

华东师范大学计算机科学技术系作业

	华东师范大学计算机科学技术系作业	
课程名称：编程导论Python	年级：2018级	作业成绩：
指导教师：兰曼	姓名：魏明达	提交作业日期：2018年10月31日
专业：计算机系	学号：10185102232	作业编号： 4

一、第1章介绍了合并两个有序列表，使得合并后的列表仍然有序的问题，并写出了解决该问题的算法，但是没有给出具体的python代码。

现在将上述问题用函数merge(L1,L2)来实现，该函数的功能为：输入参数是两个从小到大排好序的整数列表L1和L2，返回合成后的从小到大排好序的大列表X。例如，merge([1,4,5],[2,7])会返回[1,2,4,5,7]；merge([], [2,3,4])会返回[2,3,4]。请按下列要求用python实现该函数。

要求：

- (1) 程序中比较两列表元素大小的次数不能超过len(L1)+len(L2)。
 - (2) 只能用列表append()和len()函数。
- (20分)

In [3]:

```
def merge(L1, L2):
    if L1 == []:
        return L2
    elif L2 == []:
        return L1
    else:
        L = []
        i = 0
        j = 0
        while len(L1)>i and len(L2)>j:
            if L1[i] < L2[j]:
                L.append(L1[i])
                i += 1
            else:
                L.append(L2[j])
                j += 1
        if len(L1)<=i:
            for each in range(j, len(L2)):
                L.append(L2[each])
        else:
            for each in range(i, len(L1)):
                L.append(L1[each])
        return L

L1 = [1, 4, 5]
L2 = [2, 7]
print(merge(L1, L2))
```

[1, 2, 4, 5, 7]

二、请改写本章中<程序：整数的十-二进制转换>python程序，完成十进制到二进制的包括小数的转换。输入是一个带小数点的十进制数，输出是一个带有小数点的二进制数，假设精确度是8位。

(20分)

In [1]:

```
def fun1(a):
    x = int(a)
    r = 0
    Rs = []
    while x:
        r = x % 2
        x //= 2
        Rs = [r] + Rs
    ans = ''
    for i in range(len(Rs)):
        ans = ans + str(Rs[i])
    return ans

def fun2(a):
    a = float('0.' + a)
    k = ''
    x = 0.5
    while len(k) <= 8 and a != 0:
        if a >= x:
            k = k + '1'
            a = a - x
        else:
            k = k + '0'
            x /= 2
    return k

b = str(float(input()))
x, y = b.split('.')
x = fun1(x)
y = fun2(y)
print(x+'.'+y)
```

1024.875

10000000000.111

三、黑色星期五（13号是一个星期五）。13号在星期五比在其他日子少吗？为了回答这个问题，写一个程序，要求计算每个月的13号分别为周一到周日的次数。给出N年的一个周期，要求计算1900年1月1日至1900+N-1年12月31日中13号落在周一到周日的次数，N为正整数且不大于400。

输入为：一个数字N。

输出为：7个整数，分别表示13号在周一到周日的次数。

提示：

(1) 1900年1月1日是星期一。

(2) 4、6、9和11月有30天，其他月份除了2月都有31天。闰年2月有29天，平年2月有28天。

(3) 年份可以被4整除的为闰年(1992=4*498 所以 1992年是闰年,但是1990年不是闰年)

(4) 以上规则不适合于世纪年。可以被400整除的世纪年为闰年,否则为平年。所以,1700,1800,1900和2100年是平,而2000是闰年。

(20分)

In [16]:

```
def run(n):
    if (n + 1900) % 4 == 0 and not ((n + 1900) % 400 and (1900 + n) % 100 == 0):
        return 1
    else:
        return 0

def plus(k):
    if 0 <= k <= 5:
        k += 1
    else:
        k = 0
    return k

def num(N, k, L):
    if run(N):
        month = [31, 29, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31]
    else:
        month = [31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31]
    for j in month:
        for i in range(j):
            if i == 12:
                L[k] += 1
            k = plus(k)
    return L, k

N = int(input())
k = 0
L = [0 for i in range(7)]
for i in range(N):
    L, k = num(i, k, L)
print(L)
```

2
[2, 4, 4, 3, 4, 4, 3]

四、挤牛奶，三个农民每天清晨5点起床，然后去牛棚给3头牛挤奶。第一个农民在300秒(从5点开始计时)给他的牛挤奶，一直挤到1000秒。第二个农民在700秒开始，在1200秒结束。第三个农民在1500秒开始，2100秒结束。期间至少有一个农民在挤奶的最长连续时为900秒(从300秒到1200秒)，而无人挤奶的最长连续时间(从挤奶开始一直到挤奶结束)为300秒(从1200秒到1500秒)。

要求编一个Python程序，输入N个农民 ($1 \leq N \leq 5000$) 挤N头牛的工作时间列表，计算以下两个问题 (均以s(秒)为单位)：

- (1) 最长至少有一人在挤奶的时间段。
- (2) 最长的无人挤奶时间段 (从有人挤奶开始算起)。

例如，输入为：[[300,1000],[700,1200],[1500,2100]]，则该输入的每一个元素为一个农民的挤奶时间段。输出：900 300。

(20分)

In [20]:

```
def sor(L):    # sort L in bubble
    n = len(L)
    for j in range(0, n-1):
        for i in range(0, n-1-j):
            if L[i][0] > L[i+1][0]:
                L[i], L[i+1] = L[i+1], L[i]
    return L

def fun(k):    # k is a list
    end = 0
    star = 0
    L = []     # a list to store the continuous time
    for i in range(1, len(k)):
        if k[i][0] < k[end][1] < k[i][1]:
            end = i
        elif k[i][0] > k[end][1]:
            L.append([k[star][0], k[end][1]])
            end = i
            star = i
    L.append([k[star][0], k[end][1]])
    return L

def time_in(L):    # the longest time that have at least one person working
    n = []
    for i in L:
        n.append(i[1]-i[0])
    x = max(n)
    return x

def time_out(L):    # the longest time that has nobody working
    n = []
    for i in range(1, len(L)):
        n.append(L[i][0] - L[i-1][1])
    x = max(n)
    return x

P = [[300, 1000], [700, 1200], [1500, 2100]]
s = fun(sor(P))    # the s has been sorted
print(time_in(s), end=' ')
print(time_out(s))
```

900 300

五、编写一个 Python 程序，求回文平方数。给定一个进制 B ($2 \leq B \leq 10$, 由十进制表示)，输出所有满足以下条件的十进制数 N ，条件：

(1) $1 \leq N \leq 300$ 。

(2) N 的平方用 B 进制表示时是回文数，即回文平方数。

例如，输入为： $K=2$ 。

输出为：

```
1   1   1
3   9  1001
```

输出中，每一行的第一个数为回文平方（十进制表示），第二列为该数的平方（十进制），第三列为平方的 K 进制表示。

(20分)

In [8]:

```
def fun(x, t):
    l = []
    while x:
        r = x % t
        x //= t
        l.append(r)
    ans = ''
    for i in l:
        ans = ans + str(i)
    return ans

B = int(input())
L = []
for n in range(1, 301):
    k = n * n
    p = fun(k, B)
    if p == p[::-1]:
        L.append(n)
for each in L:
    print(each, end=' ')
    print(each**2, end=' ')
    print(fun(each**2, B))
```

```
2
1 1 1
3 9 1001
```