Assignment #7: 20250402 Mock Exam

Updated 1624 GMT+8 Apr 2, 2025

2025 spring, Complied by 同学的姓名、院系

> **说明：**
>
> 1. 月考：AK（请改为同学的通过数）。考试题目都在"题库（包括计概、数算题目）"里面，按照数字题号能找到，可以重新提交。作业中提交自己最满意版本的代码和截图。
> 2. **解题与记录：**
>
>   对于每一个题目，请提供其解题思路（可选），并附上使用Python或C++编写的源代码（确保已在OpenJudge， Codeforces，LeetCode等平台上获得Accepted）。请将这些信息连同显示"Accepted"的截图一起填写到下方的作业模板中。（推荐使用Typora https://typoraio.cn 进行编辑，当然你也可以选择Word。）无论题目是否已通过，请标明每个题目大致花费的时间。
>
> 3. **提交安排：**提交时，请首先上传PDF格式的文件，并将.md或.doc格式的文件作为附件上传至右侧的"作业评论"区。确保你的Canvas账户有一个清晰可见的头像，提交的文件为PDF格式，并且"作业评论"区包含上传的.md或.doc附件。
> 4. **延迟提交：**如果你预计无法在截止日期前提交作业，请提前告知具体原因。这有助于我们了解情况并可能为你提供适当的延期或其他帮助。
>
> 请按照上述指导认真准备和提交作业，以保证顺利完成课程要求。

1. 题目

E05344:最后的最后

http://cs101.openjudge.cn/practice/05344/

思路：

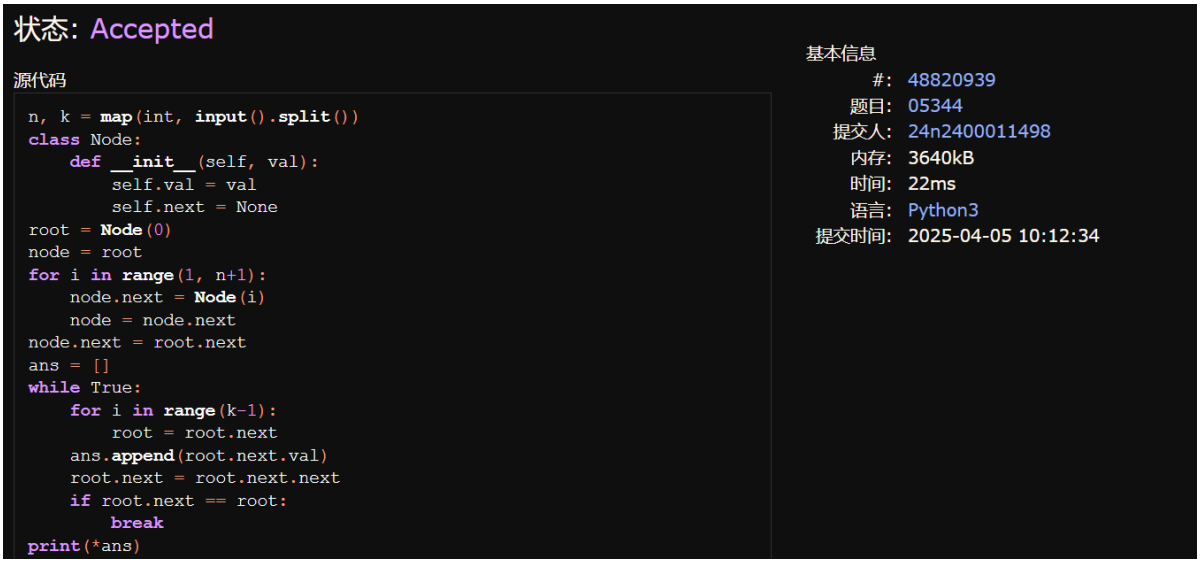
直接模拟实现即可。

代码：

```
n, k = map(int, input().split())
class Node:
    def __init__(self, val):
        self.val = val
        self.next = None
root = Node(0)
node = root
for i in range(1, n+1):
    node.next = Node(i)
    node = node.next
node.next = root.next
ans = []
```

```
while True:
    for i in range(k-1):
        root = root.next
    ans.append(root.next.val)
    root.next = root.next.next
    if root.next == root:
        break
print(*ans)
```

