MATHEMATICS OF COMPUTATION
VOLUME 39, NUMBER 159
JULY 1982, PAGES 225-234

A $p + 1$ Method of Factoring

By H. C. Williams

Abstract. Let N have a prime divisor p such that $p + 1$ has only small prime divisors. A method is described which will allow for the determination of p, given N. This method is analogous to the $p - 1$ method of factoring which was described in 1974 by Pollard. The results of testing this method on a large number of composite numbers are also presented.

1. Introduction. In 1974 Pollard [8] introduced a method of factorization which has since been called the $p - 1$ factorization technique. Actually, the test was known to D. N. and D. H. Lehmer many years before this but it was never published because, without a fast computer, it was not possible to determine how effective it would be in practice. For the convenience of the reader we give a brief description of this test.

Suppose N is a number to be factored and that N has a prime factor p such that

$$p = \left(\prod_{i=1}^{k} q_i^{\alpha_i}\right) + 1, \tag{1.1}$$

where q_i is the ith prime and $q_i^{\alpha_i} \leqslant B_1$. Let $q_i^{\beta_i}$ be that power of q_i such that $q_i^{\beta_i} \leqslant B_1$ and $q_i^{\beta_i+1} > B_1$ and put

$$R = \prod_{i=1}^{k} q_i^{\beta_i}. \tag{1.2}$$

Clearly, $p - 1 \mid R$ and since $a^{p-1} \equiv 1 \pmod{p}$ when $(N, a) = 1$, we have $a^R \equiv 1 \pmod{p}$. Thus, $p \mid (N, a^R - 1)$.

The algorithm now proceeds as follows. For a given B_1 put

$$R = r_1 r_2 r_3 \cdots r_m,$$

(for example, $m = k$, $r_i = q_i^{\alpha_i}$), $a_0 = a$, where $(a, N) = 1$ and define

$$a_i \equiv a_{i-1}^{r_i} \pmod{N} \; (i = 1, 2, 3, \ldots, m).$$

The values of a_i can be easily calculated by a power algorithm such as those mentioned in Knuth [4, p. 441ff.]. We now evaluate $a_m \equiv a^R \pmod{N}$ and $(a_m - 1, N)$. Even for fairly small values of B_1 it frequently occurs that $(a_m - 1, N)$ yields a nontrivial factor of N.

Pollard also gives in [8] two versions of a second step which can be appended to the above algorithm. We give one of these here.

Suppose instead of (1.1) we have

$$p = s\left(\prod_{i=1}^{k} q_i^{\alpha_i}\right) + 1,$$

Received March 30, 1981; revised September 28, 1981.
1980 *Mathematics Subject Classification.* Primary 10A25.

where s is a prime and $B_1 < s \leqslant B_2$. In this case we have $p \mid (a_m^s - 1, N)$. Let $\{s_j: j = 1, 2, \ldots, k\}$ be the ordered set of all primes such that $B_1 < s_j \leqslant B_2$, and put $2d_j = s_{j+1} - s_j$. Since the differences between successive primes increase very slowly, we see that there will not be very many distinct values for the d_j's. In fact, if we let $d(x)$ be the largest value of d_j for all primes between 1 and x, we have $d(200000) = 43$, $d(10^6) = 57$, and $d(4.444 \times 10^{12}) = 326$; see Brent [1]. Thus, it is not too difficult to tabulate $a_m^{2d_j}$ for all the distinct d_j. In fact, it is not really necessary to tabulate these values for all $d_j < d(x)$. The larger d_j occur very seldom and the method is almost as fast if the table extends to only $K \log x$ for some moderate value of K instead of $d(x)$ which seems to be $O((\log x)^2)$. This remark applies also to the second step of the $p + 1$ method.

