User's Manual To Accompany "RAP" Rational Approximation Software

David T. Ashley, (DTASHLEY@AOL.COM)
Joseph P. DeVoe (JDEVOE@VISTEON.COM)
Cory Pratt (CORY_PRATT@3COM.COM)
Karl Perttunen (KPERTTUN@VISTEON.COM)
Anatoly Zhigljvasky (ZHIGLJAVSKYAA@CARDIFF.AC.UK)
David Eppstein (EPPSTEIN@ICS.UCI.EDU)
David M. Einstein (DEINST@WORLD.STD.COM)

10/18/2000

TABLE OF CONTENTS

1. Introduction And Overview

1.1 Scope Of This Document

This manual accompanies the rational approximation software program submitted to CALGO (the ACM collected algorithms), in support of a paper entitled "*On Best Rational Approximations Using Large Integers*" submitted to TOMS (ACM Transactions On Mathematical Software) in late 2000. The software program is named RAP (mnemonic for *r*ational *ap*proximation), and this acronym will be used throughout this document to refer to the software program. This document explains the operation and limitations of RAP.

1.2 Overview Of RAP Functionality

RAP implements a set of useful algorithms on very large integers and rational numbers with large integer components. RAP implements operations on long integers by representing all integers as character strings, and performing multiplication, division, and other primitive operations using the same "long-hand" techniques taught in elementary school.

RAP is able to implement the following operations on integers and rational numbers. Each distinct operation includes the command code for the operation used to specify it to RAP, in square brackets. Each operation is explained in detail later in the manual.

- **Integer Sum [+].**
- **Integer Difference [-].**
- **Integer Product [*].**
- **Integer Quotient [/].**
- **Integer Remainder [%].**
- **Integer Raised To An Integer Power [**].**
- **Greatest Common Divisor [GCD].**
- **Decimal Approximation [DAP].**
- **Rational Number Sum [+].**
- **Rational Number Difference [-].**
- **Rational Number Product [*].**
- **Rational Number Quotient [/].**
- **Rational Number Raised To An Integer Power [**].**
- **Continued Fraction Partial Quotients And Convergents Of A Rational Number [CF].**
- **Rational Number With The Smallest Denominator In An Interval [MIND].**
- **Enclosing Rational Numbers In The Farey Series Of Order N [FN].**
- **Upper Bound On Distance Between Farey Terms In An Interval [FNDMAX].**
- **Enclosing Rational Numbers In A Rectangular Area Of The Integer Lattice [FAB].**
- **Upper Bound On Distance Between Members Of $F_{A,B}$ In An Interval [FABDMAX].**

1.3 RAP Absolute Maximums

1.3.1 Maximum Data Sizes

To avoid confusion in the discussion which follows, integers which are in binary format and are primitive data types in the ‘C’ programming language are called *native* integers; and integers which are maintained by RAP as long strings of decimal digits are called *synthetic* integers.

RAP is designed to freely operate within the Farey series of up to order 2^{1536}. (2^{1536} is about 2.4×10^{462}.) Because RAP manipulates synthetic integers as decimal character strings, the limit of 2^{1536} is enforced by prohibiting synthetic integers as input which exceed 463 digits (meaning that RAP will actually accept synthetic integers up to and including 10^{463}-1, a number somewhat larger than 2^{1536}.). Additionally, no internal intermediate result or output result in RAP may exceed 4,000 decimal digits.

The limits of 2^{1536} and 4,000 were chosen out of convenience rather than necessity. The absolute maximums are specified as compile-time preprocessor constants, and allow RAP to operate more efficiently (and allow the code to be more compact) than if internal data structures were allowed to grow arbitrarily. In RAP, *every* synthetic integer used for applying the algorithms presented in the TOMS paper has 4,000 digits when it is created—it is never necessary to dynamically resize the storage allocated for such a synthetic integer. Additionally, the limit of 4,000 keeps RAP comfortably clear of platform-specific data size limits for native integers used to index into strings—the ANSI ‘C’ standard guarantees that an unsigned native integer variable can attain at least 65,535 on any platform. RAP can be recompiled with larger upper limits, but this is neither recommended nor supported.

If the data size limits of RAP are exceeded, RAP will announce the error and abort gracefully. There is no known circumstance where RAP will give incorrect results due to an undetected synthetic integer size overflow.

1.3.2 Maximum Length Of Standard Input Stream

If RAP is used in batch mode (described later), the standard input stream must not exceed 31,999 characters before RAP encounters the end-of-file (this is because RAP buffers the input stream before processing it). If this maximum is violated, RAP will announce the error and gracefully abort.

1.3.3 Maximum Number Of Farey Neighbors Generated

RAP will generate no more than 10,000 Farey neighbors on either side of a rational number. This limit is enforced to avoid platform-specific issues with native integers if RAP is ported.

1.4 RAP Authors And Algorithm Authors

The RAP software was developed exclusively by Dave Ashley. However, the algorithms that RAP applies were developed by all of the contributors listed below. Note that David Eppstein and David M. Einstein are not authors on the TOMS paper—via the `sci.math` newsgroup, they contributed approaches for finding a rational number with the smallest denominator within an interval. All of the contributors listed below should be listed as authors for the algorithms in CALGO.

Algorithm Author	E-mail Address	Home Page
David T. Ashley	dtashley@aol.com	N/A
Joseph P. DeVoe	jdevoe@visteon.com	N/A
Cory Pratt	cory_pratt@3com.com	N/A
Karl Perttunen	kperttun@visteon.com	N/A
Anatoly Zhigljavsky	zhigljavskyaa@cardiff.ac.uk	http://www.cf.ac.uk/maths/zhigljavskyaa/
David Eppstein	eppstein@ics.uci.edu	http://www.ics.uci.edu/~eppstein/
David M. Einstein	deinst@world.std.com	N/A

1.5 Statement Of Reliability And Intended Use

RAP is a research-grade tool, and is intended only to demonstrate for research purposes the algorithms presented in the TOMS paper. RAP has not been subjected to standard software engineering practices which help to minimize the probability of defects, such as unit-testing or peer review. RAP is research-grade, intended only to illustrate concepts, and absolutely unsuitable for any application where injury, loss of life, or financial losses could occur because of improper operation of the software. RAP is provided "as-is" with no warranty of any kind.

1.6 Embedded Version Control Information

For convenience, both this document (in the footer) and the RAP software (information emitted on every invocation) contain embedded version control information, which is inserted automatically by a version control tool. Versions of this document which have different version control information in the footer are necessarily different. Executable copies of RAP which emit different version control information when the software is invoked are necessarily different.

This document and the RAP program are version-controlled separately. This means that this document generally won't have the same revision number as the RAP executable, and there is no way to make the association between document version and RAP executable version. However, the most current versions of each (i.e. the versions which "belong" together) are distributed through CALGO.

1.7 Bug Reports

The probability of finding a bug in the operation of the RAP program is very low. Because RAP performs a small number of repetitive operations on large data (adding digits and processing carries, for example), RAP does not fit the profile of software which is good at concealing defects. Defects which would become apparent with large data would generally also become apparent with small data, and the program was exercised with much small data.

Please submit all bug reports to Dave Ashley (and the other contributors if you have difficulty in reaching Dave). If the bug can be reproduced, it will be fixed and RAP will be re-released to CALGO.

2. Portability Of RAP Source Code

RAP is provided to CALGO with a binary executable for Win32-capable platforms (this should include Windows 95, Windows 98, Windows ME, Windows NT, and Windows 2000). Use of RAP on any of these platforms should not require recompilation because the binary executable is already available. This section contains portability notes in the event RAP must be compiled for other platforms.

RAP is written in ANSI ‘C’, in a single software module. Because only one software module is used, no header files are included. Enumerated below are requirements on any target system and issues which may affect portability favorably or unfavorably.