代码运行截图 （至少包含有"Accepted"）



M02774: 木材加工

binary search, http://cs101.openjudge.cn/practice/02774/

思路：

使用二分查找直接实现即可，注意确定好上下界，以及边界条件。

代码：

```
n, k = map(int, input().split())
logs = []
for i in range(n):
    logs.append(int(input()))
def is_ok(a):
    cnt = 0
    for log in logs:
        cnt += log//a
    return cnt >= k
def binary_search():
    low = 1
    high = max(logs)
    while low < high -1:
        mid = (low+high)//2
        if is_ok(mid):
```

```
            low = mid
        else:
            high = mid-1
    if is_ok(high):
        return high
    elif is_ok(low):
        return low
    else:
        return 0
print(binary_search())
```

代码运行截图 （至少包含有"Accepted"）

M07161:森林的带度数层次序列存储

tree, http://cs101.openjudge.cn/practice/07161/

思路：

关键在于如何建树，注意到使用后序会比较清楚。

代码：

```
from collections import deque
n = int(input())
class Node:
    def __init__(self,val):
        self.val = val
        self.children = deque()

def postorder(node):
    order = []
```

```
    for child in node.children:
        order += (postorder(child))
    order.append(node.val)
    return order
ans = []
nodes = deque([])
for _ in range(n):
    s = (list(input().split()))
    while s:
        degrees, val = int(s.pop()), s.pop()
        node = Node(val)
        if degrees != 0:
            for i in range(degrees):
                node.children.appendleft(nodes.pop())
        nodes.appendleft(node)
    root = nodes.pop()
    ans += postorder(root)
print(*ans)
```

代码运行截图 （至少包含有"Accepted"）

M18156:寻找离目标数最近的两数之和

two pointers, http://cs101.openjudge.cn/practice/18156/

思路:

使用双指针可以方便的搞定，注意记录过程中所讨论到的位置的值。

代码:

```
t = int(input())
s = list(map(int, input().split()))
s.sort()
def two_pointers():
    left = 0
    right = len(s)-1
    minv = float('inf')
    maxv = float('inf')
    while left < right:
        temp = s[left]+s[right]
        if temp < t:
            left += 1
            minv = min(minv, t- temp)
        elif temp > t:
            right -= 1
            maxv = min(maxv, temp - t)
        else:
            return t
    if maxv < minv:
        return t+maxv
    else:
        return t-minv
print(two_pointers())
```

