# Assignment #7: 20250402 Mock Exam

Updated 1624 GMT+8 Apr 2, 2025

2025 spring, Complied by 颜鼎堃 工学院

> **说明：**
>
> 1. 月考：没参加（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：

- 正好上次约瑟夫环是用环形链表做的

代码：

```python
class circleLinkedList:
    def __init__(self, val, next):
        self.val = val
        self.next = next

n, k = map(int, input().split())
head = circleLinkedList(1, None)
p = head
for i in range(2, n + 1):
    p.next = circleLinkedList(i, None)
    p = p.next
p.next = head
for i in range(n - 1):
    for j in range(k - 1):
        head = head.next
        p = p.next
    p.next = head.next
    print(head.val, end=" ")
    head = head.next
```

代码运行截图（至少包含有"Accepted"）


OJ-05344

## M02774: 木材加工

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

思路：

- 二分

代码：

```python
def check(length, n):
    ans = 0
    if length == 0:
        return True
    for i in logs:
        ans += i // length
    return ans >= n

N, K = map(int, input().split())
logs = [int(input()) for i in range(N)]
lo = 0
hi = 10000
mid = 0
while lo < hi:
    mid = (lo + hi) // 2
    if check(mid, K):
        lo = mid + 1
    else:
        hi = mid
print(lo - 1)
```

代码运行截图 （至少包含有"Accepted"）


OJ-02774

## M07161:森林的带度数层次序列存储

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

思路：

- 看起来挺简单的，做起来挺难的，写起来挺简单的

代码：

Python

```python
from collections import deque
ans = []
class treeNode:
    def __init__(self, name, deg):
        self.name = name
        self.deg = deg
        self.val = []

def backTra(tree):
    for node in tree.val:
        backTra(node)
    ans.append(tree.name)

tree_seq = [input().split() for i in range(int(input()))]
for seq in tree_seq:
    node = deque()
    queue = deque()
    queue.append(node)
    for i in range(0, len(seq), 2):
        node.append(treeNode(seq[i], int(seq[i + 1])))
    queue = deque()
    head = node[0]
    queue.append(node.popleft())
    while queue:
        n = queue.popleft()
        for i in range(n.deg):
            n.val.append(node.popleft())
```

```
            queue.append(n.val[-1])
    backTra(head)
print(*ans)
```

代码运行截图 （至少包含有"Accepted"）


OJ-07161

## M18156:寻找离目标数最近的两数之和

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

思路：

- 上学期做过
- $O(n^2)$都不给过

代码：

```python
abs_diff = diff = min_diff = min_abs_diff = 1e9
T = int(input())
S = sorted(map(int, input().split()))
i, j = 0, len(S) - 1
while i < j:
    diff = S[i] + S[j] - T
    abs_diff = abs(diff)
    if (abs_diff, diff) < (min_abs_diff, min_diff):
        min_diff = diff
        min_abs_diff = abs_diff
    else:
        if diff > 0:
            j -= 1
        elif diff < 0:
            i += 1
        else:
            break
print(T + min_diff)
```

代码运行截图 （至少包含有"Accepted"）


OJ-18156

## M18159:个位为 1 的质数个数

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

思路：

- 欧拉筛

代码：

Python

```python
from bisect import bisect_left
prime = []
isprime = [True] * 10001
isprime[0] = isprime[1] = False
for i in range(2, 10001):
    if isprime[i]:
        prime.append(i)
    for j in prime:
        if i * j ⩾ 10001:
            break
        isprime[i * j] = False
        if i % j == 0 and j ≠ i:
            break
prime1 = [p for p in prime if p % 10 == 1]
for i in range(1, 1 + int(input())):
    ans = []
    n = int(input())
    print(f"Case{i}:")
    if n ⩽ 11:
        print("NULL")
    else:
        ind = bisect_left(prime1, n)
        ans = prime1[:ind]
        print(*ans)

```

代码运行截图（至少包含有"Accepted"）


OJ-18159

## M28127:北大夺冠

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

思路：

- 排序
- 读入可以用`defaultdict`

代码：

```Python
from collections import defaultdict
commits = defaultdict(lambda: [set(), 0])
for i in range(int(input())):
    univ, question, status = input().split(",")
    if status == "yes":
        commits[univ][0].add(question)
    commits[univ][1] += 1
rate = sorted(commits, key=lambda t: (-len(commits[t][0]), commits[t][1], t))[:12]
for i in range(len(rate)):
    print(i + 1, rate[i], len(commits[rate[i]][0]), commits[rate[i]][1])
```

代码运行截图（AC代码截图，至少包含有"Accepted"）


OJ-28127

## 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。

这次月考没参加因为当时别的课作业要写不完了

感觉这次难度应该不算特别大，但如果我真在考场上写这个估计也就对个5个吧