- RAP was compiled for Win32 as a Win32 console application under Microsoft Visual C++ Version 6.0. The file RAP_C.C contains a function named *main_c()*, which is the main function of the program, called from a C++ wrapper in another file. For recompilation on another platform, only the file RAP_C.C is required, and the function *main_c()* must be renamed to *main()*.
- The only known possible portability issue with RAP is the use of the preprocessor symbol “__LINE__”, which is used with a home-made assertion mechanism to emit the line number where an assertion fails. If “__LINE__” won’t compile, it would be necessary to either find the name of a similar symbol for the target system, or else replace all occurrences of __LINE__ with a constant integer, such as 0.
- The C++ record oriented comment delimiter “//” is not used, although many ‘C’ compilers allow its use. Only “/*” and “*/” are used.
- Nested comments are not used, because not all compiler support them.[1]
- RAP assumes the existence of the standard dynamic memory allocation functions *malloc()*, *realloc()*, and *free()*. *malloc()*, *realloc()*, and *free()* must work correctly on the target system.
- For command-line parameters, the target system must support *argv* and *argc* in the standard way.
- Both signed and unsigned integers in the target system must be at least 16 bits, and must be able to attain a positive value of at least 32,100.
- On input in batch mode, the *getchar()* family of functions is used to read from the standard input stream. The target system must support these.
- On output, the *printf()* family of functions is used. The target system must support these.
- The character set of the target machine must have the digits ‘0’ through ‘9’ as contiguous and increasing (RAP performs some calculations making this assumption—this is a nearly universal feature of most character sets).
- RAP performs no console manipulation and does not write to *stderr* (this should increase its portability).
- RAP should port to Unix systems with no changes whatsoever, except the possible issue with __LINE__ noted above.

3. Invoking RAP And Specifying Commands

3.1 RAP Invocation Modes

RAP operates in two different modes, depending on the command line parameters when RAP is invoked.

- **Interactive Mode.** If all of the information that RAP requires to perform an individual command is present on the command line, RAP will perform the command and exit. This allows RAP to be used as an interactive calculator to answer impromptu queries.

  For example, invoking RAP with the command line:

[1] A nested comment is multiple occurrences of “/*” before a matching “*/”. For example, “/* /* */ */” is a nested comment. A compiler that tolerates nested comments is sometimes helpful because it allows large blocks of code containing comments to be removed from the compilation stream by enclosing the block in “/*” and “*/”.

rap fn 3.14159265359 255

will cause RAP to provide the two best rational approximations to π with a maximum denominator of 255.[2]

On a Win32 system, invoking RAP in interactive mode will require opening a DOS shell and then invoking the RAP program.

- **Batch Mode.** If RAP is invoked with the single parameter "BATCH", it will take input from the standard input stream until end-of-file on the stream, treating each token in the same way it would a command-line parameter. The batch mode allows numbers to be specified which are too long for the operating system to tolerate on the command-line.

  For example, preparing an input file named RAP_IN.TXT with the following contents and invoking RAP with the command line "`rap batch <RAP_IN.TXT`" will give the same results as the command-line invocation described above.[3]

  ```
  fn
  3.14159265359
  255
  ```

  It is also noteworthy that some computing platforms can be "tricked" into accepting input from the keyboard which is longer than allowed on a command line. For example, on a Win32 platform, the invocation:

  ```
  rap batch | more
  ```

  will allow input to be entered from the keyboard (stdin) which is much longer than allowed on the command line, but the input must be terminated with CONTROL-Z. The "`| more`" pipelining command causes all output to go to a file before it is displayed, and eliminates display problems due to intermingling of input and output.

Invoking RAP with no parameters will result in a help message. Supplying RAP with unexpected input will result in an error message.

[2] This statement should be qualified somewhat. RAP will not operate on irrational numbers—every number which RAP can accept as input is rational. When using RAP to find rational numbers which are near an irrational, a rational approximation of the irrational must be used (in this case, 3.1415926539 for π). RAP will provide the terms of F_{255} which enclose 3.1415926536, which are not required to be the rational numbers which enclose π. RAP will provide an incorrect result if and only if there is a term of the Farey series of interest in between the irrational number one is trying to approximate and the rational approximation supplied to RAP. The probability of one getting "unlucky" in this way increases with a)increasing order of the Farey series, because the terms are, on average, closer together, or b)decreasing precision of the rational number specified to RAP. It is tedious to enter large numbers of digits for irrational numbers: commercial symbolic manipulation software such as *Mathematica* may provide the ability to calculate arbitrarily many partial quotients of irrational numbers, and so may be more convenient in this regard than RAP.

[3] On a command line, with many operating systems, "<" will cause the standard input for the program to be taken from the specified file.

3.2 Token Concatenation

When preparing an input file for use in the batch mode of RAP, it may occur that a number to be supplied to RAP exceeds a convenient length for a line of text. For this reason, RAP treats the backslash ("\") as a token concatenation operator. RAP first parses its input, identifying tokens (groups of letters, digits, and symbols separated by spaces, tabs, and newline characters), then tokens ending in the backslash character are concatenated with the following token. Most readers of this document will have substantial experience with programming languages and scripting languages, so this concept doesn't need further explanation.

For example, the input file RAP_IN.TXT specified in the example above could be prepared as shown below with no change in RAP's behavior.

```
f\
n
3.14\ 1592\ 65\
359
255
```

Token concatenation is also applied in interactive mode, although this is probably of no practical value.

3.3 Case Sensitivity Of RAP

RAP is case-insensitive in all input.

However, an operating system may be case-sensitive. In the example above involving batch mode, the command line "`rap batch <rap_in.txt`" might not work correctly, as an operating system may treat `RAP_IN.TXT` as a different file than `rap_in.txt`. It is the operating system, not RAP, which arranges for redirection of the standard input and output.

3.4 Comments In Batch Mode

RAP has no provision for comments in the input file when used in batch mode.

3.5 Saving RAP Output To A File

All RAP output is written to the standard output stream, which is the console by default. To redirect RAP output to a file, include "`>filename`" on the command line (Win32 DOS boxes and Unix). Output may be concatenated to a file using "`>>filename`" (Win32 DOS boxes and Unix). Output may also be displayed a screen at a time using "`| more`".

3.6 Error Behavior Of RAP

RAP will respond to errors in five different ways.

- **Input parsing errors.** Some feature of the input command or data violates RAP's parsing rules (illegal characters, ill-formed numbers, etc.). The diagnostic messages provided in these cases should be adequate to locate the problem.
- **Command Template Matching Errors.** RAP contains an internal "template" table which is a list of all commands and the types of arguments that each will accept. If RAP can parse all input parameters, but can't match the number and type of parameters to the requested command, RAP will issue a nebulous catchall "template matching" error message. For example, "`rap gcd 2 2/3`" will fail because the "gcd" command won't accept a non-

integer argument. Similarly, "`rap 2 3 4`" will fail because the "gcd" command can only accept two arguments.

- **Context-Sensitive Error Messages.** There are a small number of situations where RAP will issue specific error messages. In these cases, the information provided in the error message should be adequate to understand the nature of the error.
- **NAN.** Any internal calculation in RAP which overflows or produces an undefined result will cause the value of NAN (*N*ot *A N*umber) to be assigned to the result. Any operation with NAN as an input will produce NAN as an output (NANs propagate). For example, exponentiating 1000000 to the 1000th power ("rap ** 1000000 1000") will produce NAN.
- **Assertion Failures.** The source code of RAP is liberally decorated with calls to the function *asAssert()*, which behaves similarly to the standard C-language *assert()* macro. If an assertion fails, it represents a serious internal software error. Only a line number will be displayed. The source code must be consulted to determine the nature of the error. (An assertion failure is a bug—please report any such failures as bugs.)

3.7 Specification Of Integers To RAP

An integer is specified as a series of digits 0-9, with an optional leading unary "-" sign, and optional commas.

- Commas in integers are simply discarded by RAP. Integers are not parsed to determine if the commas are placed correctly within the integer. Example: "1234", "12,34", "1,234", and "1,,2,,3,,4" are treated equivalently by RAP; each represents the integer 1,234.
- The integer *zero* must be specified as a single instance of the digit "0". Multiple instances are illegal. A unary "-" is illegal.
- Leading 0's on non-zero integers are not allowed.

3.8 Specification Of Rational Numbers To RAP

A rational number may be specified to RAP in two different ways.

