

# 2022年全国青少年信息素养大赛Python

## 国赛

### 1、分苹果

#### 题目背景：

把一堆苹果分给 $n$ 个小朋友，每个人拿到的苹果数量不同，并且每个人至少有一个。

任意输入小朋友的数量 $n$ ，问这堆苹果至少应该有多少个。

#### 输入描述：

任意输入小朋友的数量 $n$

#### 输出描述：

输出这堆苹果至少应该有多少个

#### 样例输入：

3

#### 样例输出：

6

#### 注意：

`input()`内不添加任何参数

只有完全正确才可提交，若无法点击提交说明答题存在错误，可及时进行检查并修改

```
1 # 方法1, 变量法
2 n = int(input())
3 s = 0
4 for i in range(1, n + 1):
5     s += i
6 print(s)
```

## 2、韩信点兵

### 题目背景：

韩信点兵又称为中国剩余定理，相传汉高祖刘邦问大将军韩信统御士兵多少，韩信答说，每3人一组余2人、5人一组余4人、7人一组余6人。

输入任意一个数字 $n$  ( $0 \leq n \leq 10000$ ) 表示刘邦总共的兵力，计算韩信可能统御的士兵数量并依次输出。

### 输入描述：

输入任意一个数字 $n$  ( $0 \leq n \leq 10000$ ) 表示刘邦总共的兵力

### 输出描述：

计算韩信可能统御的士兵数量并依次输出，每行一个，从小到大

### 样例输入：

500

### 样例输出：

104

209

314

419

### 注意：

input()内不添加任何参数

只有完全正确才可提交，若无法点击提交说明答题存在错误，可及时进行检查并修改

```
1 # 将输入的数字转成整数
2 n = int(input())
3
4 # 使用枚举算法，判断每个数字是否同时满足3个条件
5 for i in range(n + 1):
6     if i % 3 == 2 and i % 5 == 4 and i % 7 == 6:
7         print(i)
```

### 3、借书问题

#### 题目背景：

小明家里有很多不同的趣味书籍，有3个好朋友每人来借一本书（同一本书一次不能借给两个人）

任意输入书籍的数量 $n(3 \leq n \leq 100)$ ，输出在所有的书籍中，找3本书给3个人有多少种不同的排列组合方法

#### 输入描述：

任意输入书籍的数量 $n(3 \leq n \leq 100)$

#### 输出描述：

输出在所有的书籍中，找3本书给3个人有多少种不同的排列组合方法

#### 样例输入：

3

#### 样例输出：

6

#### 注意：

`input()`内不添加任何参数

只有完全正确才可提交，若无法点击提交说明答题存在错误，可及时进行检查并修改

```
1 # 方法1, 数学方法
2 n = int(input())
3 print(n * (n - 1) * (n - 2))
```

#### 4、四位数密码

##### 题目背景：

情报员使用4位数字来传递信息，同时为了防止信息泄露，需要将数字进行加密。数据加密的规则是：

- 1). 每个数字都进行如下处理：该数字加上5之后除以10的余数，替换原数字；
- 2). 将处理后第一位数字与第三位数字进行交换，第二位数字与第四位数字进行交换；
- 3). 现在任意输入一个四位数 $n$  ( $1000 \leq n \leq 9999$ )，输出加密之后的数字。

##### 输入描述：

一个四位数

##### 输出描述：

加密后的四位数

##### 样例输入：

1234

##### 样例输出：

8967

##### 注意：

`input()`内不添加任何参数

只有完全正确才可提交，若无法点击提交说明答题存在错误，可及时进行检查并修改

```
1 n = int(input())
2 # 分别获取个位、十位、百位和千位
```

```

3  gw = n % 10
4  sw = n // 10 % 10
5  bw = n // 100 % 10
6  qw = n // 1000
7  # 加密处理
8  gw = (gw + 5) % 10
9  sw = (sw + 5) % 10
10 bw = (bw + 5) % 10
11 qw = (qw + 5) % 10
12 # 数字交换
13 gw, bw = bw, gw
14 sw, qw = qw, sw
15 # 直接输出
16 print(qw,bw,sw,gw,sep="")

```

## 5、邮票面值

### 题目背景：

我们在寄信的时候都要贴邮票，在邮局有一些小面值的邮票，通过这些小面值邮票中的一张或者几张的组合，可以满足不同邮件的不同邮资。

已知每个信封上最多能贴5张邮票，邮票的种类至少需要3种，任意输入一个数字n代表邮票的种类，然后依次输入n个数字代表不同种类的邮票面值，计算并输出每个信封上可以构成的邮资的最大值。

### 输入描述：

第一行输入一个数字n代表邮票的种类；

然后依次输入n行数字代表n个不同种类的邮票面值；

### 输出描述：

输出每个信封上可以构成的邮资的最大值。

### 样例输入：

4

1

2

3

4

样例输出：

17

注意：

input()内不添加任何参数

只有完全正确才可提交，若无法点击提交说明答题存在错误，可及时进行检查并修改

```
1 # 获取数字n
2 n = int(input())
3 # 定义列表stamps, 用于保存不同的面值
4 stamps = []
5 # 依次输入不同邮票面值, 并存入列表中
6 for i in range(n):
7     stamps.append(int(input()))
8
9 # 对列表进行排序
10 stamps.sort()
11
12 # 取最后三项, 其中最后一项取3次
13 _max = sum(stamps[-3:]) + stamps[-1] * 2
14 print(_max)
```