We calculate $b_1 \equiv a_m^{s_1} \pmod N$ and define

$$b_{j+1} \equiv a_m^{2d_j} b_j \pmod N.$$

We now compute

$$(1.3) \quad G_t = \left(\prod_{i=0}^{c} (b_{t+i} - 1), N \right) \quad \text{for } t = 1, c+1, 2c+1, \ldots, [B_2/c]c + 1.$$

Since $b_j = a_m^{s_j} \pmod N$, we see that p must divide some G_i. Because greatest common divisors are more expensive to evaluate than products, we usually have $c > 1$.

In [3] Guy and Conway suggest that, by using Lucas functions, the first step of the $p - 1$ method can be converted into a factorization algorithm for finding a prime divisor p of N when $p + 1$ has only small prime factors. In this paper we give a description of how this can be done. We also present a number of new factorizations which have been obtained by using either the $p - 1$ or $p + 1$ method. It should be mentioned here that John Brillhart and Earl Ecklund have also implemented a version of the first step of the $p + 1$ method. However, in their few computer runs they were only able to find factors that had been previously discovered by the $p - 1$ method. We also point out that a version of the method using the finite field of p^2 elements is also possible, if the reader wishes to avoid Lucas functions. Indeed, the author has been informed that R. P. Brent has an implementation of the $p + 1$ method based on this interpretation.

2. The Lucas Functions. In order to develop the method and formulas required in the next section, we give here a description of some of the basic properties of the Lucas functions.

Let P, Q be integers, and let α, β be the zeros of $x^2 - Px + Q$. We define the Lucas functions by

$$(2.1) \qquad U_n(P, Q) = (\alpha^n - \beta^n)/(\alpha - \beta), \quad V_n(P, Q) = \alpha^n + \beta^n.$$

We also put $\Delta = (\alpha - \beta)^2 = P^2 - 4Q$. When there is no doubt as to the values of the arguments P and Q, we often omit them. These functions satisfy a large number of identities. We will require those given below

$$(2.2) \qquad \begin{cases} U_{n+1} = PU_n - QU_{n-1}, \\ V_{n+1} = PV_n - QV_{n-1}, \end{cases}$$

$$\text{(2.3)} \qquad \begin{cases} U_{2n} = V_n U_n, \\ V_{2n} = V_n^2 - 2Q^n, \end{cases}$$

$$\text{(2.4)} \qquad \begin{cases} U_{2n-1} = U_n^2 - QU_{n-1}^2, \\ V_{2n-1} = V_n V_{n-1} - PQ^{n-1}, \end{cases}$$

$$\text{(2.5)} \qquad \begin{cases} \Delta U_n = PV_n - 2QV_{n-1}, \\ V_n = PU_n - 2QU_{n-1}, \end{cases}$$

$$\text{(2.6)} \qquad \begin{cases} U_{m+n} = U_m U_{n+1} - QU_{m-1}U_n, \\ \Delta U_{m+n} = V_m V_{n+1} - QV_{m-1}V_n, \end{cases}$$

$$\text{(2.7)} \qquad \begin{cases} U_n\big(V_k(P,Q), Q^k\big) = U_{nk}(P,Q)/U_k(P,Q), \\ V_n\big(V_k(P,Q), Q^k\big) = V_{nk}(P,Q). \end{cases}$$

These identities can all be verified by direct substitution from (2.1), using the simple facts that $P = \alpha + \beta$, and $Q = \alpha\beta$.

We also note that if $(N, Q) = 1$ and $P'Q \equiv P^2 - 2Q \pmod N$, then $P' \equiv \alpha/\beta + \beta/\alpha$ and $Q' \equiv \alpha/\beta \cdot \beta/\alpha = 1$; hence,

$$\text{(2.8)} \qquad U_{2m}(P,Q) \equiv PQ^{m-1}U_m(P',1) \pmod N.$$

Finally, we need the following

THEOREM (SEE LEHMER [5]). *If p is an odd prime, $p \nmid Q$ and the Legendre symbol $(\Delta/p) = \varepsilon$, then*

$$U_{(p-\varepsilon)m}(P,Q) \equiv 0 \pmod p$$
$$V_{(p-\varepsilon)m}(P,Q) \equiv 2Q^{m(1-\varepsilon)/2} \pmod p.$$

3. The First Step of the Algorithm. Suppose that p is a prime divisor of N and

$$p = \left(\prod_{i=1}^{k} q_i^{\alpha_i}\right) - 1,$$

where q_i is again the ith prime and $q_i^{\alpha_i} \leqslant B_1$. If R is defined as in (1.2), we have $p + 1 \mid R$. By the theorem of Section 2 we see that if $(Q, N) = 1$ and $(\Delta/p) = -1$, then $p \mid U_R(P, Q)$, and therefore $p \mid (U_R(P, Q), N)$.

