POJ3292【基础】

题目大意：
H-number 是形如 4*n+1 这样的数，如 1，5，9，等。H-Primes 是这样一个 H-number：它只能唯一分解成 1*它本身，而不能表示为其他两个 H-number 的乘积。一个 H-Semi-Prime 是一个这样的 H-number：它正好能表示成两个 H-Primes 的乘积（除了 1*它本身），这种表示法可以不唯一，但它不能表示为 3 个或者以上 H-primes 的乘积。现在给出一个整数 N，要求区间[1,N]内有多少个 H-Semi-Prime。（1<=N<=1000000）

输入：
有若干组测试数据，每一组测试数据只有一个整数 N，当 N=0 时测试数据结束。

输出：
对每一组测试数据，输出一行，包含两个整数，即 H 和[1,N]内有多少个 H-Semi-Prime。

题解：
筛法的基础应用。首先筛出 H-number，然后筛出 H-Primers，再筛出或两两枚举组合 H-Primers 得到 H-Semi-Primers。注意在筛 H-Semi-Primers 的时候用一下两个 H-Primers 相乘小于 1000000 的限制条件。

POJ2402【基础】

题目大意：
从 1 开始算，求第 n 个回文数。

输入：
有若干组测试数据，每一组测试数据有一行，包含一个整数 n，n<=2*10^9。
n=0 时测试数据结束。

输出：
每一组测试数据输出一行，包含一个整数，即第 n 个回文数。

题解：
显然，回文数的半长度（回文长度）增加 1 位，回文数的个数就是以前的 10 倍。1 位和 2 位的回文数有 9 个，依次类推可以算出各位有多少个回文数。因此得到 n 以后，先求出第 n 个回文数的长度（设为 s），并求出是这个长度中的第几个回文数（设为 k）。那么回文数前半段一定是 10^(s-1)+(k-1)。然后正反输出一次就可以了。

POJ2453【基础】

题目大意：

给出一个整数 N（1<=N<=1000000），求一个比它大的数 N'，要求 N' 和 N 在二进制下有相同多的 1，并且 N' 尽可能地小。

输入：

有若干组测试数据，每一组一行，包含一个整数 N，N=0 时测试数据结束。

输出：

每一组测试数据输出一行，包含一个整数 N'。

题解：

很容易想到从低位向高位搜索，找到某一位=1 而它的高一位=0 时交换，并且把比它低的位中所有的 1 移动到最低的几位里去。这题主要考位运算，当然先转成二进制再在数组里操作也是可以的，不过慢很多。

POJ1401【基础】

题目大意：

给出一个整数 N（1<=N<=1000000000），求 N! 末尾有多少个零。

输出：

第一行有一个整数 t，表示有 t 组测试数据。接下来有 t 行，每一行有一个整数 N。

输出：

每一组测试数据输出一行，包含一个整数，即 N!末尾有多少个 0。

题解：

因为同一个 N 内 2 的倍数总比 5 的倍数多，所以每逢一个 5 的倍数就能产生一个末尾的 0。只要算出有多少个 5 的倍数就好了。

POJ2407【基础】

题目大意：
给出一个正整数 N（1<=N<=1000000000），求有多少个比 N 小的素数。

输入：
有若干组测试数据，每一组测试数据一行，包含一个整数 N，N=0 时测试数据结束。

输出：
每一组测试数据输出一行，包含一个整数。

题解：
欧拉函数：少于或等于 n 的数中与 n 互质的数的数目。通式：φ(x)=x(1-1/p1)(1-1/p2)(1-1/p3)(1-1/p4)……(1-1/pn)，其中 p1、 p2……pn 为 x 的所有质因数，x 是不为 0 的整数。φ(1)=1（唯一和 1 互质的数就是 1 本身）。

POJ1142【基础】

题目大意：

给定一个整数 Num，求出一个大于 Num 的非素数 x，满足 x 的质因子的各位数之和之和等于这个数本身的各位数之和（比如 4937775=3*5*5*65837，4+9+3+7+7+7+5=42，3+5+5+6+5+8+3+7=42）。

输入：

有若干组测试数据，每一组测试数据一行，有一个整数 N。N 最多有八位数。
N=0 时测试数据结束。

输出：
每一组测试数据输出一行，包含一个满足要求的整数。

题解：
先打一个素数表出来，然后逐个求素因子即可。我打了一万个素数的表，235ms，后来网上看到一千个素数的表也能过，看来题目数据比较水。

POJ1091【难】

题目大意、输入输出要求见原题（有中文）。

题解：
这题实际上的题意是：输入整数 n 和 m，找 n 个数字（都小于等于 m），加上 m 共 n+1 个数字，使得这个 n+1 个数字满足存在 x1…xn+1 使得下列方程成立：a1*x1+a2*x2+a3*x3+…an*xn+m*xn+1=1，问共有多少组这样 n+1 个数字。
显然，由 Bezout 公式的扩展可以知道，实质上是要求有多少组 a1、a2……an 的组合满足这些 ai 之间互素。由于 m 固定，所以只需要先对 m 进行质数分解，找到前 n 个数字不都含有 m 的某个质因子，有多少种组合即可。
计算使用容斥原理：去掉都含其中一个质因子的，假设质因子为 x，则一个位置的数去掉含 x 的质因子的，有 m/x 个实根数，然后加上含两个的……经过计算即可得到结果。