADT Queue 基本操作9个，用链式存储结构实现；

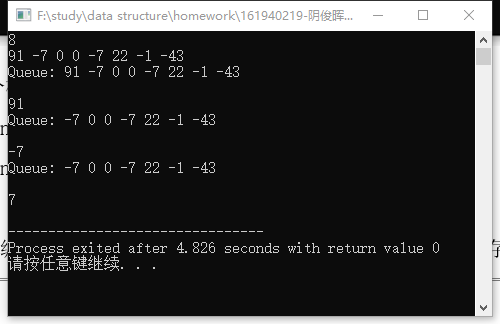
算法思想：

出队：先让front指向后一元素，再返回front所指元素。

入队：先在rear所指结点之后增加结点，再让rear指向增加的结点。

返回队首元素：直接返回front所指结点。

运行结果：



结果分析：

T(n) = O(1)

S(n) = O(1)

附源程序。

3、题目：利用栈操作实现八皇后问题求解。

算法思想：

让cur\_row记录当前行数，初始化为1。

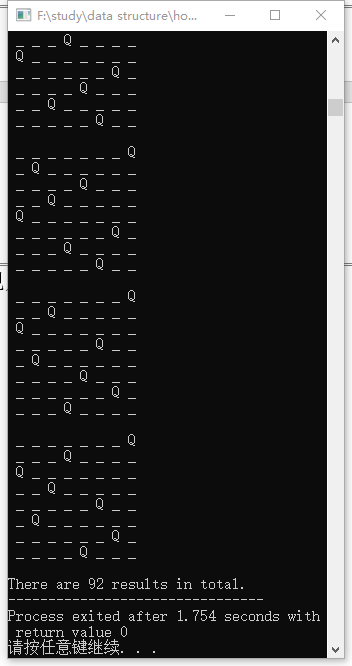
主循环：若回溯后cur\_row为0，循环退出，否则一直执行。

前进：若当前行数为N,输出结果，若当前行数在1到N-1之间，在寻找下一行皇后的位置，若成功找到，cur\_row加1，继续找下一行皇后的位置，若没找到，回溯。

回溯：只要没有退到第0行，就一直回溯，直到皇后找到下一个位置为止。

若在当前行中，处于最后一列，cur\_row减1，直到当前行的当前列不是最后一列为止。若不在最后一列，列数加1，直到这一列合法为止。

运行结果：



结果分析：

T(n) = O()

S(n) = O(1)

附源程序。

4、题目：CSP题目

问题描述：小明今天生日，他有n块蛋糕要分给朋友们吃，这n块蛋糕（编号为1到n）的重量分别为a1, a2, …, an。小明想分给每个朋友至少重量为k的蛋糕。小明的朋友们已经排好队准备领蛋糕，对于每个朋友，小明总是先将自己手中编号最小的蛋糕分给他，当这个朋友所分得蛋糕的重量不到k时，再继续将剩下的蛋糕中编号最小的给他，直到小明的蛋糕分完或者这个朋友分到的蛋糕的总重量大于等于k。  
　　请问当小明的蛋糕分完时，总共有多少个朋友分到了蛋糕。

输入格式：

　　输入的第一行包含了两个整数n, k，意义如上所述。  
　　第二行包含n个正整数，依次表示a1, a2,…, an。

输出格式：

　　输出一个整数，表示有多少个朋友分到了蛋糕。

样例输入：

6 9  
2 6 5 6 3 5

样例输出：

3

样例说明：

　　第一个朋友分到了前3块蛋糕，第二个朋友分到了第4、5块蛋糕，第三个朋友分到了最后一块蛋糕。

评测用例规模与约定：

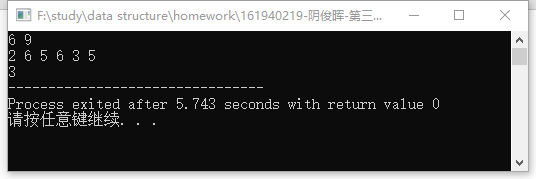
　　对于所有评测用例，1≤n≤1000，1≤k≤10000，1≤ai≤1000。

算法思想：

先让所有蛋糕重量依次入队。

一次循环：让所有蛋糕重量依次出队，若出队的蛋糕重量不够k就继续出队，若够，则让分到蛋糕的总人数加1，循环继续。若队空，循环终止，再判断最后一个人分到的蛋糕重量，若小于k，分到蛋糕的总人数再加1，若等于，不进行操作。

运行结果：



结果分析：

T(n) = O(n)

S(n) = O(n)

附源程序。

5、题目：CSP题目

问题描述：给定n个数，请找出其中相差（差的绝对值）最小的两个数，输出它们的差值的绝对值。

输入格式：

　　输入第一行包含一个整数n。  
　　第二行包含n个正整数，相邻整数之间使用一个空格分隔。

输出格式：

　　输出一个整数，表示答案。

样例输入：

5  
1 5 4 8 20

样例输出：

1

样例说明：

　　相差最小的两个数是5和4，它们之间的差值是1。

样例输入：

5  
9 3 6 1 3

样例输出：

0

样例说明：

　　有两个相同的数3，它们之间的差值是0.

