

2023年NOC初中组国赛真题

二进制回文

编程实现：

输入一个正整数，判断它的二进制形式是否是回文数，如果是输出True，不是输出False

例：数字3的二进制形式是11，是一个回文数，输出True。

输入描述

输入一个正整数n。

输出描述

判断数字n的二进制形式是否是回文数，如果是输出True，不是输出False。

输入样例

3

输出样例

True

参考代码

```
1 n = int(input())
2 s = bin(n)[2:]
3 print(s == s[::-1])
```

拍卖

编程实现：

为了分配一件物品，两个小伙伴决定进行一场拍卖，为了避免价格过高或者被一口价买走，定下以下规则：

- 1、起拍价必须小于100；
- 2、谁先出价100谁获得这件物品；
- 3、加价为1到5之间的整数；

现在由你先出价，假设对方能按最优策略加价，输入你报出的起拍价，请根据这个价格判断最后是否能获得这件物品，如果能获得，输出True，否则输出False。

例:你出价95，那么对方加价5，你一定不能获得这件物品，输出False；你出价94，那么对方不管怎么加价，你都能直接加价到100，直接获得这件物品，输出True。

输入描述

输入一个正整数n，表示你出的起拍价（ $1 \leq n < 100$ ）。

输出描述

输出是否能获得这件物品，如果能获得，输出True，否则输出False。

输入样例

94

输出样例

True

参考代码

```
1 n = int(input())
2 ls = [False for i in range(100)]
3 num = 94
4 while num >= 1:
5     ls[num] = True
6     num -= 6
7 print(ls[n])
8
9 # 法2
10 n = int(input())
11 if (100 - n) % 6 == 0:
12     print(True)
13 else:
14     print(False)
```

数字谜题

编程实现：

现在有一串只包含1，2，3的数字，如果将这串数字中的一个123替换成一个321，算成一次操作，那么一共要经过多少次操作才能让这串数字中的123全部消失。

例：数字串12123，操作一次：12321，操作一次：32121，一共经过2次操作。

输入描述

输入一串长度为n的数字（ $1 \leq n < 100000$ ），只包含数字1，2，3。

输出描述

输出替换的操作次数。

输入样例

12123

输出样例

2

参考代码

```
1 s = input()
2 cnt = 0
3 while '123' in s:
4     n = s.count('123')
5     s = s.replace('123', '321')
6     cnt += n
7 print(cnt)
```

```

8
9 # 优化之后通过全部评测点
10 # ls = list(input())
11 # cnt = 0
12 # i = 0
13 # while i <= len(ls) - 3:
14 #     if ls[i:i+3] == ['1', '2', '3']:
15 #         cnt += 1
16 #         ls[i:i+3] = ['3', '2', '1']
17 #         i -= 2
18 #         if i < 0:
19 #             i = 0
20 #     else:
21 #         i += 1
22 # print(cnt)

```

电报加解密

编程实现

A国和B国正在陷入战火之中，双方都互派间谍，潜入对方内部，伺机行动。为了保证信息不被泄露，A国司令部向小猴发送了最新的军用密码的编码规则。

加密的规则：

- 电报中的小写字母，按字母表顺序替换为**后一个**对应的大写字母，如果超出字母表，则转回到第一个字母；
- 电报中的大写字母，按字母表顺序替换为**前一个**对应的小写字母，如果超出字母表，则转回到最后一个字母；
- 电报中的数字，按数码从小到大的顺序替换为**后一个**对应的数码，例如 0 替换为 1、1 替换为 2、9 替换为 0。
- 其他字符不做变化。

解密的规则为加密规则的逆过程。

例如，字母 g 以字母 H 代替，字母 N 以字母 m 代替。因此电报 gN96 经过加密之后为 Hm07，电报 Hm07 经过解密之后为 gN96。

现在，小猴收到了A国司令发来的 n 份电报信息，请你帮助小猴对这些信息进行解密，并输出**解密后**的结果。

输入描述

第一行，包含一个正整数 n。($1 \leq n \leq 100$)