To find $U_R(P, Q)$, Guy and Conway seem to suggest that the first formulas of (2.2), (2.3), and (2.4) be used together with the second formula of (2.5) to obtain

$$U_{2n-1} = U_n^2 - QU_{n-1}^2,$$
$$U_{2n} = U_n(PU_n - 2QU_{n-1}),$$
$$U_{2n+1} = PU_{2n} - QU_{2n-1}.$$

These formulas can be used in a power algorithm routine similar to that suggested by Lehmer [6] to find $U_R(P, Q)$. The problem with this method is that R can be very large (for example, when $B_1 = 10^5$, $R > 10^{43410}$), and it is difficult to store its value in the computer. Also, if B_1 is increased to obtain a new R value, say R', we would have to start all over again at $U_1(P, Q)$ and $U_2(P, Q)$ to find $U_{R'}(P, Q)$ instead of continuing on from $U_R(P, Q)$. These problems can be overcome by using a different technique.

If $p \mid U_R(P, Q)$, then by (2.3) $p \mid U_{2R}(P, Q)$; thus, from (2.8) we have $p \mid U_R(P', 1)$. It follows that we lose no generality in assuming that $Q = 1$. Further, by the Theorem of Section 2, we also have

$$V_{(p-\varepsilon)m}(P, 1) \equiv 2 \pmod p;$$

hence, if $p \mid U_R(P, 1)$, then $p \mid (V_R(P, 1) - 2)$. We will assume throughout the remainder of this paper that $Q = 1$ in our Lucas functions.

The first step of our $p + 1$ algorithm is now the following:

Let $R = r_1 r_2 r_3 \cdots r_m$ as above and find P_0 such that $(P_0^2 - 4, N) = 1$. Define $V_n(P) = V_n(P, 1)$, $U_n(P) = U_n(P, 1)$ and

$$P_j \equiv V_{r_j}(P_{j-1}) \pmod N \; (j = 1, 2, 3, \ldots, m).$$

By the second formula of (2.7), we see that

$$P_m \equiv V_R(P_0) \pmod N. \tag{3.1}$$

We then calculate $(P_m - 2, N)$.

To find $V_r = V_r(P)$ from P we need only use the formulas

$$\begin{cases} V_{2f-1} \equiv V_f V_{f-1} - P, \\ V_{2f} \equiv V_f^2 - 2, \\ V_{2f+1} \equiv P V_f^2 - V_f V_{f-1} - P \pmod N, \end{cases} \tag{3.2}$$

(see the second formulas of (2.2), (2.3), and (2.4)).

Let

$$r = \sum_{i=0}^{t} b_i 2^{t-i} \qquad (b_i = 0, 1),$$

$f_0 = 1$, and $f_{k+1} = 2 f_k + b_{k+1}$; then $f_t = r$. Also, if $V_0(P) = 2$, $V_1(P) = P$, then, to find the pair $(V_{f_{k+1}}, V_{f_{k+1}-1})$ from $(V_{f_k}, V_{f_k - 1})$, we need only use the formula

$$\left(V_{f_{k+1}}, V_{f_{k+1}-1}\right) = \begin{cases} \left(V_{2f_k}, V_{2f_k - 1}\right) & \text{when } b_{k+1} = 0, \\ \left(V_{2f_k+1}, V_{2f_k}\right) & \text{when } b_{k+1} = 1, \end{cases} \tag{3.3}$$

together with (3.2).

