



尔真 识理 真释 理尔

协和OJ代码题解析

广州协和学校 信息技术科

2025年5月



广州协和学校



尔 识 真 理
真 理 释 尔

G2024003 A+B问题

- 题意：输出A、B之和。
- 题解：

```
A = int(input())
```

```
B = int(input())
```

```
print(A+B)
```





尔 识 真 理
真 理 释 尔

G2024004 Hello,World!

- 题意：输出Hello,World!
- 题解：直接输出

```
print('Hello,World!')
```

或

```
print("Hello,World!")
```





尔 识 真 理
真 理 释 尔

G2024005 字符菱形2

- 题意：输入字符a，输出a对应的字符菱形
- 题解：一行一行输出，字符串连接用+

```
c = input()
print(' '+c)
print(' '+c+c+c)
print(c+c+c+c+c)
print(' '+c+c+c)
print(' '+c)
```




尔 识 真 理
真 理 释 尔

G2024006 输出A和B的幂、商、余数

- 题意：输出A和B的幂、商、余数
- 题解：运算符分别是、`//`、`%`。

```
A = int(input())
```

```
B = int(input())
```

```
print(A ** B)
```

```
print(A // B)
```

```
print(A % B)
```



尔 识 真 理
真 理 释 尔

G2024007 字符菱形1

- 题意：输出由*构成的菱形
- 题解：直接输出，注意空格

```
print("  *")  
print(" ***")  
print("*****")  
print(" ***")  
print("  *")
```





尔 识 真 理
真 理 释 尔

G2024008 闰年判断

- 题意：输入一个年份，判断是否为闰年
- 题解：使用分支语句，若满足闰年的两个条件之一，便是闰年。

```
n = int(input())
if n % 4 == 0 and n % 100 != 0:
    print("yes")
elif n % 4 == 0 and n % 400 == 0:
    print("yes")
else:
    print("no")
```

简化：

```
year = int(input())
if year % 4 == 0 and year % 100 != 0 or year % 400 == 0:
    print('yes')
else:
    print('no')
```



尔 识 真 理
真 理 释 尔

G2024009 奇偶判断

- 题意：输入一个数，判断其为奇数还是偶数。
- 题解：除以2，余数为1，为奇数；否则为偶数。

```
n = int(input())
```

```
if n % 2 == 1:
```

```
    print("odd")
```

```
else:
```

```
    print("even")
```




尔 识 真 理
真 理 释 尔

G2024010 三个数排序

- 题意：输入a、b、c三个数，从小到大输出。
- 题解：首先找最小的数，在最前面输出，然后剩下的两个数比大小，依次输出。

```
a = int(input())  
b = int(input())  
c = int(input())  
if a > b:  
    a, b = b, a  
if a > c:  
    a, c = c, a  
if b > c:  
    b, c = c, b  
print(a, b, c)
```



尔 识 真 理
真 理 释 尔

G2024011 分段函数

- 题意：输入一个数，输出对应分段函数的值。
- 题解：分支语句判断即可。

```
x = float(input())
```

```
if 0 <= x < 2:
```

```
    print(-x + 2.5)
```

```
elif x < 4:
```

```
    print(2 - 1.5 * (x - 3) * (x - 3))
```

```
else:
```

```
    print(x / 2 - 1.5)
```





尔 识 真 理
真 理 释 尔

G2024012 n个整数求和

- 题意：输入n个数，输出他们的和。
- 题解：依次相加即可。

```
n = int(input())  
total = 0  
for i in range(n):  
    num = int(input())  
    total += num  
print(total)
```

简化：

```
n = int(input())  
total = 0  
for i in range(n):  
    total += int(input())  
print(total)
```



尔 识 真 理
真 理 释 尔

G2024013 找最小值

- 题意：输入n个数，输出其中的最小值。
- 题解：先设定一个较大的数作为当前最小值，遍历列表，假如当前数小于当前最小值，则将更新当前最小值为该数，直到遍历完列表，则可以得到列表最小值。

```
n = int(input())
a = list(map(int, input().split()))
minn = 1005 # 假设最小值为1005
for i in range(n): # 循环n次
    if a[i] < minn: # 如果当前的数小于当前最小值
        minn = a[i] # 那么将其设置为最小值
print(minn)
```

```
# 使用Python内置的排序函数进行排序，
然后输出第一个数，即为最小值。
n = int(input())
a = list(map(int, input().split()))
a.sort()
print(a[0])
```




