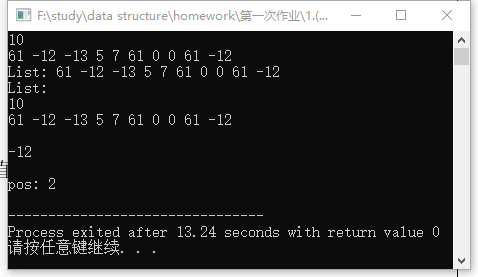
1、题目：编程实现书P12 ADT List 基本操作13个：

* 1. 用顺序存储结构实现； （2）用链式存储结构实现；

运行结果：



算法思想：主要描述插入操作与删除操作的思路

（1）数组实现

插入元素：若在表尾插入直接插入即可，若在表内部插入，需从后往前逐项移位再插入

结果分析：

T(n) = O(n)

S(n) = O(1)

删除元素：若在表尾删除直接删除即可，若在表内部删除，先删除再从前往后逐项移位

结果分析：

T(n) = O(n)

S(n) = O(1)

（2）链表实现

插入元素：若在表尾插入还需更新rear指针，若在表内部插入，先申请结点所需空间赋给p，再让p指向cur的后继，再使cur指向p

结果分析：

T(n) = O(1)

S(n) = O(1)

删除元素：若在表尾删除还需更新rear指针，若在表内部删除，先令tem指向所删结点，再使所删结点的前驱指向所删结点的后继，再释放tem

结果分析：

T(n) = O(n)

S(n) = O(1)

附源程序。

2、题目：设元素值为整型的线性表L，分别采用顺序结构和链式结构存储，编写函数，实现线性表的就地逆置（书P31 4）。

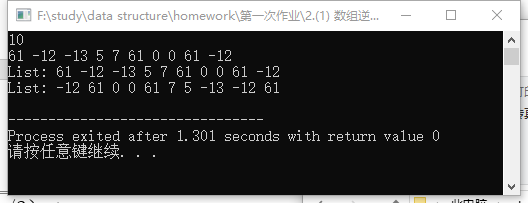
（1）数组实现

算法思想：

若表长为1，不逆转。

若表长大于1，使第i个元素与第length-i个元素依次交换（i遍历0到length/2）

运行结果：



结果分析：

T(n) = O(n)

S(n) = O(1)

* 1. 链表实现

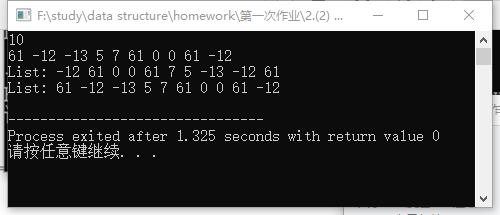
算法思想：

若表长为1，不逆转。

若表长大于1，定义cur、next、tem三个指针，分别指向表中第i、i+1、i+2个结点，主循环中，每第i次操作使得第i+1个结点指向第i个结点，tem记录next的后一个结点，以更新next，同时更新cur，tem为空时循环终止，此时令表头结点指向next。

为使得表尾结点的next值为空，一开始还需定义tem2，用来记忆未逆置前的第一个结点，循环终止前令其next置空。

运行结果：



结果分析：

T(n) = O(n)

S(n) = O(1)

附源程序。

3、题目：设线性表L，元素值为整型的且存在相同值，分别采用顺序结构和链式结构存储，编写函数，利用原空间，删除重复的元素值。

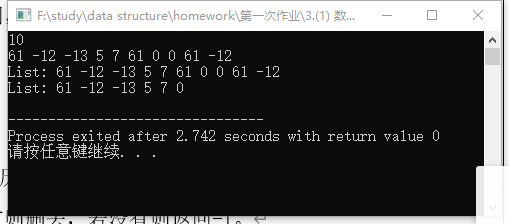
（1）数组实现

算法思想：

共两层循环。

第一层循环使得i遍历数组，第二层循环从i位置之后遍历数组，检查第i个元素重复元素的位置，若有则删去，若没有则返回-1。

运行结果：



结果分析：

T(n) = O()

S(n) = O(1)

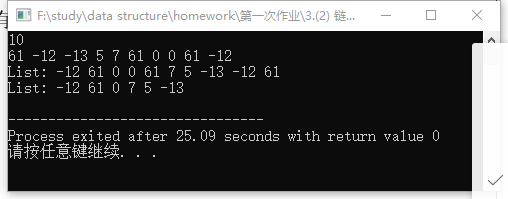
（2）链表实现

算法思想：

共两层循环。

第一层循环使得i遍历链表，第二层循环从i位置之后遍历链表，检查第i个元素重复元素的位置，若有则删去，若没有则返回空指针。

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：一次放学的时候，小明已经规划好了自己回家的路线，并且能够预测经过各个路段的时间。同时，小明通过学校里安装的“智慧光明”终端，看到了**出发时刻路上**经过的所有红绿灯的指示状态。请帮忙计算小明此次回家所需要的时间。

输入格式：

　　输入的第一行包含空格分隔的三个正整数 r、y、g，表示红绿灯的设置。这三个数均不超过 106。  
　　输入的第二行包含一个正整数 n，表示小明总共经过的道路段数和路过的红绿灯数目。  
　　接下来的 n 行，每行包含空格分隔的两个整数 k、t。k=0 表示经过了一段道路，将会耗时 t 秒，此处 t 不超过 106；k=1、2、3 时，分别表示**出发时刻**，此处的红绿灯状态是红灯、黄灯、绿灯，且倒计时显示牌上显示的数字是 t，此处 t 分别不会超过 r、y、g。