- Two integers separated by a forward slash ("/"). Rational numbers may never contain whitespace. Only the numerator may contain an optional '-'-sign.
- A floating-point constant with or without a positive or negative exponent. For example, -3.121934204e-3 is legal and will be treated internally by RAP as the rational number -3,121,934,204/1,000,000,000,000.[4] RAP will also accept -0.003121934204 or -3121934204/1000000000000 and treat it equivalently.

RAP will not accept other intuitively plausible specifications of rational numbers. For example, RAP's parser will not accept 1e30/5.

[4] This isn't completely true. RAP will remove any g.c.d. from numerator and denominator before using a rational number internally.

4. Detailed Descriptions Of Commands

Commands are polymorphic in that RAP will differentiate between integers and rational numbers and may use slightly different algorithms or format results differently depending on whether input arguments are integers or rational numbers. Note that every integer is a rational number, so RAP will always accept integers where rational numbers are specified. RAP will also accept rational numbers where integers are required, if the rational number reduces to an integer.

4.1 Integer Sum [+]

4.1.1 Command Line Invocation Forms

rap + I1 I2

Adds integer I1 to integer I2 to produce an integer result.

4.1.2 Detailed Algorithm Description

Adds integers to produce an integer result. The algorithm applied is addition of ASCII digits.

4.1.3 Example Invocation

Output from the invocation of RAP to add two large integers is reproduced below.

```
C:\>rap + 3,142,991,002 7,934,333
-------------------------------------------------------------------------------
RAP ($Revision: 15 $ $Date: 11/13/00 3:59a $) execution begins.
-------------------------------------------------------------------------------
                arg1:                          3,142,991,002   (   10 digits)
-------------------------------------------------------------------------------
                arg2:                              7,934,333   (    7 digits)
-------------------------------------------------------------------------------
         arg1 + arg2:                          3,150,925,335   (   10 digits)
-------------------------------------------------------------------------------
RAP execution ends.
-------------------------------------------------------------------------------
```

4.2 Integer Difference [-]

4.2.1 Command Line Invocation Forms

rap - I1 I2

Subtracts integer I2 from integer I1 to produce an integer result.

4.2.2 Detailed Algorithm Description

Subtracts integers to produce an integer result. The algorithm applied is subtraction of ASCII digits.

4.2.3 Example Invocation

Output from the invocation of RAP to subtract two large integers is reproduced below.

```
C:\>rap - -343926469248723687426946 284622838352848
------------------------------------------------------------------------------
RAP ($Revision: 15 $ $Date: 11/13/00 3:59a $) execution begins.
------------------------------------------------------------------------------
                arg1: -     343,926,469,248,723,687,426,946   (   24 digits)
------------------------------------------------------------------------------
                arg2:                   284,622,838,352,848   (   15 digits)
------------------------------------------------------------------------------
         arg1 - arg2: -     343,926,469,533,346,525,779,794   (   24 digits)
------------------------------------------------------------------------------
RAP execution ends.
------------------------------------------------------------------------------
```

4.3 *Integer Product [*]*

4.3.1 Command Line Invocation Forms

rap * I1 I2

Multiplies integer I1 by integer I2 to produce an integer result.

4.3.2 Detailed Algorithm Description

Multiplies integers to produce an integer result. The algorithm used is essentially long-hand multiplication of ASCII digits (multiplication by a single digit, shifting, addition).

4.3.3 Example Invocation

Output from the invocation of RAP to multiply two large integers is reproduced below.

```
C:\>rap * -294328649236462394616946 826482348525
------------------------------------------------------------------------------
RAP ($Revision: 15 $ $Date: 11/13/00 3:59a $) execution begins.
------------------------------------------------------------------------------
                arg1: -     294,328,649,236,462,394,616,946   (   24 digits)
------------------------------------------------------------------------------
                arg2:                       826,482,348,525   (   12 digits)
------------------------------------------------------------------------------
         arg1 * arg2: -                         243,257,433,  (   36 digits)
                        259,142,387,965,858,847,843,104,650
------------------------------------------------------------------------------
RAP execution ends.
------------------------------------------------------------------------------
```

4.4 Integer Quotient [/]

4.4.1 Command Line Invocation Forms

rap / I1 I2

Divides integer I1 by non-zero integer I2 to produce an integer result.

4.4.2 Detailed Algorithm Description

Divides two integers to produce an integer quotient. The result produced is specified by the equation below. The algorithm is long-hand division of ASCII digits (shifting, trial subtraction).

$$\text{Result} = \left\lfloor \frac{I1}{I2} \right\rfloor$$

4.4.3 Example Invocation

Output from the invocation of RAP to divide two large integers is reproduced below. Note that the remainder is also supplied.

```
C:\>rap / 9274639462975692736497259623964932 287463864289
------------------------------------------------------------------------------
RAP ($Revision: 15 $ $Date: 11/13/00 3:59a $) execution begins.
------------------------------------------------------------------------------
            dividend:                             9,274,639,  (   34 digits)
                        462,975,692,736,497,259,623,964,932
------------------------------------------------------------------------------
             divisor:                       287,463,864,289   (   12 digits)
------------------------------------------------------------------------------
  dividend % divisor:                       275,260,681,237   (   12 digits)
------------------------------------------------------------------------------
  dividend / divisor:        32,263,670,725,762,915,010,255   (   23 digits)
------------------------------------------------------------------------------
RAP execution ends.
------------------------------------------------------------------------------
```

4.5 Integer Remainder [%]

4.5.1 Command Line Invocation Forms

rap % I1 I2

Divides integer I1 by non-zero integer I2 to produce an integer remainder.

4.5.2 Detailed Algorithm Description

RAP will divide two integers to produce an integer remainder (the modulo remainder function). The result produced is specified by the equation below. The algorithm employed is long-hand division of ASCII digits.

$$\text{Result} = I1 - I2\left\lfloor \frac{I1}{I2} \right\rfloor$$

4.5.3 Example Invocation

Output from the invocation of RAP to divide two large integers and thereby produce the integer remainder is reproduced below. Note that the quotient is also supplied.

```
C:\>rap % 987243769234692364923 23984236496
------------------------------------------------------------------------------
RAP ($Revision: 15 $ $Date: 11/13/00 3:59a $) execution begins.
------------------------------------------------------------------------------
            dividend:           987,243,769,234,692,364,923   (   21 digits)
------------------------------------------------------------------------------
             divisor:                        23,984,236,496   (   11 digits)
------------------------------------------------------------------------------
  dividend / divisor:                        41,162,192,901   (   11 digits)
------------------------------------------------------------------------------
  dividend % divisor:                         3,136,050,027   (   10 digits)
------------------------------------------------------------------------------
RAP execution ends.
------------------------------------------------------------------------------
```

4.6 Integer Raised To An Integer Power [**]

4.6.1 Command Line Invocation Forms

rap ** I1 I2

Raises integer I1 to the power of non-negative integer I2.

4.6.2 Detailed Algorithm Description

RAP will exponentiate a non-negative integer to a positive integral value, according to the equation below.

$$\text{Result} = I1^{I2}$$

The algorithm applied is to examine the bit pattern of the exponent and to repeatedly square the argument and to selectively multiply in the repeated square. For example, $I1^5$ can be rewritten as $(I1^4)(I1^1)$, and calculated using the following steps:

- multiplier := I1
- result := 1
- result := result * multiplier
- multiplier := multiplier * multiplier
- multiplier := multiplier * multiplier
- result := result * multiplier

4.6.3 Example Invocation

```
C:\>rap ** 117 54
-------------------------------------------------------------------------------
RAP ($Revision: 15 $ $Date: 11/13/00 3:59a $) execution begins.
-------------------------------------------------------------------------------
                 arg:                                    117   (    3 digits)
-------------------------------------------------------------------------------
            exponent:                                     54   (    2 digits)
-------------------------------------------------------------------------------
     arg ** exponent:                                  4,808,  (  112 digits)
                         797,999,524,921,631,067,632,458,892,
                         309,152,880,430,748,856,242,309,594,
                         970,691,360,198,993,667,007,419,994,
                         123,210,723,254,173,781,195,143,529
-------------------------------------------------------------------------------
RAP execution ends.
-------------------------------------------------------------------------------
```

4.7 *Greatest Common Divisor [GCD]*

4.7.1 Command Line Invocation Forms

rap gcd I1 I2

Calculates the g.c.d. of positive integers I1 and I2 using Euclid’s algorithm.