尔 识 真 理
真 理 释 尔

G2024014 从小到大输出n的因子

- 题意：输入一个数，从小到大输出它的因子。
- 题解：从1遍历到n，如果能整除，则输出。

```
n = int(input())  
for x in range(1, n + 1):  
    if n % x == 0:  
        print(x)
```



尔 识 真 理
真 理 释 尔

G2024015 从大到小输出n的因子

- 题意：输入一个数，从小到大输出它的因子。
- 题解：从n遍历到1，如果能整除，则输出。

```
n = int(input())  
for x in range(n, 0, -1):  
    if n % x == 0:  
        print(x)
```



尔 识 真 理
真 理 释 尔

G2024016 一尺之棰

- 题意：给定初始木棍长度 a ，每天切一半（向下取整），求第几天长度变为1。
- 题解：初始化天数为1，若当前长度不为1，循环将其除以2并增加天数，直至长度为1时停止。

```
n = int(input())  
ans = 1  
while n > 1:  
    ans += 1  
    n //= 2  
print(ans)
```



尔 识 真 理
真 理 释 尔

G2024017 斐波那契数列第k项

- 题意：输出斐波那契数列第k项。
- 题解：
 - 如果 k 是 1 或 2，直接输出 1，因为斐波那契数列的前两项都是 1。
 - 如果 k 大于 2，初始化变量 a1 和 a2 为 1，分别表示斐波那契数列的前两项 F(1) 和 F(2)。
 - 循环从第 3 项开始计算，一直到第 k 项。每次迭代时：a1 更新为 a2 的值（即前一项）。a2 更新为 a1 + a2 的值（即新的当前项）。这样每次循环 a2 就是当前的斐波那契数。
 - 最终循环结束后，a2 就是第 k 项的值，输出它。

```
k = int(input())
if k == 1 or k == 2:
    print(1)
else:
    a1 = a2 = 1
    for i in range(k - 2):
        a1, a2 = a2, a1 + a2
    print(a2)
```




尔 识 真 理
真 理 释 尔

G2024018 计算总成绩

- 题意：给定三个非负整数A、B、C，计算其加权和（ $A \times 20\% + B \times 30\% + C \times 50\%$ ），输出整数结果。
- 题解：将三项分别乘以权重系数后求和

```
a = int(input())  
b = int(input())  
c = int(input())  
d = a * 0.2 + b * 0.3 + c * 0.5  
print(int(d))
```



尔 识 真 理
真 理 释 尔

G2024019 肥胖问题

• 题意

- 输入体重 m 和身高 h ，计算 BMI 值，并根据范围输出对应结果：`Underweight`、`Normal` 或 `Overweight`（需同时输出 BMI 值）。

• 题解

1. 读取两个浮点数 m 和 h 。
2. 计算 BMI。
3. 根据条件判断：
 - 若 $BMI < 18.5$ ，输出 `Underweight`；
 - 若 $18.5 \leq BMI < 24$ ，输出 `Normal`；
 - 否则输出 BMI 值和 `Overweight`。

```
m = float(input())
h = float(input())
bmi = m / (h * h)
if bmi < 18.5:
    print("Underweight")
elif bmi < 24:
    print("Normal")
else:
    print(bmi)
    print("Overweight")
```



尔 识 真 理
真 理 释 尔

G2024020 极差计算

• 题意

- 给定一组整数，求其极差（最大值与最小值的差）。

• 题解

1. 读取整数个数 n 和列表 a 。
2. 遍历列表，找出最大值和最小值（或使用函数 $\max$ 和 $\min$ ）。
3. 输出两者的差值。

```
# 复制该段代码到IDLE，并补充
# 读取输入的数字个数
n = int(input())
# 读取n个非负整数并转为列表a
a = list(map(int, input().split()))
# 在下面补充你的代码
maxn = -1
minn = 1001
for i in range(n):
    maxn = max(a[i], maxn)
    minn = min(a[i], minn)
print(maxn - minn)
```

简化：

```
n = int(input())
a = list(map(int, input().split()))
max_val = max(a)
min_val = min(a)
print(max_val - min_val)
```



