**2024转计算机学院机考试题（回忆版）——Python**

2024年转专业机考，我是生死未卜。今年与以往两年有所不同，题目类型变了，是蓝桥杯里的题目了。

1.

10~1000内满足条件的回文整数。要求如下

1. 这个整数本身是回文数，假设该整数为i，则i的平方，i的立方也均为回文数。
2. 逐行输出这些符合条件的数字，并把其对应的平方，立方在同一行输出。

输出结果展示：

11 121 1331

101 10201 1030301

111 12321 1367631

def is\_palindrome(num\_str):

    return num\_str == num\_str[::-1]

for i in range(10, 1000):

    if all(is\_palindrome(str(x)) for x in (i, i\*\*2, i\*\*3)):

        print(f"{i}, {i\*\*2}, {i\*\*3}")

2.

买礼物，输入礼物的价格(升序空格隔开)，挑选其中三件，输出有多少种购买方式(组合)

例如:

输入 1 2 3 4 5

输出 10

def combinations\_count(n):

    #计算组合数C(n,3)

    if n < 3:

        return 0

    if n == 3:

        return 1

    result = 1

    for i in range(1, 4):

        result \*= (n - i + 1)

        result //= i

    return result

gift\_prices = list(map(int,input().split()))

num = len(gift\_prices)

print(combinations\_count(num))

3.

小蓝最近学习了一些排序算法，其中冒泡排序让他印象深刻。在冒泡排序中，每次只能交换相邻的两个元素。小蓝发现，如果对一个字符串中的字符排序，只允许交换相邻的两个字符，则在所有可能的排序方案中，冒泡排序的总交换次数是最少的。

例如，对于字符串 lan 排序，只需要 1次交换。对于字符串 qiao 排序，总共需要 4 次交换。小蓝的幸运数字是 V，他想找到一个只包含小写英文字母的字符串，对这个串中的字符进行冒泡排序，正好需要 V 次交换。

请帮助小蓝找一个这样的字符串。如果可能找到多个，请告诉小蓝最短的那个。如果最短的仍然有多个，请告诉小蓝字典序最小的那个。请注意字符串中可以包含相同的字符。

输入格式

输入一行包含一个整数"V" ，为小蓝的幸运数字。

输出格式

输出一个字符串，为所求的答案。

样例输入

4

样例输出

bbaa

样例输入

100

样例输出

Jihgfeeddccbbaa

def length(v):

    i = 1

    while i \* (i - 1) // 2 < v:

        i += 1

    return i

def create\_string(length):

    goal\_string = ""

    alpha = 'a'

    for \_ in range(length):

        goal\_string += alpha

        alpha = chr((ord(alpha) - ord('a') + 1) % 26 + ord('a'))

    return goal\_string

def add(add\_number, demo\_string, length):

    list\_demo\_string = list(demo\_string)

    alpha = 'a'

    index = 0

    while add\_number > 0:

        # 由于index从0开始，所以插入位置应该对应demo\_string的索引+1

        list\_demo\_string.insert(index + 1, alpha)

        index += 2

        alpha = chr((ord(alpha) - ord('a') + 1) % 26 + ord('a'))

        add\_number -= 1

    return ''.join(sorted(list\_demo\_string,reverse=True))

V = int(input())

len\_needed = length(V)

add\_chars = len\_needed \* (len\_needed - 1) // 2 - V

demo\_string = create\_string(len\_needed - add\_chars)

final\_string = add(add\_chars, demo\_string, len\_needed)

print(final\_string)

鼠鼠真的是太菜了！当我看到这道题目时候，写出来冒泡排序后,就没有任何思路了5555....考完试重新做，加上参考大佬的做法，发现好像...主要还是自己题目理解有问题。没搞清楚就一头雾水的扎进去了，当然...这也与备考题目类型差异过大有关（不是对自己菜的辩白），我想机考是一个应试考试，准备上来说...的确不是可以只看某一方面的内容的。

这道题，感觉题目的确写的比较抽象，字典序最小，那里是关键所在。比起正常的来说困难的地方我觉得在补这个过程中，容易出问题。

4.

儿童节那天有K位小朋友到小明家做客。小明拿出了珍藏的巧克力招待小朋友们。小明一共有N块巧克力，其中第i块是Hi x Wi的方格组成的长方形。为了公平起见，小明需要从这 N 块巧克力中切出K块巧克力分给小朋友们。切出的巧克力需要满足：

（1）形状是正方形，边长是整数（2）大小相同

例如，一块6x5的巧克力可以切出6块2x2的巧克力，或者2块3x3的巧克力。

当然小朋友们都希望得到的巧克力尽可能大，你能帮小Hi计算出，最大的边长是多少么？

输入

第一行包含两个整数N和K。(1 <= N, K <= 100000)

以下N行每行包含两个整数Hi和Wi。(1 <= Hi, Wi <= 100000)

输入保证每位小朋友至少能获得一块1x1的巧克力。

输出

输出切出的正方形巧克力最大可能的边长。

运行示例

输入

2 10

6 5

5 6

输出

2

def cho\_minx(cho0, n, k):

    x\_min = 0

    for i in range(n):

        if min(cho0[i]) > x\_min:

            x\_min = min(cho0[i])

    x\_min = x\_min // k

    return x\_min

def cho\_att(x\_min, n, k):

    count = k + 1

    x = x\_min - 1

    while count >= k:

        x += 1

        if x == 0:

            x = 1

        count = 0

        for i in range(n):

            a = cho0[i][0] // x

            b = cho0[i][1] // x

            count += a \* b

    return x - 1

n, k = map(int, input().split())

cho0 = []

for i in range(n):

    cho0.append(list(map(int, input().split())))

x\_min = cho\_minx(cho0, n, k)

x = cho\_att(x\_min, n, k)

print(x)

**5.**

2,3,5,7,11,13,....是素数序列。

类似我1们有：7，37，67，97，127，157这样完全由素数组成的等差数列，叫等差素数数列，而这个等差素数数列的公差为30，长度为 6。

2004 年，格林与华人陶哲轩合作证明了：存在任意长度的素数等差数列。 这是数论领域一项惊人的成果！

有这一理论为基础，请你借助手中的计算机，满怀信心地搜索：

长度为10的等差素数列，其公差最小值是多少？

机考时候我极其昏头......想不出来解决办法，只知道暴力算（鼠鼠脑子太笨了）还是只写了一半放上去，这绝对是我的大错qaq。

暴力方法：

def is\_prime(num):

    if num <= 1:

        return False

    for i in range(2, int(num\*\*0.5) + 1):

        if num % i == 0:

            return False

    return True

diff = 0 # 公差

for i in range(2,1000):

    if not is\_prime(i):

        continue

    else:

        for j in range(1,501):

            length = 0  # 记当前等差素数数列长度,同时每一次重新开始查找时重置为0

            for k in range(i + j,i + j \* 11,j):

                if not is\_prime(k):

                    break

                else:

                    length += 1

                if length == 9: # i是数列中第一个元素，我们检查时候是从i+j开始的

                    diff = j

                    break

print(diff)

本人能力较差，写的内容也比较平庸，望见谅。