4.7.2 Detailed Algorithm Description

Calculates the greatest common divisor of two positive integers using Euclid’s algorithm. Euclid’s algorithm and its proof are not discussed explicitly in the TOMS paper. Olds’ CF book explains the relationship between Euclid’s algorithm and the apparatus of continued fractions, and supplies a proof. A search of the web would also reveal many pages that discuss Euclid’s g.c.d. algorithm in great detail.

4.7.3 Example Invocation

Output from the invocation of RAP to calculate the g.c.d. of two integers is reproduced below.

```
C:\ >rap gcd 13433 34048
------------------------------------------------------------------------------
RAP ($Revision: 15 $ $Date: 11/13/00 3:59a $) execution begins.
------------------------------------------------------------------------------
                arg1:                                13,433   (    5 digits)
------------------------------------------------------------------------------
                arg2:                                34,048   (    5 digits)
------------------------------------------------------------------------------
     gcd(arg1, arg2):                                   133   (    3 digits)
------------------------------------------------------------------------------
RAP execution ends.
------------------------------------------------------------------------------
```

4.8 Decimal Approximation [DAP]

4.8.1 Command Line Invocation Forms

rap dap R1

Performs long division on a rational number R1 to obtain the more familiar decimal approximation. By default, four lines (≈100) of significant figures beyond the decimal point are displayed.

rap dap R1 D

Displays R1 as a rational approximation with D as the denominator.

4.8.2 Detailed Algorithm Description

The DAP command provides crude functionality to display a more familiar and intuitive decimal approximation of a rational number.

Let h/k be the rational number to be displayed in a more familiar form, and let N/D be an approximation to h/k such that:

$$\frac{N}{D} \leq \frac{h}{k} < \frac{N+1}{D}$$

The choice of N given below will meet this inequality.

$$N = \left\lfloor \frac{Dh}{k} \right\rfloor$$

DAP operates by accepting h, k, and D as input parameters, and choosing N as specified above. Note that if D is chosen to be a power of ten, the digits of N will give the decimal form of a rational number, where the last digit will be truncated rather than rounded (note the form of the inequality above).

DAP is described as crude because it is up to the user of RAP to put the decimal point in the right place, and because it will not truncate trailing zeros or round.

The power of ten specified effectively sets the "number of decimal places" that the decimal approximation is displayed to. A value of D=1e2 will display two decimal places, a value of 1e100 will display 100 decimal places, etc. It would be possible to specify D as other than a power of ten, and in some cases this may be give useful information. For example, in microcontroller work, if one is performing $h/2^q$ scaling, choosing $D=2^q$ would give the required value of h.

4.8.3 Example Invocation

Output from the DAP command when applied to the rational number 2,043,926/7 is reproduced below. Note that the default choice of D is 10^{108} (the right length so that the imaginary decimal point occurs between lines, which allows easier interpretation). From that output below, it can be seen that 2,043,926/7 ≈ 291,989.428571428571… .

```
C:\>rap dap 2043926/7
```

```
------------------------------------------------------------------------------
RAP ($Revision: 15 $ $Date: 11/13/00 3:59a $) execution begins.
------------------------------------------------------------------------------
               arg_h:                             2,043,926   (    7 digits)
------------------------------------------------------------------------------
               arg_k:                                     7   (    1 digit)
------------------------------------------------------------------------------
                   N:                               291,989,  (  114 digits)
                        428,571,428,571,428,571,428,571,428,
                        571,428,571,428,571,428,571,428,571,
                        428,571,428,571,428,571,428,571,428,
                        571,428,571,428,571,428,571,428,571
------------------------------------------------------------------------------
                   D:                                     1,  (  109 digits)
                        000,000,000,000,000,000,000,000,000,
                        000,000,000,000,000,000,000,000,000,
                        000,000,000,000,000,000,000,000,000,
                        000,000,000,000,000,000,000,000,000
------------------------------------------------------------------------------
RAP execution ends.
------------------------------------------------------------------------------
```

The value of D used can be changed by invoking DAP with a third parameter, as shown in the output below. In the output below, 216 decimal places are displayed.

```
C:\>rap dap 2043926/7 1e216
------------------------------------------------------------------------------
RAP ($Revision: 15 $ $Date: 11/13/00 3:59a $) execution begins.
------------------------------------------------------------------------------
               arg_h:                             2,043,926   (    7 digits)
------------------------------------------------------------------------------
               arg_k:                                     7   (    1 digit)
------------------------------------------------------------------------------
                   N:                               291,989,  (  222 digits)
                        428,571,428,571,428,571,428,571,428,
                        571,428,571,428,571,428,571,428,571,
                        428,571,428,571,428,571,428,571,428,
                        571,428,571,428,571,428,571,428,571,
                        428,571,428,571,428,571,428,571,428,
                        571,428,571,428,571,428,571,428,571,
                        428,571,428,571,428,571,428,571,428,
                        571,428,571,428,571,428,571,428,571
------------------------------------------------------------------------------
                   D:                                     1,  (  217 digits)
                        000,000,000,000,000,000,000,000,000,
                        000,000,000,000,000,000,000,000,000,
                        000,000,000,000,000,000,000,000,000,
                        000,000,000,000,000,000,000,000,000,
                        000,000,000,000,000,000,000,000,000,
                        000,000,000,000,000,000,000,000,000,
                        000,000,000,000,000,000,000,000,000,
                        000,000,000,000,000,000,000,000,000
------------------------------------------------------------------------------
RAP execution ends.
------------------------------------------------------------------------------
```

Finally, DAP can also be used to with denominators not an integral power of 10. In the example invocation below, DAP is invoked with a denominator of 2^{16}=65536. The output below shows that (for microcontroller work), if a rational approximation were performed by multiplying by an integer and shifting right a number of bits, using a multiplier of 28,036 and then shifting right by 16 bits would be a good approximation of 3/7.

```
------------------------------------------------------------------------------
RAP execution ends.
------------------------------------------------------------------------------

C:\>rap dap 3/7 65536
------------------------------------------------------------------------------
RAP ($Revision: 15 $ $Date: 11/13/00 3:59a $) execution begins.
------------------------------------------------------------------------------
```

```
               arg_h:                                     3   (    1 digit)
------------------------------------------------------------------------------
               arg_k:                                     7   (    1 digit)
------------------------------------------------------------------------------
                   N:                                28,086   (    5 digits)
------------------------------------------------------------------------------
                   D:                                65,536   (    5 digits)
------------------------------------------------------------------------------
RAP execution ends.
------------------------------------------------------------------------------
```

4.9 Rational Number Sum [+]

4.9.1 Command Line Invocation Forms

rap + R1 R2

Forms the sum of R1 and R2, presenting the sum in lowest terms.

4.9.2 Detailed Algorithm Description

Forms the sum of two arbitrary rational numbers using standard algebraic techniques. The sum is presented as a rational number in lowest terms.

4.9.3 Example Invocation

Output from the invocation of RAP to form the sum of 7/91 and 3.14 is reproduced below. Note that the inputs and outputs are presented in reduced form.

```
C:\ >rap + 7/91 3.14
-------------------------------------------------------------------------------
RAP ($Revision: 15 $ $Date: 11/13/00 3:59a $) execution begins.
-------------------------------------------------------------------------------
              arg1_h:                                      1   (    1 digit)
-------------------------------------------------------------------------------
              arg1_k:                                     13   (    2 digits)
-------------------------------------------------------------------------------
              arg2_h:                                    157   (    3 digits)
-------------------------------------------------------------------------------
              arg2_k:                                     50   (    2 digits)
-------------------------------------------------------------------------------
            result_h:                                  2,091   (    4 digits)
-------------------------------------------------------------------------------
            result_k:                                    650   (    3 digits)
-------------------------------------------------------------------------------
RAP execution ends.
-------------------------------------------------------------------------------
```

4.10 *Rational Number Difference [-]*

4.10.1 Command Line Invocation Forms

rap - R1 R2

Subtracts R2 from R1, presenting the difference in lowest terms.

