

2016 年“梦想杯”江苏省小学生“信息与未来”夏令营

IT 小能手 PK 之程序设计

1. 数列求和（10 分）

小明写出了一个数列，第 i 项 a_i 的值为 i^2 。数列从第一项 ($i=1$) 开始如下：1, 4, 9, 16, 25, ……，编程求出这个数列前 n 项的和。

输入：整数 n ($1 \leq n \leq 1000$)

输出：一个整数： $a_1+a_2+\dots+a_n$ 的值。

	输入	输出
样例数据 1	6	91
样例数据 2	987	320988850

2. 幸运数字（15 分）

小明认为，如果以下条件之一成立，则正整数 x 就是一个幸运数字：

1、 x 是 4 的倍数；

2、 x 是 7 的倍数；

3、 x 转换为字符串后，包含子串“44”或“77”。

例如，105（7 的倍数）、442（包含“44”）、1284（4 的倍数）都是幸运数字，而 474 则不是。编程统计 1 到 n 中幸运数字的数量。

输入：整数 n ($1 \leq n \leq 1000000$)

输出：一个整数：1, 2, ……， n 中幸运数字的数量。

	输入	输出
样例数据 1	32	11
样例数据 2	987654	406754

3. 洗牌（15 分）

小明把 n (n 为偶数) 张牌按编号顺序 1、2、3、……、 n 排成一堆，然后开始洗牌。一次洗牌的过程如下：

1、对于一堆编号为 a_1, a_2, \dots, a_n ，首先将牌分成均匀的两堆：

a_1, a_2, \dots, a_m 和 $a_{m+1}, a_{m+2}, \dots, a_n$ （其中 $m=n/2$ ）

2、然后按顺序交叉插入：

$a_1, a_{m+1}, a_2, a_{m+2}, \dots, a_m, a_n$

洗牌过程总共重复了 k 次，请你编程帮助小明模拟洗牌的过程。

例如 $n=6$ ，初始时，牌堆中牌的编号为 1、2、3、4、5、6。

首次洗牌时，会将牌分成 1、2、3 和 4、5、6 两堆，交叉插入后的结果为 1, 4, 2, 5,

3, 6。

再次洗牌，会将牌分成 1, 4, 2 和 5, 3, 6 两堆，交叉插入后，得到 1, 5, 4, 3, 2,

6。

输入：正整数 n (牌的数量), k (洗牌的次数), i (牌的位置)。 $1 \leq n, k \leq 1000, 1 \leq i \leq n$, 保证 n 是偶数。

输出： n 张牌洗牌 k 次后，牌堆中第 i 张牌的编号。

	输入	输出
样例数据 1	6, 2, 5	2
样例数据 2	400, 300, 200	368

4. 字符串替换 (20 分)

小明最近迷上了字符串操作。对每个字符串，小明每次可以执行以下两种操作之一：

- 1、把字符串的某个字符改成任意一个其他字符，花费 1 的代价；
- 2、交换字符串中的两个字符，花费 0 的代价。

小明发现，把一个字符串通过一系列的操作，可以转换成任何一个与之等长的字符串，例如，把“hello”变为“world”的一种代价为 3 的操作序列如下：

- 1、hello -> wello (替换 h 为 w, 代价为 1)
- 2、wello -> wolle (交换 e 和 o, 代价为 0)
- 3、wolle -> worle (替换 l 为 r, 代价为 1)
- 4、worle -> world (替换 e 为 d, 代价为 1)

小明发现，无法用少于 3 次的代价将“hello”变为“world”。

显然，不同的转换方案花费的代价是不同的，请编程帮助小明计算把一个字符串变为另一个字符串的最小代价。

本题中的字符串根据给定的初始数值 s 按以下规则生成：

for $i=1, 2, \dots, n$

$s \leftarrow (s * 345) \bmod 19997$

第二个字符串的第 i 个字符的 ASCII 码为 $(97 + (s \bmod 26))$

输入：正整数 n (字符串长度), s (数据生成器的初始数值)。 $1 \leq n \leq 1000, 1 \leq s \leq 19997$ 。

输出：将第一个字符串转换为第二个字符串的最少代价。

	输入	输出
样例数据 1	4, 35	2
样例数据 2	100, 31	29

在样例 1 中，生成的字符串是“lzvv”和“xylv”，将第一个字符串变为第二个的最小代价为 2。

5. 方格取数 (20 分)

在 n 行、 m 列的方格矩阵中，每个方格都包含一个数字。小明可以从任意方格出发开始移动。每次移动可以移到与当前方格有一条边相邻的方格（即向上、下、左或右方向移动 1 格，且不能移出边界）。除此之外，你移动到的方格中的数字必须比当前方格中的数字更大。

请你帮助小明编程规划移动路径，使路径上经过的所有数字之和最大。

本题方格中的数据根据输入的初始数字 s 按照如下算法生成：

for $i=1, 2, \dots, n$

for $j=1, 2, \dots, m$

$s \leftarrow -(s * 345) \bmod 19997$

矩阵第 i 行第 j 列方格中的数字为 $(s \bmod 10) + 1$

输入：正整数 n 、 m （方格的大小）， s （数据生成器的初始数值）。 $1 \leq n, m \leq 100$, $1 \leq s \leq 19997$ 。

输出：所有合法路径中的最大数字和。

	输入	输出
样例数据 1	4, 5, 97	24
样例数据 2	40, 50, 1	47

样例数据 1 对应的矩阵如下。图中路径数字之和 $4+5+7+8=24$ 。

9	7	10	10	8
2	9	2	5	3
2	5	5	7	7
5	8	4	8	5

6. 素数分解（20 分）

素数，又称质数，是指除 1 和其自身之外，没有其他约数的正整数。例如，2、3、5、13 都是质数，而 4、9、12、18 则不是。

虽然素数不能分解成除 1 和其自身之外整数的乘积，但却可以分解成更多素数的和。你需要编程求出一个正整数最多能分解成多少个互不相同的素数的和。

例如， $21=2+19$ 是 21 的合法分解方法。 $21=2+3+5+11$ 则是分解为最多素数的方法。

输入： n （ $10 \leq n \leq 200$ ）

输出： n 最多能分解成多少个不同的素数的和。

	输入	输出
样例数据 1	21	4
样例数据 2	128	9