数据规模和约定：

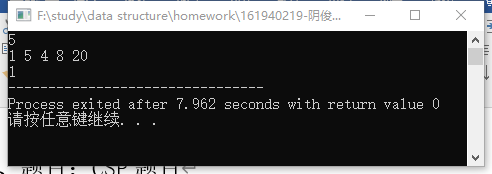
　　对于所有评测用例，2≤n≤1000，每个给定的整数都是不超过10000的正整数。

算法思想：

先用冒泡排序，使数组中元素升序排列。

将最小差值初始化为无穷大，然后遍历数组，若第i个元素与第i+1个元素差值比最小差值小，则更新最小差值，遍历结束之后输出最小差值。

运行结果：



结果分析：

T(n) = O()

S(n) = O(1)

附源程序。

-------------------------------------------------------------------------

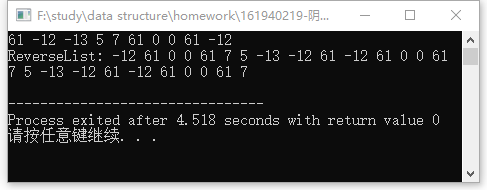
选做题**2.32 已知有一个单向循环链表，其每个结点中含三个域：pre，data和next，其中data为数据域，next为指向后继结点的指针域，pre也为指针域，但它的值为空，试编写算法将此单向循环链表改为双向循环链表，即使pre成为指向前驱结点的指针域。**

算法思想：

令p1指向头结点，p2指向p1后继。

一重循环，每次令p2的pre指向p1，然后p1指向p1后继，p2指向p2后继。

运行结果：



结果分析：

T(n) = O(n)

S(n) = O(1)

附源程序。

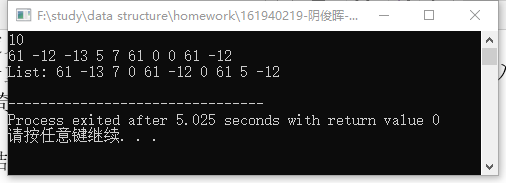
选做题**2.37 设以带头结点的双向循环链表表示的线性表。试写一时间复杂度O(n)的算法，将L改造为。**

算法思想：

定义一个指向头结点的指针p。

一重循环，遍历链表，若p指向偶数结点，先将其插入表尾，再将其删除，若p指向奇数结点，无操作。

运行结果：



结果分析：

T(n) = O(n)

S(n) = O(1)

附源程序。

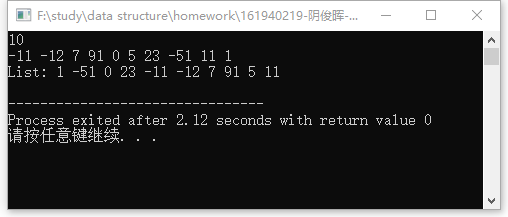
选做题**2.38 设有一个双向循环链表，每个结点中除有pre，data和next三个域外，还增设了一个访问频度域freq。在链表被起用之前，频度域freq的值均初始化为零，而每当对链表进行一次Locate(L,x)的操作后，被访问的结点（即元素值等于x的结点）中的频度域freq的值便增1，同时调整链表中结点之间的次序，使其按访问频度非递增的次序顺序排列，以便始终保持被频繁访问的结点总是靠近表头结点。试编写符合上述要求的Locate操作的算法。**

算法思想：

依据频率更新的函数：先利用Find函数找到x应当插入的新位置，再插入将x中的数据插入到新位置，并使其频率加1，再删除原结点。

Find函数：只要p的前一结点频率比x频率小，p就前移，直到找到位置，并返回。

运行结果：



结果分析：

T(n) = O(n)

S(n) = O(1)

附源程序。

二、代码行数及小结

|  |  |
| --- | --- |
| 题目 | 代码行数 |
| 1 | 105 + 22 = 127 |
| 2 | 141 + 20 = 161 |
| 3 | 132 |
| 4 | 44 |
| 5 | 49 |
| 选2.32 | 30 |
| 选2.37 | 40 |
| 选2.38 | 26 |
| 总计 | 609 |