## 6、取球

题目背景：

一个箱子里面存放着50个球，其中10个红色，10个黄色，30个蓝色。

现在从箱子里面任意取n个球，计算并输出不同颜色的球的数量组合一共有多少种可能性。

输入描述：

一个数字n

输出描述：

一个数字，可能的组合总数

样例输入：

2

样例输出：

6

注意：

input()内不添加任何参数

只有完全正确才可提交，若无法点击提交说明答题存在错误，可及时进行检查并修改

```
1 # 获取数字n
2 n = int(input())
3 # 枚举算法
4 result = []
5 for x in range(10):
6     for y in range(10):
7         for z in range(30):
8             if x + y + z == n and (x,y,z) not in result:
9                 result.append((x,y,z))
10 # 输出组合总数
11 print(len(result))
```

## 7、统计连续字符

题目背景：

依次输入一个字符串s与一个数字n，寻找字符串中所有连续出现n次的字符，统计符合要求的字符数量并输出。

例如输入abbcccdddd，输入2，字符串中连续出现2次的字符包括b、c、d这3种。

输入描述：

第一行一个字符串（保证有连续的n个字符）

第二行一个数字n

输出描述：

符合要求的字符数量

样例输入：

abbcccdddd

2

样例输出：

3

input()内不添加任何参数

只有完全正确才可提交，若无法点击提交说明答题存在错误，可及时进行检查并修改

```
1 # 获取字符串
2 s = input()
3 # 获取数字n
4 n = int(input())
5
6 res = []
7 i = 0
8 while i < len(s):
9     j = i + 1
10    # 判断前后连个字符是否相等
11    while j < len(s) and s[i] == s[j]:
12        j += 1
13    # 将连续字符取出来，存入到列表中
14    res.append(s[i:j])
15    i = j
16
17 # 对结果进行过滤
18 res = [r for r in res if len(r) >= n]
19
20 # 输出次数
21 print(len(res))
```

## 8、n的n次方

题目背景：



当数字的位数高达上百位时，这样庞大的数字在计算机中的计算已经完全不能用简单的加减乘除来实现了，但是有时候对于这种高精度数字的处理也是编程中必不可缺的。

任意输入一个数字 $n$  ( $99999 < n < 999999$ )，计算 $n$ 的 $n$ 次方（即连续 $n$ 个 $n$ 相乘）计算之后，最后的3位数字是多少。

**输入描述：**

一个数字 $n$  ( $99999 < n < 999999$ )

**输出描述：**

计算 $n$ 的 $n$ 次方（即连续 $n$ 个 $n$ 相乘）之后,最后的3位数字是多少

**样例输入：**

123456

**样例输出：**

856

input()内不添加任何参数

只有完全正确才可提交，若无法点击提交说明答题存在错误，可及时进行检查并修改

```
1 n = int(input())
2 # 计算乘法，然后取余数
3 p = n ** n % 1000
4 print("%03d"%p)
```

## 9、海盗搜身

**题目背景：**

一群渔民被海盗抓住了，依次坐在编号为1~ $n$ 的凳子上，其中有一个人身上有藏宝图。

现在海盗想要找那个身上带着宝藏的渔民，海盗先将1号凳子上面的人搜身，没找到就隔1个凳子，将3号凳子上面的人搜身，没找到就隔2个凳子，将6号凳子上面的人搜身。

以后每次多隔一个凳子去搜索.....这样下去找了100次没找到，最终放弃了把渔民释放了。

任意输入一个数字 $n$  ( $5 \leq n \leq 20$ )代表渔民的数量，依次输出哪些编号的渔民不会被搜身。

输入描述：

任意输入一个数字 $n$  ( $5 \leq n \leq 20$ )代表渔民的数量

输出描述：

依次输出哪些编号的渔民不会被搜身，每行一个

样例输入：

10

样例输出：

2

4

7

9

注意：

input()内不添加任何参数

```
1  n = int(input())
2  peo = [i+ 1 for i in range(n)]
3  # 100次查找
4  j=0
5  step = 2
6  for i in range(100):
7      peo[j] = -1
8      j= (j + step) % n
9      step += 1
10 peo =[p for p in peo if p > 0]
11 for p in peo:
12     print(p)
```

## 10、父与子

### 题目背景：

学校举办亲子运动会，所有的父亲一组，孩子一组，出场规则是：父亲组先派一个人上场之后孩子组才能派一个人上场，假设每队3个人，可能的出场策略包括5种：

父父父子子子、父父子子父子、父父子父子子、父子父父子子、父子父子父子

任意输入父子的对数 $n(3 \leq n \leq 15)$ ，计算并输出有多少种出场策略。

### 输入描述：

输入父子对数 $n$

### 输出描述：

输出共有多少种出场策略

### 样例输入：

3

### 样例输出：

5

### 注意：

input()内不添加任何参数

只有完全正确才可提交，若无法点击提交说明答题存在错误，可及时进行检查并修改

```
1 # 对于给定元组，逐个判断，对元组进行切片操作，统计父和子的个数，进行比较
2 def is_valid(tp):
3     for i in range(n * 2):
4         t = tp[:i + 1]
5         father = t.count('父')
6         son = t.count('子')
7         if father < son:
8             return False
```

```
9         return True
10 n = int(input())
11 arr = ['父'] * n + ['子'] * n
12 res = []
13 from itertools import permutations# 获取全排列，并进行过滤操作
14 for a in permutations(arr,n *2):
15     if is valid(a) and a not in res:
16         res.append(a)
17 print(len(res))
```

-END-