尔 识 真 理
真 理 释 尔

G2024021 小苹果

• 题意

- 每天从左起每隔两个苹果拿一个，求拿完所有苹果的天数及编号为 n 的苹果被拿走的天数。

• 题解

1. 总天数：循环计算每天拿走 $\left\lceil \frac{n}{3} \right\rceil$ 个苹果，直到 n 归零，统计循环次数。
2. 特定苹果天数：当剩余苹果数 $n \% 3 == 1$ 时，当天必拿走最后一个苹果（即编号 n 的苹果），记录此时的天数。

```
import math
```

```
n = int(input())
```

```
res1 = 0 # 记录操作次数
```

```
res2 = 0 # 记录第一次  $n \% 3 == 1$  时的操作序号
```

```
# 循环计算
```

```
i = 1
```

```
while n > 0:
```

```
    if res2 == 0 and  $n \% 3 == 1$ :
```

```
        res2 = i # 记录第一次  $n \% 3 == 1$  的操作序号
```

```
    n -= math.ceil(n / 3.0)
```

```
    res1 += 1
```

```
    i += 1
```

```
print(res1)
```

```
print(res2)
```




尔 识 真 理
真 理 释 尔

G2024022 求阶乘

• 题意

- 输入一个正整数 n ，计算并输出其阶乘 $n!$ 的值。

• 题解

1. 输入处理：读取正整数 n 。
2. 计算阶乘：初始化结果为1，通过循环从1到 n 逐项相乘。
3. 输出结果：打印最终结果。

```
n = int(input())  
sum = 1  
for i in range(1, n + 1):  
    sum *= i  
print(sum)
```



尔 识 真 理
真 理 释 尔

G2024023 求阶乘之和

- 题意

- 输入正整数 n ，计算 $1! + 2! + 3! + \dots + n!$ 的和

- 题解

1. 初始化：总和 $\text{sum}=0$ ，当前阶乘 $f=1$ （对应 $1!$ ）。
2. 循环迭代：从 $i=1$ 到 n ，依次将 f 累加到 sum ，并更新 f （计算下一个阶乘）。
3. 返回结果：最终输出总和 sum 。

```
n = int(input())
sum = 0
for i in range(1, n + 1):
    f = 1
    for j in range(1, i + 1):
        f *= j
    sum += f
print(sum)
```



尔 识 真 理
真 理 释 尔

G2024024 九九乘法表

```
for i in range(1, 10):  
    for j in range(1, i + 1):  
        print("{0:1}*{1:1}={2:2} ".format(j, i, j * i), end="")  
    print()
```

• 题意

- 打印九九乘法表，要求每行输出对应行数的乘法算式

• 题解

1. 外层循环：for i in range(1, 10)控制行数，共 9 行。
2. 内层循环：for j in range(1, i + 1) 控制每行的列数，第 i 行输出 i 个算式。
3. 格式化输出：
 - {0:1}*{1:1}={2:2} 确保数字和符号对齐，其中 :1 和 :2 分别指定宽度。
 - end="" 避免换行，使算式在同一行连续输出。
4. 换行处理：每行末尾单独 print() 实现换行。



尔 识 真 理
真 理 释 尔

G2024025 级数求和

• 题意

- 输入一个整数 k ，求最小的正整数 n ，使得调和级数 $1 + \frac{1}{2} + \frac{1}{3} + \dots + \frac{1}{n}$ 的和首次超过 k 。

• 题解

1. 初始化: $sum = 0$ 记录当前和, $i = 1$ 表示当前项的分母。
2. 循环累加: 每次将 $\frac{1}{i}$ 加入 sum , 若 $sum > k$ 则终止循环, 否则 $i += 1$ 。
3. 输出结果: 最终的 i 即为满足条件的最小 n 。

```
k = int(input())
sum = 0
i = 1
while True:
    sum += 1 / i
    if sum > k:
        break
    i += 1
print(i)
```




尔 识 真 理
真 理 释 尔

G2024026 打印星号矩阵

• 题意

- 输入两个正整数 n 和 m ，输出一个 n 行 m 列的星号矩阵，每行的星号之间用空格分隔。

• 题解

1. 输入处理：读取整数 n 和 m 。
2. 双重循环：
 - 外层循环 `for i in range(n)` 控制行数，共 n 次。
 - 内层循环 `for j in range(m)` 控制每行的星号数量，共 m 次。
3. 格式化输出：
 - 每次打印 `*` 时指定 `end=" "`，使星号后接空格而非换行。
 - 内层循环结束后，单独执行 `print()` 实现换行。

```
n = int(input())
m = int(input())
for i in range(n):
    for j in range(m):
        print("*", end=" ")
    print()
```