接下来 n 行，每行包含一个字符串 s_i ，第 i 行的字符串 s_i 表示小猴收到的第 i 份电报信息。

保证字符串 s_i 中只包含大小写英文字母、数字、空格和英文标点符号（保证只会出现：','、':'、':'、'!' 和 '?'）。

输出描述

共 n 行，第 i 行输出一个字符串，表示第 i 份电报经过解密之后的内容。

输入样例

```
3
fPPE MVDL PO ZPVS FYBN!
aFMJFWF JO ZPVSTFMG!
h XJTI UIF FYBN B TVDDFTT!
```

输出样例

```
Good luck on your exam!
Believe in yourself!
I wish the exam a success!
```

参考代码

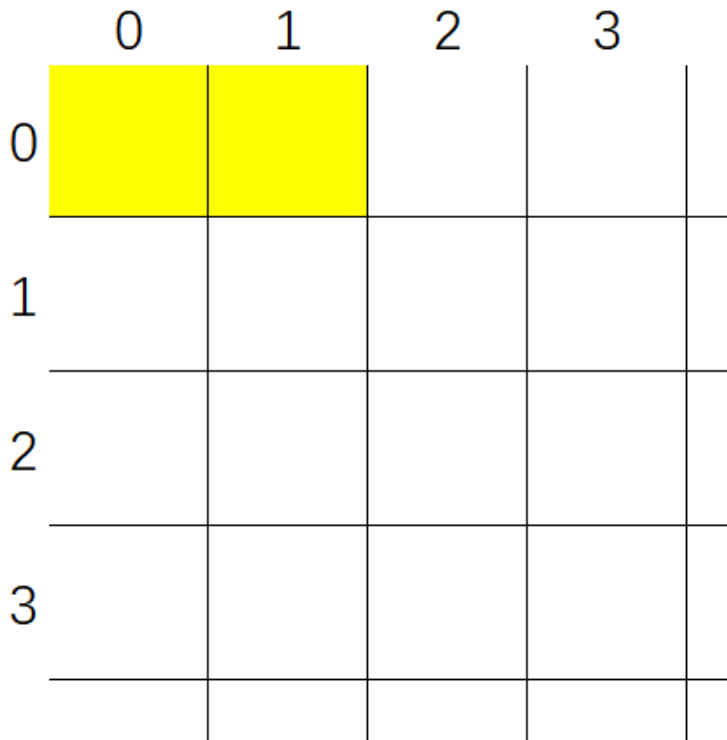
```
1 n = int(input())
2 ls = []
3 for i in range(n):
4     ls.append(input())
5 for s in ls:
6     ans = ''
7     for ch in s:
8         if ch.islower():
9             ans += chr((ord(ch) - ord('a') + 1) % 26 + ord('A'))
10        elif ch.isupper():
11            ans += chr((ord(ch) - ord('A') + 25) % 26 + ord('a'))
12        elif ch.isdigit():
13            ans += chr((ord(ch) - ord('0') + 9) % 10 + ord('0'))
14        else:
15            ans += ch
16    print(ans)
```

办事处选址

编程实现：

现将某区域的地图变成一个平面的方格图，平面上有 n 处居民住宅，已知 n 处居民住宅，每处居民住宅位置所处的行数和列数，现计划设置一处便民办事处，使办事处去到各个住宅位置的距离之和最短（只能上下左右走，且办事处可以和住宅处在同一个方格），请问最短距离之和是多少？

例如：共有两处居民住宅，位置如下图：



第一处居民住宅在0行0列，第二处居民住宅在0行1列，那么办事处可设置在0行0列处，到第一处居民住宅的距离为0，到第二处居民住宅的距离为1，最短距离之和为1。

输入描述

第一行输入一个正整数n，表示有n处居民住宅（ $1 \leq n \leq 10000$ ）。

第二行往后n行，每一行输入一对数字，表示每处居民住宅的行数和列数（ $0 \leq \text{行数} < 100$ ， $0 \leq \text{列数} < 100$ ），中间用空格隔开。