4. The Second Step of the Algorithm. Suppose

$$p = s\left(\prod_{i=1}^{k} q_i^{\alpha_i}\right) - 1, \tag{4.1}$$

where s is a prime, and $B_1 < s \leqslant B_2$. Define s_j and $2d_j$ as in Section 1. If $(\Delta/p) = -1$ and $p \nmid P_m - 2$, then $p \mid (U_s(P_m), N)$ by (2.7) and (3.1).

Let $U[n] \equiv U_n(P_m)$, $V[n] \equiv V_n(P_m) \pmod N$, and tabulate $U[2d_j - 1]$, $U[2d_j]$, $U[2d_j + 1]$ for the distinct d_j by using

$$U[0] = 0, \quad U[1] = 1 \quad \text{and} \quad U[n+1] = P_m U[n] - U[n-1].$$

Put

$$T[s_i] \equiv \Delta U_{s_i}(P_m) = \Delta U_{s_i R}(P_0)/U_R(P_0) \pmod N,$$

by the first formula of (2.7) and (3.1). From the second formula of (2.6), we have

$$(4.2)\qquad \begin{cases} T[s_1] \equiv P_m V[s_1] - 2V[s_1 - 1], \\ T[s_1 - 1] \equiv 2V[s_1] - P_m V[s_1 - 1] \pmod N, \end{cases}$$

and from the second formula of (2.6) we get

$$(4.3)\qquad \begin{cases} T[s_{i+1}] \equiv T[s_i]U[2d_i + 1] - T[s_i - 1]U[2d_i], \\ T[s_{i+1} - 1] \equiv T[s_i]U[2d_i] - T[s_i - 1]U[2d_i - 1] \pmod N. \end{cases}$$

Thus, to execute the second step of the algorithm we need only use (4.2) and (4.3) to obtain $T[s_i]$, $i = 1, 2, 3, \cdots$, and then evaluate

$$(4.4)\qquad H_t = \left(\prod_{i=0}^{c} T[s_{i+t}], N \right)$$

for $t = 1,\ c+1,\ 2c+1, \ldots, c[B_2/c] + 1$. We must have $p \mid H_i$ for some i if p satisfies (4.1) and $(\Delta/p) = -1$.

5. Implementation and Results. One of the difficulties in implementing the $p+1$ algorithm of Sections 3 and 4 is the possibility that p in (4.1) is such that $(P_0^2 - 4/p) = +1$ for the selected value of P_0. There is no way of knowing beforehand that this will not occur. If we assume that the values of P_0 such that $(\Delta/p) = -1$ are randomly distributed, the probability that $(\Delta/p) = 1$ is the same as the probability that $(\Delta/p) = -1$, i.e., $\frac{1}{2}$. Thus the probability that $(\Delta_i/p) = 1$, $i = 1, 2, 3, \ldots, n-1$, for each of n trials at a P_0 value and $(\Delta_n/p) = -1$ for the nth trial is $(\frac{1}{2})^n$. (We assume that the P_0 values selected are independent.) It follows that the probability that we will find some Δ_i such that $(\Delta_i/p) = -1$ after at most n trials at a P_0 value is

$$\sum_{i=1}^{n} (\tfrac{1}{2})^i = 1 - (\tfrac{1}{2})^n.$$

Thus, if N has a prime factor p which satisfies (4.1), and we use the algorithm of Section 3 with three trials at a P_0 value, we would expect to find that $p \mid (P_m - 2, N)$ for seven of every eight such N tested. The referee has pointed out that, instead of making three guesses at P_0, one could make many guesses to obtain $\Delta_1, \Delta_2, \Delta_3, \ldots$ as possible values of Δ for which $(\Delta/p) = -1$. One could then, by time sharing, test each of these Δ_k values a fraction β_k of the time, where, of course, $\sum_{i=1}^{\infty} \beta_i = 1$. Let T_0 (a function of the largest prime factor of $p+1$) be the time required by the algorithm if we were able to choose a Δ for which $(\Delta/p) = -1$. Then the time-sharing algorithm succeeds with probability 1 in an expected time