尔 识 真 理
真 理 释 尔

G2024028 圆的面积

• 题意

- 输入一个整数 r ，计算以 r 为半径的圆的面积 (π 取 3.14)，并输出结果。

• 题解

- 1. 输入处理：读取整数 r 。
- 2. 面积计算：使用公式 $S = 3.14 \times r^2$ 计算圆的面积。
- 3. 输出结果：直接打印计算得到的面积值。

```
r = int(input())  
S = 3.14 * r * r  
print(S)
```



尔 识 真 理
真 理 释 尔

G2024029 计算 $f(x)$

• 题意

- 输入一个整数 x ，输出其绝对值 $|x|$ 。

• 题解

1. 输入处理：读取整数 x 。
2. 条件判断：
 - 若 $x > 0$ ，直接输出 x 。
 - 若 $x < 0$ ，输出 $-x$ （即取相反数）。

```
x = int(input())  
if x >= 0:  
    print(x)  
else:  
    print(-x)
```



尔 识 真 理
真 理 释 尔

G2024030 输出变量B的值

- 题意
 - 根据流程图写代码

- 题解

$n = 1$

$A = 1$

$B = 2$

while $n \leq 3$:

$A = A + B$

$B = A + B$

$n = n + 1$

print(B)





尔 识 真 理
真 理 释 尔

G2024031 生肖计算

• 题意

- 输入若干个年份（正整数），当输入0时结束。对每个非零年份，输出其对应的生肖（以1900年为基准，1900年为鼠年，按顺序循环）。

• 题解

- 1. 生肖周期：生肖每12年循环一次，因此年份与1900的差值对12取余，可得到对应索引。
- 2. 索引计算： $(year - 1900) \% 12$ ，结果为0~11，分别对应生肖列表["鼠", "牛", ..., "猪"]的位置。

```
def shengxiao(year):
```

```
    # 求生肖
```

```
    zodiac = ["鼠", "牛", "虎", "兔", "龙", "蛇", "马", "羊", "猴",  
              "鸡", "狗", "猪"]
```

```
    # 后续代码补充在下面
```

```
    zodiac_index = (year - 1900) % 12
```

```
    return zodiac[zodiac_index]
```

```
# 以下为主程序
```

```
while True:
```

```
    myyear = int(input())
```

```
    if myyear == 0:
```

```
        break
```

```
    print(shengxiao(myyear))
```



尔 识 真 理
真 理 释 尔

G2024032 火柴棒数计算1

• 题意

- 输入一个整数 $snum$ ，表示可用火柴根数。输出 1 到 100 之间所有能用 $snum$ 根火柴组成的数字（每个数字的火柴数由列表 $f = [6, 2, 5, 5, 4, 5, 6, 3, 7, 6]$ 定义，如数字 0 需 6 根，1 需 2 根）。

• 题解

- 1. 火柴映射：预定义列表 f 表示数字 0-9 的火柴数。
- 2. 火柴计算：对每个 i ，将其拆分为各位数字，累加对应火柴数（如 $12 \rightarrow 2+5=7$ ）。
- 3. 条件筛选：若总火柴数等于 $snum$ ，则输出该数字 i 。

```
def match_num(num):  
    f = [6, 2, 5, 5, 4, 5, 6, 3, 7, 6]  
    if num == 0:  
        total = f[0]  
    else:  
        total = 0  
        while num > 0:  
            x = num % 10  
            total = total + f[x]  
            num = num // 10  
        return total
```

```
snum = int(input())  
for i in range(1, 101):  
    if match_num(i) == snum:  
        print(i)
```



尔 识 真 理
真 理 释 尔

G2024033 火柴棒数计算2

• 题意

- 输入火柴数 $snum$ ，统计 1 到 100 中数字的各位火柴数之和等于 $snum$ 的个数。

• 题解

- 1. 火柴映射：定义列表 f 表示 0-9 的火柴数。
- 2. 拆分计算：对每个数拆分为各位数字，累加对应火柴数（如 $12 \rightarrow 2+5=7$ ）。
- 3. 统计匹配：遍历 1-100，统计火柴数等于 $snum$ 的数字个数。

```
def match_num(num):  
    f = [6, 2, 5, 5, 4, 5, 6, 3, 7, 6]  
    if num == 0:  
        total = f[0]  
    else:  
        total = 0  
        while num > 0:  
            x = num % 10  
            total = total + f[x]  
            num = num // 10  
        return total
```

```
snum = int(input())  
ans = 0  
for i in range(1, 101):  
    if match_num(i) == snum:  
        ans += 1  
print(ans)
```