4.10.2 Detailed Algorithm Description

Forms the difference of two arbitrary rational numbers, using standard algebraic techniques. The sum is presented as a rational number in lowest terms.

4.10.3 Example Invocation

Output from the invocation of RAP to subtract two rational numbers is reproduced below.

```
C:\>rap - 92493234924/233472634 872434/23643
------------------------------------------------------------------------------
RAP ($Revision: 15 $ $Date: 11/13/00 3:59a $) execution begins.
------------------------------------------------------------------------------
              arg1_h:                        46,246,617,462   (   11 digits)
------------------------------------------------------------------------------
              arg1_k:                           116,736,317   (    9 digits)
------------------------------------------------------------------------------
              arg2_h:                               872,434   (    6 digits)
------------------------------------------------------------------------------
              arg2_k:                                23,643   (    5 digits)
------------------------------------------------------------------------------
            result_h:                   991,564,044,668,488   (   15 digits)
------------------------------------------------------------------------------
            result_k:                     2,759,996,742,831   (   13 digits)
------------------------------------------------------------------------------
RAP execution ends.
------------------------------------------------------------------------------
```

4.11 Rational Number Product [*]

4.11.1 Command Line Invocation Forms

rap * R1 R2

Forms the product of R1 and R2, presenting the product in lowest terms.

4.11.2 Detailed Algorithm Description

Forms the product of two arbitrary rational numbers, using standard algebraic techniques. The product is presented as a rational number in lowest terms.

4.11.3 Example Invocation

Output from the invocation of RAP to multiply two rational numbers is reproduced below.

```
C:\>rap * 0.14 7/3
------------------------------------------------------------------------------
RAP ($Revision: 15 $ $Date: 11/13/00 3:59a $) execution begins.
------------------------------------------------------------------------------
              arg1_h:                                     7   (    1 digit)
------------------------------------------------------------------------------
              arg1_k:                                    50   (    2 digits)
------------------------------------------------------------------------------
              arg2_h:                                     7   (    1 digit)
------------------------------------------------------------------------------
              arg2_k:                                     3   (    1 digit)
------------------------------------------------------------------------------
            result_h:                                    49   (    2 digits)
------------------------------------------------------------------------------
            result_k:                                   150   (    3 digits)
------------------------------------------------------------------------------
RAP execution ends.
------------------------------------------------------------------------------
```

4.12 Rational Number Quotient [/]

4.12.1 Command Line Invocation Forms

rap / R1 R2

Divides R1 by R2, presenting the quotient in lowest terms.

4.12.2 Detailed Algorithm Description

Forms the quotient of two arbitrary rational numbers, which will also be a rational number. The quotient is presented as a rational number in lowest terms.

The quotient is rephrased in integer terms:

$$\frac{\frac{a}{b}}{\frac{c}{d}} = \frac{ad}{bc}$$

4.12.3 Example Invocation

Output from the invocation of RAP to form the rational quotient of two rational numbers is reproduced below.

```
C:\>rap /  3.14 -157/2
------------------------------------------------------------------------------
RAP ($Revision: 15 $ $Date: 11/13/00 3:59a $) execution begins.
------------------------------------------------------------------------------
              arg1_h:                                   157   (    3 digits)
------------------------------------------------------------------------------
              arg1_k:                                    50   (    2 digits)
------------------------------------------------------------------------------
              arg2_h: -                                 157   (    3 digits)
------------------------------------------------------------------------------
              arg2_k:                                     2   (    1 digit)
------------------------------------------------------------------------------
            result_h: -                                   1   (    1 digit)
------------------------------------------------------------------------------
            result_k:                                    25   (    2 digits)
------------------------------------------------------------------------------
RAP execution ends.
------------------------------------------------------------------------------
```

4.13 Rational Number Raised To An Integer Power [**]

4.13.1 Command Line Invocation Forms

rap ** R1 I1

Raises R1 to the positive power of I1. The result is presented in lowest terms.

4.13.2 Algorithm Description

Raises a non-negative rational number to a positive integral power. The result is presented in lowest terms. First, the rational number is reduced so that the numerator and denominator are coprime. Then, the numerator and denominator (each integers) are each raised to an integer power using the algorithm described in section 4.6.

4.13.3 Example Invocation

Output from the invocation of RAP to raise a rational number to an integer power is reproduced below.

```
C:\>rap ** 3.14 21
------------------------------------------------------------------------------
RAP ($Revision: 15 $ $Date: 11/13/00 3:59a $) execution begins.
------------------------------------------------------------------------------
               arg_h:                                   157   (    3 digits)
------------------------------------------------------------------------------
               arg_k:                                    50   (    2 digits)
------------------------------------------------------------------------------
            exponent:                                    21   (    2 digits)
------------------------------------------------------------------------------
   arg_h ** exponent:            12,998,483,899,984,396,303,  (   47 digits)
                        172,397,297,010,470,279,141,762,157
------------------------------------------------------------------------------
   arg_k ** exponent:                           476,837,158,  (   36 digits)
                        203,125,000,000,000,000,000,000,000
------------------------------------------------------------------------------
RAP execution ends.
------------------------------------------------------------------------------
```

4.14 *Continued Fraction Partial Quotients And Convergents Of A Rational Number [CF]*

4.14.1 Command Line Invocation Forms

rap cf R1

Forms the partial quotients and convergents of non-negative rational number R1.

4.14.2 Detailed Algorithm Description

Forms the continued fraction partial quotients of an arbitrary non-negative rational number. The partial quotients are then used to calculate the convergents.

4.14.3 Example Invocation

The output below shows the invocation of RAP to form the partial quotients and convergents of a 12-digit rational approximation to π (obtained using the "π"-key on a pocket calculator).[5]

```
C:\>rap cf 3.14159265359
------------------------------------------------------------------------------
RAP ($Revision: 15 $ $Date: 11/13/00 3:59a $) execution begins.
------------------------------------------------------------------------------
************************   Inputs To CF Calculation   ************************
------------------------------------------------------------------------------
                h_in:                       314,159,265,359   (   12 digits)
------------------------------------------------------------------------------
                k_in:                       100,000,000,000   (   12 digits)
------------------------------------------------------------------------------
**************************   CF Partial Quotients   **************************
------------------------------------------------------------------------------
                a(0):                                     3   (    1 digit)
------------------------------------------------------------------------------
                a(1):                                     7   (    1 digit)
------------------------------------------------------------------------------
                a(2):                                    15   (    2 digits)
------------------------------------------------------------------------------
                a(3):                                     1   (    1 digit)
------------------------------------------------------------------------------
                a(4):                                   292   (    3 digits)
------------------------------------------------------------------------------
                a(5):                                     1   (    1 digit)
------------------------------------------------------------------------------
                a(6):                                     1   (    1 digit)
------------------------------------------------------------------------------
                a(7):                                     1   (    1 digit)
------------------------------------------------------------------------------
                a(8):                                     2   (    1 digit)
------------------------------------------------------------------------------
                a(9):                                     1   (    1 digit)
------------------------------------------------------------------------------
               a(10):                                     4   (    1 digit)
------------------------------------------------------------------------------
               a(11):                                     1   (    1 digit)
------------------------------------------------------------------------------
```

