$43 \cdot 86 \equiv 40 \ mod\ 1829$, we have found only a trivial relationship. Thus, we have to look for another subset of rows which sum to a row of even numbers. We notice that the sum of the first three rows and the fifth row is $2\ 2\ 2\ 2\ 2\ -\ 2$, and this gives the congruence $(42 \cdot 43 \cdot 61 \cdot 85)^2 \equiv (2 \cdot 3 \cdot 5 \cdot 7 \cdot 13)^2 \ mod\ n$, i.e., $1459^2 \equiv 901^2 \ mod\ 1829$. We conclude that a factor of 1829 is $g.c.d.(1459 + 901, 1829) = 59$.

Factor base algorithm. We now summarize a systematic method to factor a very large n using a *random* choice of the b_i. Choose an integer y of intermediate size, for example, if n is a 50-decimal-digit integer, we might choose y to be a number with 5 or 6 decimal digits. Let B consist of -1 and all primes $\leq y$. Choose a large number of random b_i, and try to express $b_i^2 \ mod\ n$ (least absolute residue) as a product of the primes in B. Once you obtain a large quantity of B-numbers $b_i^2 \ mod\ n$ ($\pi(y) + 2$ is enough, where $\pi(y)$ denotes the number of primes $\leq y$), take the corresponding vectors in $\mathbf{F}_2^h$ (where $h = \pi(y) + 1$) and by row-reduction determine a subset of the b_i whose corresponding $\vec{\epsilon}_i$ sum to zero. Then form $b = \prod b_i \ mod\ n$ and $c = \prod p_j^{\gamma_j} \ mod\ n$, as described above. Then $b^2 \equiv c^2 \ mod\ n$. If $b \equiv \pm c \ mod\ n$, start again with a new random collection of B-numbers (or, to be more efficient, choose a different subset of rows in the matrix of $\vec{\epsilon}$'s which sum to zero, if necessary finding a few more B-numbers and their corresponding rows). When you finally obtain $b^2 \equiv c^2 \ mod\ n$ and $b \not\equiv \pm c \ mod\ n$, compute $g.c.d.(b + c, n)$, which will be a nontrivial factor of n.

Heuristic time estimate. We now give a very rough derivation of an estimate for the number of bit operations it takes to find a factor of a *very* large n using the algorithm described above. We shall use several simplifying assumptions and approximations, and in any case the result will only be a probabilistic estimate. If we are very unlucky in our random choice of b_i, then the algorithm will take longer.

We shall need the following preliminary facts:

Fact 1 (*Stirling's formula*). $log(n!)$ is approximately $n \log n - n$.

By "approximately," we mean that the difference grows much more slowly than n as $n \longrightarrow \infty$. This can be proved by observing that $log(n!)$ is the right-endpoint Riemann sum (with endpoints at $1, 2, 3, \ldots$) for the definite integral $\int_1^n \log x\, dx = n \log n - n + 1$.

Fact 2. Given a positive integer N and a positive number u, the total number of nonnegative integer N-tuples α_j such that $\sum_{j=1}^N \alpha_j \leq u$ is the binomial coefficient $\binom{[u]+N}{N}$.

Here [ ] denotes the greatest integer function. Fact 2 can be proved by letting each N-tuple solution α_j correspond to the following choice of N integers β_j from among $1, 2, \ldots, [u] + N$. Let $\beta_1 = \alpha_1 + 1$, and for $j \geq 1$ let $\beta_{j+1} = \beta_j + \alpha_{j+1} + 1$, i.e., we choose the β_j's so that there are α_j numbers between β_{j-1} and β_j. This gives a 1-to-1 correspondence between the number of solutions and the number of ways of choosing N numbers from a set of $[u] + N$ numbers.