尔 识 真 理
真 理 释 尔

G2024034 十进制转二进制

• 题意

- 输入十进制，输出对应的二进制。

• 题解

1. 核心逻辑：

- 初始化空字符串binary_result存储二进制结果。
- 通过循环对 n 取余 ($n \% 2$)，将余数（当前最低有效位）插入到结果字符串的最前面，然后将 n 整除 2 ($n //= 2$)，逐步提取更高位。
- 当 n 变为 0 时终止循环，返回最终的二进制字符串。

```
def decimal_to_binary_manual(n):  
    binary_result = ""  
    while n > 0:  
        binary_result = str(n % 2) + binary_result  
        n //= 2  
    return binary_result
```

```
n = int(input())  
ans = decimal_to_binary_manual(n)  
print(ans)
```




尔 识 真 理
真 理 释 尔

G2024035 鸡兔同笼

- 题意

- 输入鸡兔总数 x 和总脚数 y , 输出鸡和兔的数量。保证至少存在一个合法解。

- 题解

- 使用双重循环遍历鸡和兔的可能数量 (均在 0 - 100 范围内)。
- 对每一对 (chicken, rabbit), 检查是否同时满足 $\text{chicken} + \text{rabbit} = x$ 且 $2 * \text{chicken} + 4 * \text{rabbit} = y$ 。
- 一旦找到符合条件的组合, 立即输出结果。

```
x = int(input())
```

```
y = int(input())
```

```
for chicken in range(0, 101):
```

```
    for rabbit in range(0, 101):
```

```
        if chicken + rabbit == x and chicken * 2 + rabbit * 4 == y:
```

```
            print(chicken)
```

```
            print(rabbit)
```



尔 识 真 理
真 理 释 尔

G2024036 真假判断

```
champion = ["A", "B", "C", "D"] # 设置选手列表
for i in champion: # 列举出所有可能的冠军
    cond = (i != "A") + (i == "C") + (i == "D") + (i != "D") # 统计说真话的人数
    if cond == 3: # 判断说真话的人数是否为3
        print(i)
```

• 题意

- 四位选手A、B、C、D中有一人是冠军。给出四个条件：1. 冠军不是A (`i != "A"`) 2. 冠军是C (`i == "C"`) 3. 冠军是D (`i == "D"`) 4. 冠军不是D (`i != "D"`)
- 要求找出唯一满足恰好三个条件为真的冠军。

• 题解

- 逐一验证每个选手是否符合：

- A: 条件1 (假)、条件2 (假)、条件3 (假)、条件4 (真) → 总真数=1
- B: 条件1 (真)、条件2 (假)、条件3 (假)、条件4 (真) → 总真数=2
- C: 条件1 (真)、条件2 (真)、条件3 (假)、条件4 (真) → 总真数=3
- D: 条件1 (真)、条件2 (假)、条件3 (真)、条件4 (假) → 总真数=2



- 只有C满足恰好三个条件为真，因此答案是C。



尔 识 真 理
真 理 释 尔

G2024037 取糖果

- 题意

- Alice和Bob轮流取糖果，每次可取1或2个，取光者胜。两
人均采取最优策略，判断谁会赢。

- 题解

- 1. 胜负规律：当糖果数 n 是3的倍数时，Bob必胜；否则
Alice必胜。
- 2. 策略分析：
 - - 若 $n \% 3 == 0$ ，Bob可通过每次与Alice取数和为3来控
制胜利。
 - - 否则，Alice先取 $n \% 3$ 个，使剩余数量为3的倍数，后
续策略同上，确保自己必胜。

```
n = int(input())  
if n % 3 == 0:  
    print("Bob")  
else:  
    print("Alice")
```




尔 识 真 理
真 理 释 尔

G2024038 水仙花数

• 题意

- 找出所有三位数中的水仙花数（各位立方和等于自身的数）。

• 题解

- 遍历100-999的数，分解个十百位，计算立方和。若等于原数则输出。例如
 $153=1^3+5^3+3^3$ 。

```
for n in range(100, 1000):  
    i = n % 10  
    j = n // 10 % 10  
    k = n // 100  
    if i3 + j3 + k3 == n:  
        print(n)
```