$$T = T_0 \sum_{k=1}^{\infty} \frac{1}{2^k \beta_k}.$$

We naturally wish to select the β_k's in such a way that T is minimized. We note that by Cauchy's inequality

$$\left(\sum_{k=1}^{\infty} \frac{1}{2^k\beta_k}\right)\left(\sum_{k=1}^{\infty} \beta_k\right) \geqslant \left(\sum_{k=1}^{\infty} \left(\frac{1}{2^k\beta_k}\right)^{1/2} \beta_k^{1/2}\right)^2;$$

hence,

$$\sum_{k=1}^{\infty} \frac{1}{2^k\beta_k} \geqslant \left(\sum_{k=1}^{\infty} 2^{-k/2}\right)^2 = \left(\sqrt{2}+1\right)^2.$$

Thus, an optimal choice of β_k is $\beta_k = 2^{-k/2}(\sqrt{2}-1)$. Compared to taking 3 equally weighted trials, this method is slower when both succeed (ratio $3 + 2\sqrt{2} : 3$) but it succeeds with probability 1 instead of $7/8$.

Both the $p - 1$ and the $p + 1$ methods were implemented on an AMDAHL 470-V7 computer and run with c in (1.3) and (4.4) put equal to 14 and $B_1 = 10^5$, $B_2 = 2 \times 10^5$. Since the $p + 1$ method is much slower (two times slower for Step 1 and about four times slower for Step 2) than the $p - 1$ method, we always ran the $p - 1$ method first on any given N value.

Our programs were run on a total of 497 numbers. From an early version of a table of Brillhart et al. [2] (the most recent version includes the factors found here) we obtained 323 of these 497 numbers. These are factors of integers of the form $b^n - 1$ with $b = 2, 3, 5, 7, 10, 11, 12$. They were obtained by first dividing the main algebraic, including aurifeuillian, factor of $b^n - 1$ by any of its algebraic divisors and then trial dividing by all primes up to 2^{35}. Those remaining composite factors which were between 42 and 60 digits made up the 323 numbers referred to above. From an as yet unpublished table of factors of Fibonacci numbers, John Brillhart provided the author with the remaining 174 integers. Eighty-four of these are factors of the Fibonacci numbers U_n ($n = 1, 2, 3, \ldots, 1000$), where $U_{m+1} = U_m + U_{m-1}$ and $U_0 = U_1 = 1$, and 90 are factors of the Lucas numbers V_n ($n = 1, 2, 3, \ldots, 500$), where $V_{m+1} = V_m + V_{m-1}$ and $V_0 = 2$, $V_1 = 1$. These numbers are between 41 and 80 digits in length and were known to have no divisors less than 2^{32} and no algebraic factors.

