NUMBERS COUNT

How long is a farey tale? This month Mike Mudge examines the construction and use of Farey Sequences.

Definition The Farey sequence, F_n, of order n is the ascending sequence of irreducible fractions between 0 and 1 whose denominators do not exceed n; the numbers 0 and 1 are included in the forms 0/1 and 1/1. For example, F_5 is: 0/1, 1/5, 1/4, 1/3, 2/5, 1/2, 3/5, 2/3, 3/4, 4/5, 1/1.

Notes:

(i) The fraction 1/2 is the middle term of every Farey Sequence.

(ii) Two terms equidistant from the middle have a sum of unity. This property can be readily used to economise on the presentation of F_n, thus we write F_5:

0 1 1 1 2 1
1 5 4 3 5 2
1 4 3 2 3 1

(iii) Clearly if m is any number smaller than n, the sequence F_m consists simply of those terms of F_n whose denominators are not larger than m, in unchanged order.

(iv) The sequence F_n exhibits, incidentally, for every value of m not larger than n, the sequence of numbers prime to m and not larger than n. The numbers prime to m and smaller than m are in ascending order the numerators which have m for denominator, while the numbers prime to m and larger than m are in descending order the denominators which have m for numerator.

Theorem The total number of terms in F_n is PHI(n) + 1 where PHI(n) is the sum-function of Euler's Totient Function phi(n). For definitions of these functions see 'Numbers Count' May 1985.

Historical Notes Farey sequences are implicit in Henry Goodwyn's *A Tabular Series of Decimal quotients* (London, 1823) and an earlier volume by the same author (1818); also in tables published by Achille Brocot in Paris in 1862; these are of order 100.

This value was not passed until 1935 when E Buckingham's *Manual of Gear Design* included an unreliable and incomplete table of F_{120}. This was corrected in RM Page's *14000 Gear Ratios* (1942); however, the 'ultimate' table of F_{1025} was designed and compiled by EH Neville and published for The Royal Society by The University of Cambridge Press in 1950. Here the 319765 terms are elegantly presented 400 to a page, 20 to a line.

Problem A Write a computer program to generate and display in a meaningful manner F_n, for a given n. Warning: do *not* attempt to reproduce Neville's table in full, although computation time would be an interesting parameter.

Problem B Devise and implement an efficient routine for locating a given term in F_n.

Note: Except for small values of n, the fractions 1/b, 2/b, 3/b, ... (b–1)/b divide F_n into subsequences, almost all of which contain approximately the same number of terms.

Would your routine be suitable for 'manual' implementation given printed tables?

Problem C Devise and implement a routine for inserting between two terms of a given Farey Sequence a batch of terms belonging to a Farey Sequence of a higher order.

Investigate the use of such a routine to generate rational approximations to say 'pi' or 'e'.

Problem D Observing that the application of Farey Sequences to the solution of the linear Diophantine Equation $Bx - ay = 1$ depends upon the result that if a/b and c/d are consecutive terms in F_n, then $bc - ad = 1$, and conversely that if $bc - ad = 1$, then a/b and c/d are consecutive in F_n for all values of n from max (b,d) to b+d−1, investigate the use of 'in-store' Farey Sequences to solve such equations.

Readers are invited to submit their attempts at some (or all) of the above problems to Mike Mudge, 'Square Acre', Stourbridge Road, Penn, Nr Wolverhampton, Staffordshire WV4 5NF. Tel: (0902) 892141.

It would be appreciated if such submissions could contain a brief summary of results obtained and thoughts relating to the problem in a form suitable for future publication in *PCW*. Submissions, which must reach me by 1 November 1986, will be judged using suitably vague criteria, and a prize will be awarded to the 'best' contribution received by the closing date.

Please note that submissions can only be returned if a suitable stamped addressed envelope is provided. Expanded reviews of previous problems, together with, subject to the approval of the contributor, copies of detailed programs from the winning entry may also be requested. However, in the interests of efficiency, interested readers are encouraged to contact the prize-winner directly. Mike Mudge welcomes correspondence on any subject within the areas of number theory and other computationally related mathematics, particularly containing suggested subject areas for future Numbers Count Articles, and will endeavour to reply to all letters.

February review

Those readers who were frightened off by the mathematics of primitive roots and indices are referred to *Elementary Number Theory* by Allan M Kirch, Intext Educational Publishers 1974, where some simple Basic Programs are given to find, for example, the smallest primitive root of a given odd prime, and then to generate a table of indices and anti-indices.

Tables suitable for checking the correctness of routines are to be found in *An Introduction to the Theory of Numbers* by IM Vinogradov, Pergamon 1955; the theory of primitive roots is developed historically in LE Dickson's *History of the Theory of Numbers* volume 1, Chelsea, New York 1952, while some difficult mathematical examples are to be found in *Exercises in Number Theory* by DP Parent, Springer-Verlag 1984.

Responses to this problem included several very detailed and extremely well-presented pieces of work, but after careful consideration the prize-winner is Dr John HE Cohn of 23 Highfield Gardens, London NW11 9HD.

John chose to write in Basic on his PET in preference to his Amstrad 8256 which he believes has certain bugs associated with its Basic.

Following the input of N, checked to be a positive integer less than 10000 and of the required form, the program then offers the following choices:

(1) to find all primitive roots;
(2) to find the least primitive root only;
(3) to find a table of indices wrt the least primitive root; and
(4) to find a table of indices wrt any supplied primitive root.

John's routines should run on almost any machine given adequate memory; about 23k is needed in the worst case, the conscious decision being taken to save execution time by using memory to avoid long sorting routines; hence the restriction N less than 10000.