[5] The approximation used in this example is relatively crude (12 digits). Many web sites list the value of π to thousands of decimal places, and when using RAP with Farey series of large order it is recommended to use a much more precise value of π. RAP has been tested to accommodate about 460 digits of π, resulting in about 900 partial quotients. This precision should be adequate for practical rational approximations.

```
               a(12):                                     1   (    1 digit)
------------------------------------------------------------------------------
               a(13):                                     1   (    1 digit)
------------------------------------------------------------------------------
               a(14):                                     1   (    1 digit)
------------------------------------------------------------------------------
               a(15):                                     1   (    1 digit)
------------------------------------------------------------------------------
               a(16):                                     3   (    1 digit)
------------------------------------------------------------------------------
               a(17):                                     1   (    1 digit)
------------------------------------------------------------------------------
               a(18):                                    68   (    2 digits)
------------------------------------------------------------------------------
               a(19):                                     2   (    1 digit)
------------------------------------------------------------------------------
               a(20):                                     4   (    1 digit)
------------------------------------------------------------------------------
               a(21):                                     2   (    1 digit)
------------------------------------------------------------------------------
*****************************   CF Convergents   *****************************
------------------------------------------------------------------------------
                p(0):                                     3   (    1 digit)
                q(0):                                     1   (    1 digit)
------------------------------------------------------------------------------
                p(1):                                    22   (    2 digits)
                q(1):                                     7   (    1 digit)
------------------------------------------------------------------------------
                p(2):                                   333   (    3 digits)
                q(2):                                   106   (    3 digits)
------------------------------------------------------------------------------
                p(3):                                   355   (    3 digits)
                q(3):                                   113   (    3 digits)
------------------------------------------------------------------------------
                p(4):                               103,993   (    6 digits)
                q(4):                                33,102   (    5 digits)
------------------------------------------------------------------------------
                p(5):                               104,348   (    6 digits)
                q(5):                                33,215   (    5 digits)
------------------------------------------------------------------------------
                p(6):                               208,341   (    6 digits)
                q(6):                                66,317   (    5 digits)
------------------------------------------------------------------------------
                p(7):                               312,689   (    6 digits)
                q(7):                                99,532   (    5 digits)
------------------------------------------------------------------------------
                p(8):                               833,719   (    6 digits)
                q(8):                               265,381   (    6 digits)
------------------------------------------------------------------------------
                p(9):                             1,146,408   (    7 digits)
                q(9):                               364,913   (    6 digits)
------------------------------------------------------------------------------
               p(10):                             5,419,351   (    7 digits)
               q(10):                             1,725,033   (    7 digits)
------------------------------------------------------------------------------
               p(11):                             6,565,759   (    7 digits)
               q(11):                             2,089,946   (    7 digits)
------------------------------------------------------------------------------
               p(12):                            11,985,110   (    8 digits)
               q(12):                             3,814,979   (    7 digits)
------------------------------------------------------------------------------
               p(13):                            18,550,869   (    8 digits)
               q(13):                             5,904,925   (    7 digits)
------------------------------------------------------------------------------
               p(14):                            30,535,979   (    8 digits)
               q(14):                             9,719,904   (    7 digits)
------------------------------------------------------------------------------
               p(15):                            49,086,848   (    8 digits)
               q(15):                            15,624,829   (    8 digits)
------------------------------------------------------------------------------
               p(16):                           177,796,523   (    9 digits)
               q(16):                            56,594,391   (    8 digits)
------------------------------------------------------------------------------
               p(17):                           226,883,371   (    9 digits)
               q(17):                            72,219,220   (    8 digits)
```

```
------------------------------------------------------------------------------
               p(18):                        15,605,865,751   (   11 digits)
               q(18):                         4,967,501,351   (   10 digits)
------------------------------------------------------------------------------
               p(19):                        31,438,614,873   (   11 digits)
               q(19):                        10,007,221,922   (   11 digits)
------------------------------------------------------------------------------
               p(20):                       141,360,325,243   (   12 digits)
               q(20):                        44,996,389,039   (   11 digits)
------------------------------------------------------------------------------
               p(21):                       314,159,265,359   (   12 digits)
               q(21):                       100,000,000,000   (   12 digits)
------------------------------------------------------------------------------
RAP execution ends.
------------------------------------------------------------------------------
```

4.15 Rational Number With Smallest Denominator In An Interval [MIND]

4.15.1 Command Line Invocation Forms

rap mind R1 R2

Locates the rational number in the interval [R1, R2] with the smallest denominator.

4.15.2 Detailed Algorithm Description

Locates the rational number with the smallest denominator in an interval of the Farey series. This information is useful to provide an upper bound on the distance between Farey terms in the interval. The algorithm used is to find the best approximation with the smallest denominator in the interval to the midpoint of the interval, (R1+R2)/2. This algorithm is described fully in the paper.

4.15.3 Example Invocation

The invocation below shows RAP used to find the rational number with the smallest denominator in the interval [0.385, 0.386]. This rational number is 22/57.

```
C:\>rap mind 0.385 0.386
------------------------------------------------------------------------------
RAP ($Revision: 15 $ $Date: 11/13/00 3:59a $) execution begins.
------------------------------------------------------------------------------
                 l_h:                                    77   (    2 digits)
------------------------------------------------------------------------------
                 l_k:                                   200   (    3 digits)
------------------------------------------------------------------------------
                 r_h:                                   193   (    3 digits)
------------------------------------------------------------------------------
                 r_k:                                   500   (    3 digits)
------------------------------------------------------------------------------
          midpoint_h:                                   771   (    3 digits)
------------------------------------------------------------------------------
          midpoint_k:                                 2,000   (    4 digits)
------------------------------------------------------------------------------
*****************   CF Representation Of Interval Midpoint   *****************
************************   Inputs To CF Calculation   ************************
------------------------------------------------------------------------------
                h_in:                                   771   (    3 digits)
------------------------------------------------------------------------------
                k_in:                                 2,000   (    4 digits)
------------------------------------------------------------------------------
**************************   CF Partial Quotients   **************************
------------------------------------------------------------------------------
                a(0):                                     0   (    1 digit)
------------------------------------------------------------------------------
                a(1):                                     2   (    1 digit)
------------------------------------------------------------------------------
                a(2):                                     1   (    1 digit)
------------------------------------------------------------------------------
                a(3):                                     1   (    1 digit)
------------------------------------------------------------------------------
                a(4):                                     2   (    1 digit)
------------------------------------------------------------------------------
                a(5):                                     6   (    1 digit)
------------------------------------------------------------------------------
                a(6):                                     3   (    1 digit)
------------------------------------------------------------------------------
                a(7):                                     3   (    1 digit)
------------------------------------------------------------------------------
                a(8):                                     2   (    1 digit)
------------------------------------------------------------------------------
```

```
*****************************   CF Convergents   *****************************
------------------------------------------------------------------------------
                p(0):                                     0   (    1 digit)
                q(0):                                     1   (    1 digit)
------------------------------------------------------------------------------
                p(1):                                     1   (    1 digit)
                q(1):                                     2   (    1 digit)
------------------------------------------------------------------------------
                p(2):                                     1   (    1 digit)
                q(2):                                     3   (    1 digit)
------------------------------------------------------------------------------
                p(3):                                     2   (    1 digit)
                q(3):                                     5   (    1 digit)
------------------------------------------------------------------------------
                p(4):                                     5   (    1 digit)
                q(4):                                    13   (    2 digits)
------------------------------------------------------------------------------
                p(5):                                    32   (    2 digits)
                q(5):                                    83   (    2 digits)
------------------------------------------------------------------------------
                p(6):                                   101   (    3 digits)
                q(6):                                   262   (    3 digits)
------------------------------------------------------------------------------
                p(7):                                   335   (    3 digits)
                q(7):                                   869   (    3 digits)
------------------------------------------------------------------------------
                p(8):                                   771   (    3 digits)
                q(8):                                 2,000   (    4 digits)
------------------------------------------------------------------------------
********   A Rational Number With Smallest Denominator In Interval    ********
------------------------------------------------------------------------------
            result_h:                                    22   (    2 digits)
------------------------------------------------------------------------------
            result_k:                                    57   (    2 digits)
------------------------------------------------------------------------------
RAP execution ends.
------------------------------------------------------------------------------
```

4.16 Enclosing Rational Numbers In The Farey Series Of Order N [FN]

4.16.1 Command Line Invocation Forms

rap fn R1 ORDER

Locates the immediate left and right Farey neighbors of non-negative rational number R1, in the Farey series of order ORDER. Any decimal approximations are presented with the default denominator (1E108, to give four lines of digits after the decimal point).

rap fn R1 ORDER NNEIGHBORS D

Same as form immediately above, except allows specification of the number of Farey neighbors on both the left and right to generate, and the denominator to use in any decimal approximations presented (see the *DAP* command, section 4.8). A value of NNEIGHBORS greater than 10,000 is treated as 10,000.