TABLE 1

b	m	b, mL	b, mM
2	$4k-2$	$2^{2k-1} - 2^k + 1$	$2^{2k-1} + 2^k + 1$
3	$6k-3$	$3^{2k-1} - 3^k + 1$	$3^{2k-1} + 3^k + 1$
5	$10k-5$	$5^{4k-2} - 5^{3k-1} + 3 \cdot 5^{2k-1} - 5^k + 1$	$5^{4k-2} + 5^{3k-1} + 3 \cdot 5^{2k-1} + 5^k + 1$
6	$12k-6$	$6^{4k-2} - 6^{3k-1} + 3 \cdot 6^{3k-1} - 6^k + 1$	$6^{4k-2} + 6^{3k-1} + 3 \cdot 6^{3k-1} + 6^k + 1$
7	$14k-7$	$7^{6k-3} - 7^{5k-2} + 3 \cdot 7^{4k-2} - 7^{3k-1}$ $+3 \cdot 7^{2k-1} - 7^k + 1$	$7^{6k-3} + 7^{5k-2} + 3 \cdot 7^{4k-2} + 7^{3k-1}$ $+3 \cdot 7^{2k-1} + 7^k + 1$
10	$20k-10$	$10^{8k-4} - 10^{7k-3} + 5 \cdot 10^{6k-3}$ $-2 \cdot 10^{5k-2} + 7 \cdot 10^{4k-2}$ $-2 \cdot 10^{3k-1} + 5 \cdot 10^{2k-1}$ $-10^k + 1$	$10^{8k-4} + 10^{7k-3} + 5 \cdot 10^{6k-3}$ $+2 \cdot 10^{5k-2} + 7 \cdot 10^{4k-2}$ $+2 \cdot 10^{3k-1} + 5 \cdot 10^{2k-1}$ $+10^k + 1$
11	$22k-11$	$11^{10k-5} - 11^{9k-4} + 5 \cdot 11^{8k-4}$ $-11^{7k-3} - 11^{6k-3} + 11^{5k-2}$ $-11^{4k-2} - 11^{3k-1} + 5 \cdot 11^{2k-1}$ $-11^k + 1$	$11^{10k-5} + 11^{9k-4} + 5 \cdot 11^{8k-4}$ $+11^{7k-3} - 11^{6k-3} - 11^{5k-2}$ $-11^{4k-2} + 11^{3k-1} + 5 \cdot 11^{2k-1}$ $+11^k + 1$
12	$6k-3$	$12^{2k-1} - 2^{2k-1}3^k + 1$	$12^{2k-1} + 2^{2k-1}3^k + 1$

The results of running the programs on these numbers are given in Tables 2, 3, and 4. As is done in [2] we use the notation $b, m-$ and $b, m+$ to denote the numbers $b^m - 1$ and $b^m + 1$. The notation b, mL and b, mM for aurifeuillians is more complicated. We give their values (taken from [2]) in Table 1.

Note that b, mL and b, mM are factors of $b^m + 1$ for $b = 2, 3, 6, 7, 10, 11, 12$ and $5, mL$ and $5, mM$ are factors of $5^m - 1$.

In the first column of Tables 2, 3, and 4, we give the number which the composite integer N divides. In the second column, we give the number of decimal digits in N. In the third and fourth columns, we give the prime factors of N found by the computer program. A factor followed by an 'E' is one which was found by using Step 2 of the appropriate algorithm. An asterisk (*) in the first column is used to denote the fact that once the prime factors found in columns 3 and 4 had been divided into N, the remaining cofactor of N is prime; hence, we have a complete factorization of the number in column 1. Primality of these numbers was established by using the program described in Williams and Judd [9]. Two asterisks (**) in the first column indicate that this cofactor of N, while composite, was subsequently factored by M. Wunderlich using the continued fraction method of Morrison and Brillhart [7]. It should be noted that the factors found here for $10,65-$ and $10,69-$ were found independently by G. J. Stevens in South Australia. He also used the $p-1$ method.

TABLE 2

N divides	D	Factor(s) found by p-1 method	Factor(s) found by p+1 method
** 2, 173+	46	47635010587	
2, 197+	59	197002597249	
** 2, 209-	54	94803416684681	
2, 235+	56		328006342461
* 2, 265-	52	197748738449921	
** 2, 291+	54	5636963037465601E	
* 2, 297+	44		6215074747201E
* 2, 298M	42	14641916303149E	
** 2, 309+	44	2400744384937	
** 2, 351-	52	571890896913727	
* 2, 363-	56	75824014993	
* 2, 394L	46	152874915601	
* 2, 410L	49	61213422340181	
** 2, 418L	54		8857714771093
* 2, 442M	49	2291059412513	
* 2, 458L	59	84948746297, 6211454306149	
* 2, 458M	50	44185520789894155033573E	
** 2, 470M	55	87255998201	
* 2, 480+	58	137603804161	
* 2, 482M	59	76119208744309	
* 2, 558M	54	775844757937	
** 2, 602M	55		236344687097
** 2, 610M	55	1621474400951381	
** 2, 642M	50		87251820842149E
** 2, 654M	54	1193312900149	
* 2, 750M	48	168069194932501	
* 2, 774M	59	14512828061449	
** 2, 870L	55	4431960464101E	
** 3, 134+	50		719571227339189
* 3, 136+	46	2670091735108484737	