代码运行截图 （至少包含有"Accepted"）

M18159:个位为 1 的质数个数

sieve, http://cs101.openjudge.cn/practice/18159/

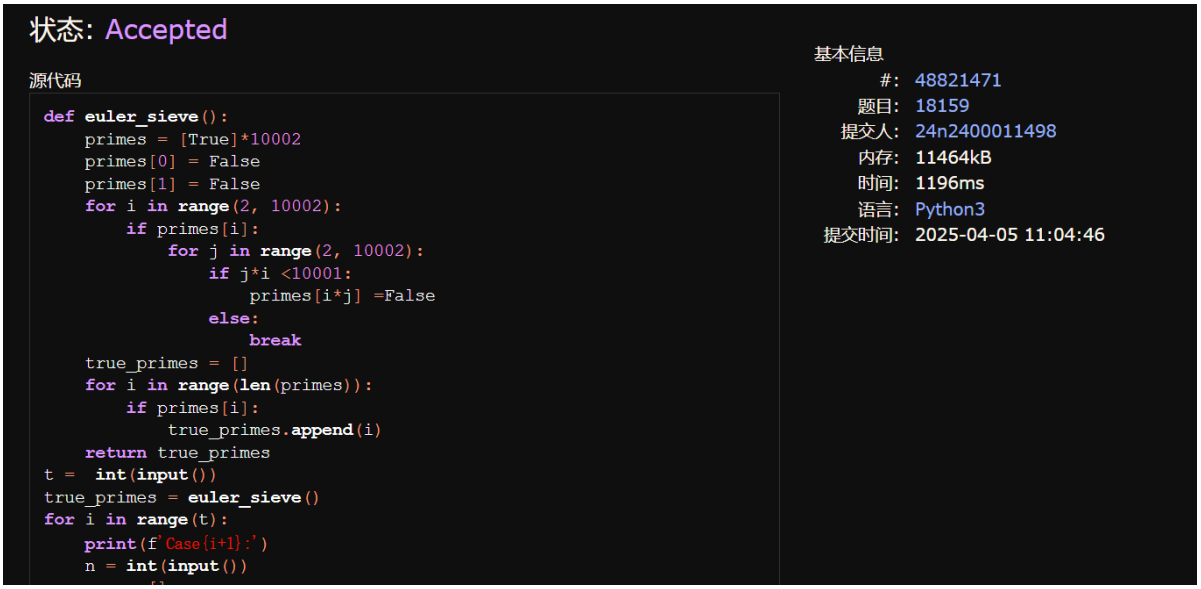
思路：

用筛法可以方便的搞定。

代码：

```
def sieve():
    primes = [True]*10002
    primes[0] = False
    primes[1] = False
    for i in range(2, 10002):
        if primes[i]:
            for j in range(2, 10002):
                if j*i <10001:
                    primes[i*j] =False
                else:
                    break
    true_primes = []
    for i in range(len(primes)):
        if primes[i]:
            true_primes.append(i)
    return true_primes
t =  int(input())
true_primes = sieve()
for i in range(t):
    print(f'Case{i+1}:')
    n = int(input())
    ans = []
    for primes in true_primes:
        if primes < n:
            if primes%10 ==1:
                ans.append(primes)
        else:
            break
    if ans:
        print(*ans)
    else:
        print('NULL')
```

代码运行截图 （至少包含有"Accepted"）



M28127:北大夺冠

hash table, http://cs101.openjudge.cn/practice/28127/

思路：

使用defaultdict直接实现，比较方便。

代码：

```
from collections import defaultdict
AC_questions = defaultdict(set)
submission_times = defaultdict(int)
m = int(input())
for _ in range(m):
    name, question, result = input().split(',')
    if result == 'yes':
        AC_questions[name].add(question)
    submission_times[name] += 1
ranks = []
for name in submission_times:
    ranks.append((len(AC_questions[name]), submission_times[name], name))
ranks.sort(key = lambda x:(-x[0], x[1], x[2]))
for i in range(12):
    if i < len(ranks):
        print(f'{i+1} {ranks[i][2]} {ranks[i][0]} {ranks[i][1]}')
    else:
        break
```

代码运行截图 （AC代码截图，至少包含有"Accepted"）

2. 学习总结和收获

如果发现作业题目相对简单，有否寻找额外的练习题目，如“数算2025spring每日选做”、LeetCode、Codeforces、洛谷等网站上的题目。

这周没有来得及去上机，所以自己课下卡时间做了一下，最终用时90minAK。感觉数算的部分不算很难（？）计概题目反而不太容易一遍过……最后用时最多的还是二分的题目，边界条件一直不太能处理好确实比较麻烦。而且上机的时候要看清楚题目与代码是否对应，这次题目中间有两个题目第一次提交没过，结构修改完代码以后没注意题目是什么，结果提交若干次正确的代码都没能通过，最后发现是题目与代码搞反了，这也非常搞笑。