4.16.2 Detailed Algorithm Description

Applies the continued fraction algorithms described in the TOMS paper to obtain Farey neighbors to an arbitrary non-negative rational number R1. There are two cases to consider: either the supplied rational number R1 is in the Farey series of order ORDER, or it is not. The algorithm will announce clearly which case applies. In either case, the algorithm applied is nearly identical.

<u>NOTE:</u> The implementation of this algorithm in the RAP program is subtly different than specified in the TOMS paper. The TOMS paper outlines an algorithm where the continued fraction decomposition of R1 is carried out only until the necessary partial quotients and convergents are obtained. However, the RAP implementation will form *all* partial quotients and convergents regardless of the value of ORDER. This has no effect on the Farey neighbors obtained, but it means that specifying R1 very precisely may noticeably slow the program, regardless of the value of ORDER. The results will always be as expected, but the software may be more sluggish for more precisely specified R1.

4.16.3 Example Invocation

The example invocation below is intended to generate the 10 best approximations to π^6 in the Farey series of order 65,535. The output below includes narrative explanations in a different font and with shading.

```
C:\>rap fn 3.14159265358979323846264338327950288419716939937510582097494459230781
164062862089 65535 5 1e108
```

In the command-line invocation above, the "1e108" tells RAP to use this large power-of-ten denominator as the denominator when presenting the decimal approximations of numbers. There are 27 digits per line, so 1e108 is a convenient value to position the decimal point between lines and to give 108 decimal places of precision.

```
-------------------------------------------------------------------------------
RAP ($Revision: 15 $ $Date: 11/13/00 3:59a $) execution begins.
-------------------------------------------------------------------------------
****************   Rational Number h_in/k_in To Approximate   ****************
-------------------------------------------------------------------------------
                h_in:    31,415,926,535,897,932,384,626,433,  (   80 digits)
                        832,795,028,841,971,693,993,751,058,
                        209,749,445,923,078,164,062,862,089
-------------------------------------------------------------------------------
                k_in:    10,000,000,000,000,000,000,000,000,  (   80 digits)
```

[6] π to 1,000 decimal places is supplied in section 5.1.

```
                        000,000,000,000,000,000,000,000,000,
                        000,000,000,000,000,000,000,000,000
------------------------------------------------------------------------------
***********************   Other Solution Parameters    ***********************
------------------------------------------------------------------------------
               Order:                                65,535   (    5 digits)
------------------------------------------------------------------------------
          NNEIGHBORS:                                     5   (    1 digit)
------------------------------------------------------------------------------
     DAP Denominator:                                     1,  (  109 digits)
                        000,000,000,000,000,000,000,000,000,
                        000,000,000,000,000,000,000,000,000,
                        000,000,000,000,000,000,000,000,000,
                        000,000,000,000,000,000,000,000,000
------------------------------------------------------------------------------
***************   Continued Fraction Expansion Of h_in/k_in    ***************
------------------------------------------------------------------------------
************************   Inputs To CF Calculation   ************************
------------------------------------------------------------------------------
                h_in:    31,415,926,535,897,932,384,626,433,  (   80 digits)
                        832,795,028,841,971,693,993,751,058,
                        209,749,445,923,078,164,062,862,089
------------------------------------------------------------------------------
                k_in:    10,000,000,000,000,000,000,000,000,  (   80 digits)
                        000,000,000,000,000,000,000,000,000,
                        000,000,000,000,000,000,000,000,000
------------------------------------------------------------------------------
**************************   CF Partial Quotients   **************************
------------------------------------------------------------------------------
                a(0):                                     3   (    1 digit)
------------------------------------------------------------------------------
                a(1):                                     7   (    1 digit)
------------------------------------------------------------------------------
                a(2):                                    15   (    2 digits)
------------------------------------------------------------------------------
```

RAP has generated 148 partial quotients. Most of them are deleted for space. They can be easily reproduced by running RAP with the command-line reproduced above.

```
------------------------------------------------------------------------------
              a(145):                                     1   (    1 digit)
------------------------------------------------------------------------------
              a(146):                                     2   (    1 digit)
------------------------------------------------------------------------------
              a(147):                                    16   (    2 digits)
------------------------------------------------------------------------------
*****************************   CF Convergents   *****************************
------------------------------------------------------------------------------
                p(0):                                     3   (    1 digit)
                q(0):                                     1   (    1 digit)
------------------------------------------------------------------------------
                p(1):                                    22   (    2 digits)
                q(1):                                     7   (    1 digit)
------------------------------------------------------------------------------
                p(2):                                   333   (    3 digits)
                q(2):                                   106   (    3 digits)
------------------------------------------------------------------------------
```

RAP has generated 148 convergents. Most of them are deleted for space. They can be easily reproduced by running RAP with the command-line reproduced above.

```
------------------------------------------------------------------------------
              p(145):       725,955,471,712,149,333,468,082,  (   78 digits)
                        037,080,816,297,117,171,943,844,090,
                        600,740,370,938,512,424,999,369,433
              q(145):       231,078,803,575,194,322,803,693,  (   78 digits)
                        901,261,807,173,022,671,765,576,571,
                        012,554,067,457,355,871,085,907,056
------------------------------------------------------------------------------
              p(146):     1,918,123,191,511,611,440,697,396,  (   79 digits)
                        987,232,138,284,053,407,628,119,185,
                        475,563,067,186,535,358,691,468,291
              q(146):       610,557,574,776,550,354,824,769,  (   78 digits)
                        131,171,137,051,686,083,014,651,464,
                        311,715,370,783,915,258,057,130,809
------------------------------------------------------------------------------
```

```
              p(147):    31,415,926,535,897,932,384,626,433,  (   80 digits)
                        832,795,028,841,971,693,993,751,058,
                        209,749,445,923,078,164,062,862,089
              q(147):    10,000,000,000,000,000,000,000,000,  (   80 digits)
                        000,000,000,000,000,000,000,000,000,
                        000,000,000,000,000,000,000,000,000
-------------------------------------------------------------------------------
```

The first rational number below is the highest-order convergent with a denominator which is not larger than 65,535. The second rational number is an intermediate fraction chosen according to a formula in the TOMS paper. It is proved in the paper that these two numbers are the two Farey neighbors to the rational number of interest (if the rational number of interest is not in the Farey series). The two numbers are not necessarily in ascending order (it depends on whether the convergent is even or odd). If the rational number of interest is already in the Farey series of interest, the convergent will be this rational number, and the intermediate fraction will be its left or right Farey neighbor.

```
*****************   Highest-Order Convergent With q(i)<=N    *****************
-------------------------------------------------------------------------------
                p(5):                               104,348   (    6 digits)
                q(5):                                33,215   (    5 digits)
-------------------------------------------------------------------------------
*******************   Accompanying Intermediate Fraction   *******************
-------------------------------------------------------------------------------
      intermediate_h:                               103,993   (    6 digits)
      intermediate_k:                                33,102   (    5 digits)
-------------------------------------------------------------------------------
```

The line below indicates that the rational number of interest (3.14…) is not in the Farey series of interest ($F_{65,535}$).. In the neighbors presented just after this in the output, the neighbors are subscripted so that any number with a negative subscript is less than the rational number of interest, any number with a positive subscript is larger than the rational number of interest, and the subscript "0" is reserved for the rational number of interest if it appears in the Farey series of interest. In this case, the rational number of interest is not in the Farey series of interest, so there will not be a number presented with subscript "0".

```
*******************************************************************************
**************    h_in/k_in IS NOT In Farey Series Of Interest   **************
*******************************************************************************
-------------------------------------------------------------------------------
************************   Farey Neighbor Index -5    ************************
```