TABLE 2 (*continued*)

N divides	D	Factor(s) found by p-1 method	Factor(s) found by p+1 method
* 3, 161+	57		5468575720021E
* 3, 183-	47	2421854958301	
* 3, 185-	43	87841814842081	49804972211E
* 3, 231-	54	73155606217	
* 3, 303L	48	6024412974817	
** 3, 327L	48	262434507271	
** 3, 387L	54	18456700293426547E	
* 5, 82+	44	1148205782281	
* 5, 86+	57	2171388367013E	
* 5, 94+	57		329573417220613E
* 5, 117+	46	43236180703	
** 5, 126+	51	4661402165281	
* 5, 141+	49	37516308093487	
5, 180+	58	356646293281	
* 5, 205L	53	1256950067521	
* 5, 245M	59	1665032891036614953147l	
** 6, 59-	46		4866979762781
* 6, 71+	50	11735704315681	
** 6, 73-	57	2436094907761	
** 6, 111-	48		187333846633
** 6, 119-	60	103198889691409	
** 6, 132+	49		332526664667473E
** 6, 141-	57	122320721569	
* 6, 162M	43	39661919912737E	
** 6, 222M	52	63717427974558037	
** 6, 270L	49	51353541541	
** 6, 270M	53	159594687181	
7, 115-	60	723461377501	
** 7, 117+	52	5075833207537	
* 7, 132+	59	98138029441	
** 7, 133L	47		265043186297E
* 7, 175L	48	8230203760252601	
** 7, 231L	48		207734163253
*10, 65-	48	162503518711	
*10, 69-	42	203864078068831E	
*10, 80+	51	947147262401	
**10, 87+	57	638453709757E	
*10, 95+	60	121450506296081	
*10, 102+	44		225974065503889
*11, 59-	58	70845409351	
**11, 59+	57	5319902584128112849915З	
**11, 67+	52	2778466094669	
*11, 84+	42	70107576001	
**11, 231M	55	130958161489	
*12, 61+	50	5188602220069	
*12, 81-	52		660198074531409E
**12, 85-	55	204560684821	
**12, 87-	55	74233562929	
12, 183M	60		563215815517E
*12, 231L	52	161409762520777	

Factors were found for slightly over one quarter (134) of the 497 numbers tested. Most of these factors were found by the $p - 1$ method (112 vs. 32). This is what we would expect since (i) the $p - 1$ method was used first and (ii) for the numbers in Table 2 there is a built-in bias toward the success of the $p - 1$ test. This is because any prime divisor of $b^m - 1$ which does not divide any algebraic factor of $b^m - 1$ must be of the form $km + 1$.

TABLE 3

N	D	Factor(s) found by p-1 method	Factor(s) found by p+1 method
$*U_{247}$	46		409100738617
$*U_{307}$	55	5307027867738937	
$*U_{313}$	59		7901346123803597
$*U_{323}$	52	85542646443577	
$*U_{343}$	57	5449038756620509	
U_{361}	72	6567762529, 1196762644057	
$*U_{365}$	48	758275080626801	
U_{367}	59	5648966761	43397676601
U_{377}	71	361575655741	
$*U_{387}$	45		14279673833
$*U_{403}$	65	42136290591640129	
U_{411}	51	972663078773E	
$*U_{421}$	75	45688564527041	
U_{455}	55	36768087721	
U_{459}	57	2043118036369	
$*U_{465}$	42	6936488411701, 59666387254501	
$*U_{483}$	56		1795220677069
U_{485}	74	16892304192301, 511715857773521	
$*U_{495}$	51		1250839826281
U_{507}	63	10069148777	
$*U_{531}$	63		2192843129417
$*U_{549}$	67	5883010433, 80256319951861	10424083697
U_{555}	53	49649320649221	
U_{567}	68	49114912141, 3936504300121	
U_{591}	79	22221540969737	
U_{595}	73	8310112721, 9022425301	
U_{633}	73	41773163881	
$*U_{675}$	70	6641555895901	
U_{765}	77	72208475461	