尔 识 真 理
真 理 释 尔

G2024039 递归法求斐波那契数列第k项

• 题意

- 输入正整数k，输出斐波那契数列第k项 ($k \geq 1$)。

• 题解

递归计算：

- 终止条件：k=1或k=2时返回1
- 递推公式： $f(k) = f(k-1) + f(k-2)$

```
def f(n):  
    if n == 1 or n == 2:  
        return 1  
    return f(n - 1) + f(n - 2)
```

```
k = int(input())  
print(f(k))
```



尔 识 真 理
真 理 释 尔

G2024040 递归法求k的阶乘

• 题意

- 输入一个正整数 k ，使用递归方法计算并输出 $k!$ 的值。

• 题解

- 基准条件：当 $n=1$ 时返回 1（即 $1! = 1$ ）。
- 递归步骤：对于 $n > 1$ ，返回 $f(n-1)*n$ ，即通过递归调用计算 $(n-1)!$ 并乘以 n 。

递归法求阶乘

```
def f(n):  
    # 请在下面继续补充代码  
    if n == 1:  
        return 1  
    else:  
        return f(n - 1) * n
```

```
k = int(input())  
print(f(k))
```



尔 识 真 理
真 理 释 尔

G2024041 汉诺塔问题

• 题意

- 输入正整数 n ，输出将汉诺塔问题中 n 个圆盘从柱子 A 移动到柱子 C 的完整步骤，每步格式为 X To Y。

• 题解

递归思想：

- 将 $n-1$ 个圆盘从 A 移动到辅助柱 B（借助 C）。
- 将第 n 个圆盘从 A 直接移动到目标柱 C。
- 将 $n-1$ 个圆盘从 B 移动到 C（借助 A）。

边界处理：当 $n=1$ 时直接输出移动步骤，递归终止。

```
def hanoi(n, s, m, t):  
    if n == 1:  
        print(s + " To " + t)  
    else:  
        hanoi(n - 1, s, t, m)  
        print(s + " To " + t)  
        hanoi(n - 1, m, s, t)
```

```
n = int(input())  
hanoi(n, "A", "B", "C")
```



尔 识 真 理
真 理 释 尔

G2024042 青蛙跳台阶

• 题意

- 青蛙跳台阶问题：一只青蛙每次可以跳1级或2级台阶，输入正整数 n 表示台阶数，输出青蛙跳上 n 级台阶的所有方法数。

• 题解

1. 数学模型：

- 设 $f(n)$ 为跳上 n 级台阶的方法数。
- 递推关系： $f(n) = f(n-1) + f(n-2)$ （最后一步跳1级或2级）。
- 初始条件： $f(1) = 1$ （仅1种跳法） $f(2) = 2$ （1+1 或 直接跳2级）

2. 递归逻辑：

- 每次调用分解为计算前两项的值，直到退化到初始条件。
- 例如： $f(3) = f(2) + f(1) = 2 + 1 = 3$ $f(4) = f(3) + f(2) = 3 + 2 = 5$

```
def f(n):  
    # 基础情况  
    if n == 1:  
        return 1  
    elif n == 2:  
        return 2  
    # 递归调用  
    return f(n - 1) + f(n - 2)  
  
# 输入台阶数  
n = int(input())  
# 输出跳法数  
print(f(n))
```




尔 识 真 理
真 理 释 尔

G2024043 汉诺塔的移动次数

• 题意

- 输入正整数 n ，表示汉诺塔中盘子的数量。输出将所有盘子从起始柱移动到目标柱所需的最少移动次数。

• 题解

1. 递归逻辑：

- 基准条件：当 $n=1$ 时，直接移动唯一盘子，需 1 次。
- 递推关系：
 - 移动 n 个盘子的总次数 =
 - 将 $n-1$ 个盘子从起始柱移动到辅助柱（需 $f(n-1)$ 次）
 - 将第 n 个盘子从起始柱移动到目标柱（需 1 次）
 - 将 $n-1$ 个盘子从辅助柱移动到目标柱（需 $f(n-1)$ 次）
 - 即： $f(n) = 2 * f(n-1) + 1$ 。