输出格式：

　　输出一个数字，表示此次小明放学回家所用的时间。

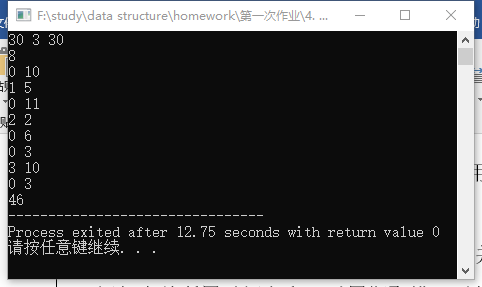
算法思想：

初始化sum为0。

一层循环，每次处理第i行数据，利用函数WaitCost（）计算所需时间并累加入sum。

函数WaitCost（）：按照顺序绿黄红，并以绿黄红总时间为一周期。令当前时间与当前所用时间之和，对周期取模，再判断小明是否需要等待，若需等待返回等待时间，若能直接通过返回0。

运行结果：



结果分析：

T(n) = O(n)

S(n) = O(1)

附源程序。

5、题目：CSP 题目

问题描述：近来，跳一跳这款小游戏风靡全国，受到不少玩家的喜爱。  
　　简化后的跳一跳规则如下：玩家每次从当前方块跳到下一个方块，如果没有跳到下一个方块上则游戏结束。  
　　如果跳到了方块上，但没有跳到方块的中心则获得1分；跳到方块中心时，若上一次的得分为1分或这是本局游戏的第一次跳跃则此次得分为2分，否则此次得分比上一次得分多两分（即连续跳到方块中心时，总得分将+2，+4，+6，+8...）。  
　　现在给出一个人跳一跳的全过程，请你求出他本局游戏的得分（按照题目描述的规则）。

输入格式：

　　输入包含多个数字，用空格分隔，每个数字都是1，2，0之一，1表示此次跳跃跳到了方块上但是没有跳到中心，2表示此次跳跃跳到了方块上并且跳到了方块中心，0表示此次跳跃没有跳到方块上（此时游戏结束）。

输出格式：

输出一个整数，为本局游戏的得分（在本题的规则下）。

算法思想：

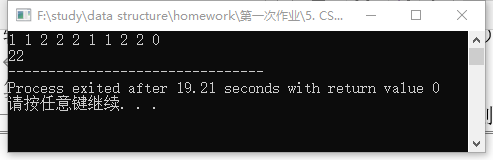
一层循环。利用flag记忆上次跳跃结果，flag为1则上次跳到了中心，flag为0则 上一次的得分为1分或这是本局游戏的第一次跳跃。利用sum累计得分。

若输入为0终止循环。

若输入为1，sum累加2，flag置0。

若输入为2，利用flag判断并累加得分，再令flag置1。

运行结果：



结果分析：

T(n) = O(n)

S(n) = O(1)

附源程序。

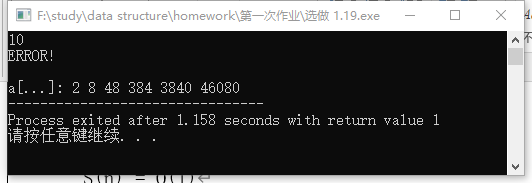
选做题**1.19 试编写算法，计算的值并存入数组a[0..arrsize-1]的第i-1个分量中(i=1,2,…,n)。假设计算机中允许的整数最大值为maxint，则当n>arrsize或对某个，使时，应按出错处理。注意选择你认为较好的出错处理方法。**

算法思想：

先判断n是否大于arrsize，若真，按出错处理。

若n≥1，先令a[0] = 2，然后一层循环，每第i次循环将a[i-1]与i、2累乘，判断是否大于maxint，若大于，出错处理，反之将其存入数组，循环继续。

运行结果：



结果分析：

T(n) = O(n)

S(n) = O(1)

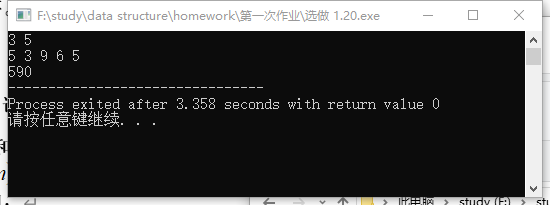
附源程序。

选做题**1.20 试编写算法求一元多项式的值的值，并确定算法中每一语句的执行次数和整个算法的时间复杂度。注意选择你认为较好的输入和输出方法。本题的输入为，和，输出为。**

算法思想：

将尾递归函数理解为一次循环。i遍历0到n-1，第i次循环将第i-1次的结果与x累乘并与下一项系数相加。

运行结果：



结果分析：

T(n) = O(n)

S(n) = O(1)

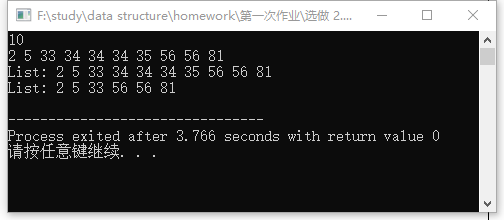
附源程序。

选做题**2.19 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。试写一高效的算法，删除表中所有值大于mink且小于maxk的元素（若表中存在这样的元素），同时释放被删结点空间，并分析你的算法的时间复杂度（注意，mink和maxk是给定的两个参变量，它们的值可以和表中的元素相同，也可以不同）。**

算法思想：

定义两个指针p1、p2，p1从头节点开始寻找最后一个小于等于mink的结点，p2从p1位置开始寻找第一个大于等于maxk的结点，将p1与p2之间的链表从原链表中分离出来逐个释放，并连接原链表。

运行结果：



结果分析：

T(n) = O(n)

S(n) = O(1)

附源程序。