



yuhuan345

(身份

证号):

手机

号: 13208011121

考生信息

成绩 (*以下结果包含考生参加该场次的所有试卷)

90 分

满分100

9 名

排名: 共206人

01:27:20

限时2h

100%

题目完成率

作弊情况

判定: 无作弊风险

页面操作: 共跳出页面4次 复制粘贴1次

试卷1报告

CSDN竞赛38期

考生 [yuhuan345](#) 于2023/03/19 10:28提交试卷

90 /100分

试卷目前得分

01:27:21

耗时

9 /206名

排名

答题情况

目录	题型	题目数	耗时	得分	得分率
编程题	编程题	4	1h27m17s	90	90%

页面操作记录

编程题

跳出页面: 共跳出考试页面4次 平均时长: 1m54s

复制粘贴: 共复制粘贴1次

答题详情

编程题 (共4道) 得分: 90/100分 得分率: 90% 用时: 1h27m17s

候选人编程总结 使用语言: PYTHON3 得分: 25/25 得分率: 100% 提交人数: 181人 平均分: 10.91/25

1、题目名称：代写匿名信

小Q想要匿名举报XX领导不务正业！小Q害怕别人认出他的字迹。他选择从报纸上剪裁下来英文字母组成自己的举报信。现在小Q找来了报纸，和自己的举报信的Txt，你能帮他确定一下是否能够完成匿名信吗？

考生解决方案

```
# 请关闭中文输入法，用英文的字母和标点符号。
# 如果你想运行系统测试用例，请点击【执行代码】按钮，如果你想提交作答结果，请点击【提交】按钮，
# 注意：除答案外，请不要打印其他任何多余的字符，以免影响结果验证
# 本OJ系统是基于 OxCoder 技术开发，网址：www.oxcoder.com
# 模版代码提供基本的输入输出框架，可按个人代码习惯修改
```

```
from collections import *
class Solution:
    def __init__(self) -> None:
        pass

    def solution(self, words, msg):
        result = "Yes"

        # TODO: 请在此编写代码
        cc = Counter(words)
        for ch in msg:
            if ch == ' ': continue
            if cc[ch] <= 0:
                result = "No"
                break
            cc[ch] -= 1
        return result

if __name__ == "__main__":
    words = input().strip()
    msg = input().strip()

    sol = Solution()
    result = sol.solution(words, msg)
    print(result)
```

代码执行总结 运行次数：108 运行时间：18ms 占用内存：8.84k

测试用例 基本测试用例通过率：100%

- | | |
|-------------|------|
| 1.out | ▶ 通过 |
| 运行时间： | |
| 占用内存：8.836k | |
| 2.out | ▶ 通过 |
| 运行时间： | |
| 占用内存：8.84k | |
| 3.out | ▶ 通过 |
| 运行时间： | |
| 占用内存：8.84k | |
| 4.out | ▶ 通过 |

运行时间:
占用内存: 8.84k

5.out ▶ 通过

运行时间:
占用内存: 8.84k

6.out ▶ 通过

运行时间:
占用内存: 8.84k

7.out ▶ 通过

运行时间:
占用内存: 8.84k

8.out ▶ 通过

运行时间:
占用内存: 8.84k

9.out ▶ 通过

运行时间:
占用内存: 8.84k

10.out ▶ 通过

运行时间:
占用内存: 8.84k

候选人编程总结 使用语言: PYTHON3 得分: 22.5/25 得分率: 90% 提交人数: 161人 平均分: 5.89/25

2、题目名称: 寻因找祖

寻找因子个数为n的最小整数x.

考生解决方案

请关闭中文输入法, 用英文的字母和标点符号。
如果你想运行系统测试用例, 请点击【执行代码】按钮, 如果你想提交作答结果, 请点击【提交】按钮,
注意: 除答案外, 请不要打印其他任何多余的字符, 以免影响结果验证
本OJ系统是基于 OxCoder 技术开发, 网址: www.oxcoder.com
模版代码提供基本的输入输出框架, 可按个人代码习惯修改

```
class Solution:
    def __init__(self) -> None:
        pass

    def solution(self, n):
        fn = n
        fs = []
        f = 2
        while f*f <= n:
            while n%f == 0:
                fs.append(f)
                n //= f
            f += 1
        if n > 1: fs.append(n)
        fs.sort(reverse=True)
```

```
ps = [2,3,5,7,11,13,17,19,23,29,31,37]
r = 1
for i, f in enumerate(fs):
    r *= ps[i]**(f-1)

# TODO: 请在此编写代码

return min(r,2**(fn-1))

if __name__ == "__main__":

    n = int(input().strip())

    sol = Solution()
    result = sol.solution(n)

    print(result)
```

代码执行总结 运行次数: 180 运行时间: 16ms 占用内存: 13.082k

测试用例 基本测试用例通过率: 90%

1.out ▶ 通过

运行时间:
占用内存: 13.078k

2.out ▶ 通过

运行时间:
占用内存: 13.082k

3.out ✖ 失败

运行时间:
占用内存: 13.082k

4.out ▶ 通过

运行时间:
占用内存: 13.082k

5.out ▶ 通过

运行时间:
占用内存: 13.082k

6.out ▶ 通过

运行时间:
占用内存: 13.082k

7.out ▶ 通过

运行时间:
占用内存: 13.082k

8.out ▶ 通过

运行时间:
占用内存: 13.082k

9.out ▶ 通过

运行时间:
占用内存: 13.082k

10.out ▶ 通过

运行时间:

占用内存: 13.082k

候选人编程总结 使用语言: PYTHON3 得分: 25/25 得分率: 100% 提交人数: 144人 平均分: 5.31/25

3、题目名称: 小Q新式棋盘

已知棋盘大小为 $n \times n$ 。每个位置都有自己的权值 q 。该棋盘中有多少对行权值和小于列权值和。

考生解决方案

```
# 请关闭中文输入法，用英文的字母和标点符号。
# 如果你想运行系统测试用例，请点击【执行代码】按钮，如果你想提交作答结果，请点击【提交】按钮，
# 注意：除答案外，请不要打印其他任何多余的字符，以免影响结果验证
# 本OJ系统是基于 OxCoder 技术开发，网址: www.oxcoder.com
# 模版代码提供基本的输入输出框架，可按个人代码习惯修改
```

```
class Solution:
    def __init__(self) -> None:
        pass

    def solution(self, n, mat):
        n = len(mat)
        row = [sum(mat[i][j] for j in range(n)) for i in range(n)]
        col = [sum(mat[i][j] for i in range(n)) for j in range(n)]
        row.sort()
        col.sort()
        j = 0
        r = 0
        for i in range(n):
            while j < n and row[j] < col[i]:
                j += 1
            r += j

        # TODO: 请在此编写代码

        return r

if __name__ == "__main__":

    n = int(input().strip())

    vector = []
    for i in range(n):
        vector.append([int(item) for item in input().strip().split()])

    sol = Solution()
    result = sol.solution(n, vector)

    print(result)
```

代码执行总结 运行次数: 18 运行时间: 18ms 占用内存: 13.105k

测试用例 基本测试用例通过率: 100%

1.out ▶ 通过

运行时间:
占用内存: 13.102k

2.out ▶ 通过

运行时间:
占用内存: 13.105k

3.out ▶ 通过

运行时间:
占用内存: 13.105k

4.out ▶ 通过

运行时间:
占用内存: 13.105k

5.out ▶ 通过

运行时间:
占用内存: 13.105k

6.out ▶ 通过

运行时间:
占用内存: 13.105k

7.out ▶ 通过

运行时间:
占用内存: 13.105k

8.out ▶ 通过

运行时间:
占用内存: 13.105k

9.out ▶ 通过

运行时间:
占用内存: 13.105k

10.out ▶ 通过

运行时间:
占用内存: 13.105k

候选人编程总结 使用语言: PYTHON3 得分: 17.5/25 得分率: 70% 提交人数: 129人 平均分: 2.27/25

4、题目名称: 拯救公主

在Flower Kingdom里,住着一位美丽的公主Ana,有一天Ana得了一种怪病,神医告知国王,在遥远的幽谷中有一种药能治愈Ana,但是神医只有一份不完整的地图,地图的描述如下:

该地图的共有3行,第一行有m列,m为奇数,第二行有m+1列,第三行有m+2列;

每一行用一个字符串表示,只有【两种字符】: '.'表示草地,可以从它上面通过, '*'表示岩石,每一行最多一个'*';

入口在左上角,由于在对角线方向上,因此即使对角线两边都有岩石,但是缝隙较大,人可以通过,故人可以向八个方向行走;

真实地图是由该地图的【每一行无限循环】得到的,这种神奇的药草就生长在第x行第y列的草地上,药草可能会在岩石上;

国王决定派遣勇敢的骑士David前去寻找拯救公主的解药; 现在聪明的你是否知道David能否找到该药?

考生解决方案

请关闭中文输入法，用英文的字母和标点符号。
如果你想运行系统测试用例，请点击【执行代码】按钮，如果你想提交作答结果，请点击【提交】按钮，
注意：除答案外，请不要打印其他任何多余的字符，以免影响结果验证
本OJ系统是基于 OxCoder 技术开发，网址：www.oxcoder.com
模版代码提供基本的输入输出框架，可按个人代码习惯修改

```
from math import *
class Solution:
    def __init__(self) -> None:
        pass

    def solution(self, arr, vec):
        def calc(t, delta):
            if delta%t == 0: return 0
            k = t-(delta%t)
            return k
        m, x, y = arr
        rs = [0]*3
        if vec[0][0] == '*': return False
        for i in range(3):
            for j in range(m+i):
                if vec[i][j] == '*':
                    if x == i+1 and (y-(j+1))%(m+i) == 0:
                        return False
                    rs[i] = j+1
        #print(rs)
        if any(r == 0 for r in rs):
            return True
        y1 = calc(m, rs[1]-rs[0])*(m+1)+rs[1]
        y2 = calc(m+1, rs[2]-rs[1])*(m+2)+rs[2]
        #print(y1,y2)
        if y1*y2//gcd(y1,y2) <= y:
            return False
        result = True

        # TODO: 请在此编写代码

        return result

if __name__ == "__main__":
    sol = Solution()
    T = int(input().strip())
    for i in range(T):

        arr = [int(item) for item in input().strip().split()]

        vec = []
        for i in range(3):
            vec.append(input().strip())

        result = sol.solution(arr, vec)
        print("YES" if result else "NO")
```

代码执行总结 运行次数: 216 运行时间: 17ms 占用内存: 17.961k

测试用例 基本测试用例通过率：70%

1.out ▶ 通过

运行时间：
占用内存：17.957k

2.out ▶ 通过

运行时间：
占用内存：17.961k

3.out ▶ 通过

运行时间：
占用内存：17.961k

4.out ✖ 失败

运行时间：
占用内存：17.961k

5.out ▶ 通过

运行时间：
占用内存：17.961k

6.out ▶ 通过

运行时间：
占用内存：17.961k

7.out ▶ 通过

运行时间：
占用内存：17.961k

8.out ✖ 失败

运行时间：
占用内存：17.961k

9.out ▶ 通过

运行时间：
占用内存：17.961k

10.out ✖ 失败

运行时间：
占用内存：17.961k

编程题页面操作记录

跳出页面：共跳出考试页面4次 平均时长：1m54s

复制粘贴：共复制粘贴1次