2. 数学结论：

- 该递归公式等价于 $f(n) = 2^n - 1$ ，即每次递归分解后移动次数呈指数增长。

```
def f(n):  
    if n == 1:  
        return 1  
    else:  
        return 2 * f(n - 1) + 1  
  
n = int(input())  
print(f(n))
```



尔 识 真 理
真 理 释 尔

G2024044 阿克曼 (Ackermann) 函数

• 题意

- 输入非负整数 a 和 b ，计算阿克曼函数 $Ack(a, b)$ 的值。

• 题解

- 1. 递归结构：
 - - 基准条件：当 $a=0$ 时直接返回 $b+1$ 。
 - - 递归边界：当 $b=0$ 时，递归调用 $Ack(a-1, 1)$ 。
 - - 嵌套递归：其他情况需先计算内层 $Ack(a, b-1)$ ，再代入外层 $Ack(a-1, .)$ 。

```
def ack(a, b):  
    if a == 0:  
        return b + 1  
    if b == 0:  
        return ack(a - 1, 1) # 递归边界  
    return ack(a - 1, ack(a, b - 1)) # 函数递归调用
```

```
a, b = map(int, input().split())  
print(ack(a, b))
```



尔 识 真 理
真 理 释 尔

G2024045 Hermite 多项式

• 题意

- 输入非负整数 n 和整数 x ,
输出第 n 阶 Hermite 多项式在 x 处的值。

• 题解

- 1. 递归实现:
 - - 基准条件: 直接返回 $H_0(x) = 1$ 和 $H_1(x) = 2x$ 。
 - - 递推关系: 对 $n > 2$,
通过递归计算当前值。

```
def Hermite(n, x): # 递归求解
```

```
    if n == 0:
```

```
        return 1
```

```
    if n == 1:
```

```
        return 2 * x
```

```
    else:
```

```
        return 2 * x * Hermite(n - 1, x) - 2 * (n - 1) * Hermite(n - 2, x)
```

```
n, x = map(int, input().split()) # 输入 n 和 x
```

```
print(Hermite(n, x)) # 输出结果
```



尔 识 真 理
真 理 释 尔

G2024047 G2024期末代码题

• 题意

- 输入第 n 次落地，求小球总路径长度（初始5米，每次反弹一半高度）。

• 题解

- 循环 n 次，首次下落5米，后续每次累加反弹+下落（前次高度的一半），迭代计算总距离。

```
n = int(input())  
h = 5  
s = 5  
for i in range(2, n + 1):  
    s = s + h  
    h = h / 2  
print(s)
```




尔 识 真 理
真 理 释 尔

G2024048 G2024期末选做题

• 题意

- 输入正整数 n ，输出 2 到 n 之间的所有质数（每行一个）。

• 题解

- 1. 质数判断：对每个数 $i \in [2, n]$ ，检查是否存在 $2 \leq j < i$ 能整除 i 。若无，则为质数。
- 2. 遍历输出：逐个验证 2 到 n 的数，符合条件的直接打印。

```
def is_prime(num):  
    for i in range(2, num):  
        if num % i == 0:  
            return False  
    return True
```

```
n = int(input())  
for i in range(2, n + 1):  
    if is_prime(i):  
        print(i)
```



尔 识 真 理
真 理 释 尔

G2024052 十进制转二进制（带前缀）

- 题意

- 输入一个整数 a ，输出其对应的二进制字符串（以 `0b` 开头）。

- 题解

- 1. 输入处理：读取整数 a 。
- 2. 二进制转换：使用内置函数 `bin(a)` 将十进制数转换为二进制字符串。
- 3. 输出结果：直接打印转换后的字符串（含 `0b` 前缀）。

```
a = int(input())  
b = bin(a)  
print(b)
```



尔 识 真 理
真 理 释 尔

G2024053 十进制转二进制（无前缀）

- 题意

- 输入一个非负整数 n ，输出其二进制表示（不含前缀 `0b`）。

- 题解

- 1. 转换二进制：使用 `bin(a)` 将整数转换为 `0b...` 形式的二进制字符串。
- 2. 去除前缀：通过切片 `b[2:]` 去除前两位（即 `0b`），直接输出纯二进制数字。

```
a = int(input())  
b = bin(a)  
print(b[2:])
```