The real problem with the $p+1$ test is the fact that it is quite slow. For our program we found that it was about nine times slower (when used three times for three different trials at P_0 value) than the $p-1$ test. Thus, one should probably use a higher bound for B_1 or B_2 for the $p-1$ test than for the $p+1$ test. We remark here, however, that if we had increased the $\max(B_1, B_2)$ to 10^7, the $p-1$ test would very likely have found nine of the 32 factors found here by the $p+1$ test. This is because each of the remaining numbers p is such that a prime which exceeds 10^7 divides $p-1$.

TABLE 4

N	D	Factor(s) found by p-1 method	Factor(s) found by p+1 method
$*V_{254}$	50		347366417511089201
$*V_{271}$	46	92206663291	
$*V_{283}$	56	252605941501	
V_{299}	46	143236388738249	
V_{302}	63	70963651961	
V_{304}	45	1244124101 7224321	
V_{331}	64	54184296181	
V_{338}	57		404112157123
$*V_{346}$	72	685204772026924 67E	
$*V_{352}$	67	389132418765025 6896001	
V_{358}	75	316590102769	
V_{367}	71		19997474011
$*V_{369}$	48	26024651929	18736753266019E
V_{374}	59	3827019260681	
$*V_{384}$	42		1769526527
$*V_{390}$	41	54975368761	
V_{406}	67	64690797641	
$*V_{413}$	67	33637840386809	
$*V_{418}$	59	722601451307	
V_{419}	79	316722762859	
V_{426}	55		1006118006507
$*V_{428}$	74	386610981607	
$*V_{460}$	69	28677143808961	
$*V_{469}$	70	10812055185331	
$*V_{477}$	66	49721203549E	6430515046741

6. Acknowledgements. The author wishes to thank John Brillhart and Sam Wagstaff for making available to him the tables referred to in Section 5.

Department of Computer Science
University of Manitoba
Winnipeg, Manitoba, Canada R3T 2N2

1. RICHARD P. BRENT, "The first occurrence of certain large prime gaps," *Math. Comp.*, v. 35, 1980, pp. 1435–1436.

2. JOHN BRILLHART, D. H. LEHMER, J. L. SELFRIDGE, BRYANT TUCKERMAN & S. S. WAGSTAFF, JR., *Factorizations of* $b^n \pm 1$, $b = 2, 3, 5, 7, 10, 11, 12$ *Up to High Powers.* (To appear.)

3. RICHARD K. GUY, "How to factor a number," *Congressus Numerantium* XVI, Proc. Fifth Manitoba Conf. on Numerical Math., Winnipeg, 1976, pp. 49–89.

4. DONALD E. KNUTH, *The Art of Computer Programming*, Vol. II, *Seminumerical Algorithms*, 2nd ed., Addison-Wesley, Reading, Mass., 1981.

D. H. LEHMER, "An extended theory of Lucas' functions," *Ann. of Math.* (2), v. 31, 1930, pp. 419–448.

6. D. H. LEHMER, "Computer technology applied to the theory of numbers," in *Studies in Number Theory* (W. J. LeVeque, ed.), Math. Assoc. Amer. Studies in Math., vol. 6, 1969, pp. 117–151.

7. MICHAEL A. MORRISON & JOHN BRILLHART, "The factorization of F_7," *Bull. Amer. Math. Soc.*, v. 77, 1971, p. 264.

8. J. M. POLLARD, "Theorems on factorization and primality testing," *Proc. Cambridge Philos. Soc.*, v. 76, 1974, pp. 521–528.

9. H. C. WILLIAMS & J. S. JUDD, "Some algorithms for prime testing using generalized Lehmer functions," *Math. Comp.*, v. 30, 1976, pp. 867–886.