输出描述

输出最短距离之和。

输入样例

```
2
0 0
0 1
```

输出样例

```
1
```

参考代码

```
1 n = int(input())
2 lxx = []
3 lyy = []
4 for i in range(0, n):
5     x, y = [int(i) for i in input().split()]
6     lxx.append(x)
7     lyy.append(y)
8 lxx.sort()
9 lyy.sort()
10 midx = lxx[n//2]
11 midy = lyy[n//2]
```

```
12 tot = 0
13 for i in range(0, n):
14     tot += abs(midx - lxs[i]) + abs(midy - lsy[i])
15 print(tot)
```

挑选礼物

****编程实现****

小猴家里有一片香蕉林，由于小猴的好朋友小美快要过生日了，小猴打算在一排共 n 棵香蕉树中选择两颗香蕉树作为小美的生日礼物，为了显得不是那么随意，小猴认为选取的两棵香蕉树的距离不能超过 d ，但是方案太多了，小猴想要先知道一共有多少种选择的方案。

已知一排香蕉树的位置坐标依次为 $x_1, x_2, x_3, \dots, x_n$ ，请你帮助小猴计算一下选择两棵距离不超过 d 的香蕉树的方案有多少种。

注意：方案 x_i, x_j 和方案 x_j, x_i 认为是同一种方案。

****输入描述****

第一行，包含一个整数 n ($1 \leq n \leq 5000$)。

第二行，包含 n 个整数 $x_1, x_2, x_3, \dots, x_n$ 。

第三行，包含一个整数 d 。

****输出描述****

一行，包含一个整数，表示结果。

****输入样例****

5

5 16 3 20 13

10

****输出样例****

6

参考代码

```
1 n = int(input())
2 x = list(map(int, input().split()))
3 d = int(input())
4
5 x.sort() # 将香蕉树位置从小到大排序
6 ans = 0
7
8 for i in range(n):
9     j = i + 1 # j指向i的下一个位置
10    while j < n and x[j] - x[i] <= d:
11        ans += 1
12        j += 1
13
```

```
14 print(ans)
15
```

连连看

编程实现：

现在有一个连连看卡牌游戏。有n张卡牌排列在一起，每张卡牌上有一个分数，只要将相邻两张卡牌连到一起，就会合成一张新的卡牌，卡牌上的分数为之前两张卡牌的分数之和，你将获得新卡牌分数的相应积分。经过多次操作后，最后只剩一张卡牌，游戏结束。给定初始每张卡牌的分数，请问最多能获得多少分？

例如：场上有3张卡牌，分数分别为10分、23分、5分，将10分卡牌与23分卡牌合成，可得到33分卡牌，再将33分卡牌与5分卡牌合成，可得到38分卡牌，获得总分数为33+38=71分为最多的分数。

输入描述

第一行输入一个正整数n，表示有n张卡牌（ $1 \leq n \leq 100$ ）。

第二行输入n个正整数，表示每张卡牌的初始分数（ $1 \leq \text{分数} \leq 100$ ），正整数之间由空格隔开。

输出描述

输出最多可以获得的分数。

输入样例

```
3
10 23 5
```

输出样例

```
71
```

参考代码

```
1 n = int(input())
2 a = [int(i) for i in input().split()]
3 dp = [[0] * n for i in range(n)]
4 for length in range(2, n + 1):
5     for i in range(n - length + 1):
6         j = i + length - 1
7         dp[i][j] = 0
8         for k in range(i, j):
9             dp[i][j] = max(dp[i][j], dp[i][k] + dp[k + 1][j] + sum(a[i:j + 1]))
10 print(dp[0][n - 1])
```