尔 识 真 理
真 理 释 尔

G2024057 队列模拟订单处理

• 题意

- 模拟订单队列操作：1 添加订单，2发货（先入先出），3查询订单列表，4退出。

• 题解

- 使用列表实现队列功能。
- - `append()` 添加订单至队尾。
- - `pop(0)` 按先入先出原则处理订单。

```
listque = [] # 定义列表listque存储订单
```

```
x = 0
```

```
while x != 4: # 当x!=4时，执行循环
```

```
    x = int(input()) # 输入选择项
```

```
    if x == 1:
```

```
        y = input() # 输入订单编号
```

```
        listque.append(y) # 在列表listque中添加订单号
```

```
    elif x == 2:
```

```
        if len(listque) == 0:
```

```
            print("订单列表为空")
```

```
        else:
```

```
            print(listque.pop(0)) # 删除列表listque的首元素，表示发货
```

```
    elif x == 3:
```

```
        print(listque) # 查询列表中listque中的订单号
```




尔 识 真 理
真 理 释 尔

G2024062 修改凯撒加密程序

• 题意

- 输入一段明文，将每个字母向后移动5位（如a→f），x/y/z绕回a/b/c（如x→c），非字母字符保持不变，输出密文。

• 题解

- 1. 字母转换：
 - - 对a-u/A-U直接`+5`（ASCII码转换）。
 - - 对v-z/V-Z需`+5-26`实现绕回（如x→c）。
- 2. 非字母处理：直接拼接原字符。



尔 识 真 理
真 理 释 尔

G2024062 修改凯撒加密程序

• 代码

```
1  # 凯撒密码加密程序（字母向后移动3位，x/y/z绕回a/b/c）
2
3  # 输入明文
4  c = input("")
5  b = "" # 初始化空字符串用于存储密文
6
7  for i in range(len(c)):
8      # 处理 a-w 或 A-W 的字母：直接后移3位
9      if "a" <= c[i] <= "u" or "A" <= c[i] <= "U":
10         # ord()获取ASCII码，+3后转回字符
11         b += chr(ord(c[i]) + 3) # 等效于 b = b + ...
12
13     # 处理 x-z 或 X-Z 的字母：后移3位后绕回开头
14     elif "v" <= c[i] <= "z" or "V" <= c[i] <= "Z":
15         # 例如：x(120) -> 123-26=97 -> a
16         b += chr(ord(c[i]) + 3 - 26)
17
18     # 处理非字母字符（数字、符号等）：保持原样
19     else:
20         b += c[i]
21
22 # 输出密文
23 print(b)
```



尔 识 真 理
真 理 释 尔

G2024063 修改凯撒解密程序（选做）

- 题意

- 将凯撒密码密文（字母后移3位）还原为明文。非字母字符保持不变。例如，密文'd'还原为'a'，密文'a'还原为'x'。

- 题解

- 1. 字母转换：
 - - 对'd'~'z'或'D'~'Z'的字符直接减3（如'd'→'a'）。
 - - 对'a'~'c'或'A'~'C'的字符加23（如'a'→'x'）。
- 2. 非字母处理：直接拼接原字符。
- 3. 正确性：通过反向位移（-3）和绕回（+23）正确还原原文。



尔 识 真 理
真 理 释 尔

G2024063 修改凯撒解密程序（选做）

• 代码

```
1  # 凯撒密码加密程序（字母向后移动3位，x/y/z绕回a/b/c）
2  # 现在需要你改成解密程序
3
4  c = input()
5  b = ""
6
7  for i in range(len(c)):
8      if "d" <= c[i] <= "z" or "D" <= c[i] <= "Z":
9          b += chr(ord(c[i]) - 3)
10
11     elif "a" <= c[i] <= "c" or "A" <= c[i] <= "C":
12         b += chr(ord(c[i]) + 23)
13
14     else:
15         b += c[i]
16
17  print(b)
18
```




尔 识 真 理
真 理 释 尔

G2024066 2025信息技术学考模拟考1 (代码题)

• 题意

- 输入一个正整数 n 和 n 个整数，输出其中所有的偶数。

• 题解

1. 输入处理：读取整数 n 和列表 a 。
2. 筛选偶数：遍历列表，对每个元素 x 判断是否为偶数 ($x \% 2 == 0$) 。
3. 输出结果：每找到一个偶数，立即打印。

```
# 复制该段代码到IDLE，并补充
# 读取输入的数字个数
n = int(input())
# 读取n个整数并转为列表a
a = list(map(int, input().split()))
# 在下面补充你的代码
for x in a:
    if x % 2 == 0:
        print(x)
```