The subscript (or index) of "-5" here indicates that this is the fifth Farey neighbor to the left of the rational number of interest.

```
-------------------------------------------------------------------------------
               h(-5):                               205,501   (    6 digits)
-------------------------------------------------------------------------------
               k(-5):                                65,413   (    5 digits)
-------------------------------------------------------------------------------
```

The rational number below is the rational number 205,501/65,413 expressed as a decimal approximation. From the output below, one knows that 205,501/65,413 is 3.141592649….

```
           DAP_N(-5):                                     3,  (  109 digits)
                        141,592,649,779,095,898,368,825,768,
                        578,111,384,587,161,573,387,552,932,
                        903,245,532,233,653,860,853,347,193,
                        982,847,446,226,285,294,971,947,472
-------------------------------------------------------------------------------
           DAP_D(-5):                                     1,  (  109 digits)
                        000,000,000,000,000,000,000,000,000,
                        000,000,000,000,000,000,000,000,000,
                        000,000,000,000,000,000,000,000,000,
                        000,000,000,000,000,000,000,000,000
-------------------------------------------------------------------------------
```

The rational number below is the error (approximation minus actual) of the approximation, expressed as a lowest-terms rational number. This rational number might not be useful or intuitive for Farey series of large orders.

```
         error_h(-5): -       2,492,691,451,075,568,916,304,  (   76 digits)
                        621,221,639,894,419,213,237,970,674,
                        340,506,166,311,945,843,997,827,757
-------------------------------------------------------------------------------
         error_k(-5):                                   654,  (   84 digits)
                        130,000,000,000,000,000,000,000,000,
                        000,000,000,000,000,000,000,000,000,
                        000,000,000,000,000,000,000,000,000
------------------------------------------------------------------------------
```

The rational number below is the error (approximation minus actual) of the approximation, expressed as a decimal approximation. Because the imaginary decimal point is positioned to the left of the fourth line up, it is easy to see that the approximation error is about 3.81×10^{-9}.

```
     ERROR_DAP_N(-5): -           3,810,697,340,093,817,614,  (  100 digits)
                        701,391,499,610,007,825,987,552,888,
                        071,699,060,074,162,545,432,861,706,
                        017,152,553,773,714,705,028,052,527
------------------------------------------------------------------------------
     ERROR_DAP_D(-5):                                     1,  (  109 digits)
                        000,000,000,000,000,000,000,000,000,
                        000,000,000,000,000,000,000,000,000,
                        000,000,000,000,000,000,000,000,000,
                        000,000,000,000,000,000,000,000,000
------------------------------------------------------------------------------
************************   Farey Neighbor Index -4    ************************
------------------------------------------------------------------------------
               h(-4):                               102,928   (    6 digits)
------------------------------------------------------------------------------
               k(-4):                                32,763   (    5 digits)
------------------------------------------------------------------------------
           DAP_N(-4):                                     3,  (  109 digits)
                        141,592,650,245,703,995,360,620,211,
                        824,314,012,758,294,417,483,136,464,
                        914,690,351,921,374,721,484,601,532,
                        216,219,515,917,345,786,405,396,331
------------------------------------------------------------------------------
           DAP_D(-4):                                     1,  (  109 digits)
                        000,000,000,000,000,000,000,000,000,
                        000,000,000,000,000,000,000,000,000,
                        000,000,000,000,000,000,000,000,000,
                        000,000,000,000,000,000,000,000,000
------------------------------------------------------------------------------
         error_h(-4): -       1,095,623,958,717,515,851,663,  (   76 digits)
                        863,529,949,518,610,317,265,920,126,
                        021,096,777,809,889,191,550,621,907
------------------------------------------------------------------------------
         error_k(-4):                                   327,  (   84 digits)
                        630,000,000,000,000,000,000,000,000,
                        000,000,000,000,000,000,000,000,000,
                        000,000,000,000,000,000,000,000,000
------------------------------------------------------------------------------
     ERROR_DAP_N(-4): -           3,344,089,243,102,023,171,  (  100 digits)
                        455,188,871,438,874,981,891,969,356,
                        060,254,240,386,441,684,801,607,367,
                        783,780,484,082,654,213,594,603,668
------------------------------------------------------------------------------
     ERROR_DAP_D(-4):                                     1,  (  109 digits)
                        000,000,000,000,000,000,000,000,000,
                        000,000,000,000,000,000,000,000,000,
                        000,000,000,000,000,000,000,000,000,
                        000,000,000,000,000,000,000,000,000
------------------------------------------------------------------------------
************************   Farey Neighbor Index -3    ************************
------------------------------------------------------------------------------
               h(-3):                               103,283   (    6 digits)
------------------------------------------------------------------------------
               k(-3):                                32,876   (    5 digits)
------------------------------------------------------------------------------
           DAP_N(-3):                                     3,  (  109 digits)
                        141,592,651,174,108,772,356,734,395,
                        911,911,424,747,536,196,617,593,381,
                        189,925,781,725,270,714,198,807,640,
                        832,218,031,390,680,131,402,847,061
------------------------------------------------------------------------------
           DAP_D(-3):                                     1,  (  109 digits)
                        000,000,000,000,000,000,000,000,000,
                        000,000,000,000,000,000,000,000,000,
                        000,000,000,000,000,000,000,000,000,
                        000,000,000,000,000,000,000,000,000
------------------------------------------------------------------------------
```

```
         error_h(-3): -         198,545,106,269,244,659,671,  (   75 digits)
                        742,342,052,165,352,934,639,947,425,
                        930,696,041,779,430,432,663,509,491
------------------------------------------------------------------------------
         error_k(-3):                                    82,  (   83 digits)
                        190,000,000,000,000,000,000,000,000,
                        000,000,000,000,000,000,000,000,000,
                        000,000,000,000,000,000,000,000,000
------------------------------------------------------------------------------
     ERROR_DAP_N(-3): -           2,415,684,466,105,908,987,  (  100 digits)
                        367,591,459,449,633,202,757,512,439,
                        785,018,810,582,545,692,087,401,259,
                        167,781,968,609,319,868,597,152,938
------------------------------------------------------------------------------
     ERROR_DAP_D(-3):                                     1,  (  109 digits)
                        000,000,000,000,000,000,000,000,000,
                        000,000,000,000,000,000,000,000,000,
                        000,000,000,000,000,000,000,000,000,
                        000,000,000,000,000,000,000,000,000
------------------------------------------------------------------------------
************************   Farey Neighbor Index -2    ************************
------------------------------------------------------------------------------
               h(-2):                               103,638   (    6 digits)
------------------------------------------------------------------------------
               k(-2):                                32,989   (    5 digits)
------------------------------------------------------------------------------
           DAP_N(-2):                                     3,  (  109 digits)
                        141,592,652,096,153,263,208,948,437,
                        357,907,181,181,605,989,875,413,016,
                        460,032,131,922,762,132,832,156,173,
                        269,877,838,067,234,532,723,028,888
------------------------------------------------------------------------------
           DAP_D(-2):                                     1,  (  109 digits)
                        000,000,000,000,000,000,000,000,000,
                        000,000,000,000,000,000,000,000,000,
                        000,000,000,000,000,000,000,000,000,
                        000,000,000,000,000,000,000,000,000
------------------------------------------------------------------------------
         error_h(-2): -         492,736,891,436,441,425,710,  (   75 digits)
                        075,206,467,804,213,159,853,659,281,
                        424,471,556,425,554,269,757,454,021
------------------------------------------------------------------------------
         error_k(-2):                                   329,  (   84 digits)
                        890,000,000,000,000,000,000,000,000,
                        000,000,000,000,000,000,000,000,000,
                        000,000,000,000,000,000,000,000,000
------------------------------------------------------------------------------
     ERROR_DAP_N(-2): -           1,493,639,975,253,694,945,  (  100 digits)
                        921,595,703,015,563,409,499,692,804,
                        514,912,460,385,054,273,454,052,726,
                        730,122,161,932,765,467,276,971,111
------------------------------------------------------------------------------
     ERROR_DAP_D(-2):                                     1,  (  109 digits)
                        000,000,000,000,000,000,000,000,000,
                        000,000,000,000,000,000,000,000,000,
                        000,000,000,000,000,000,000,000,000,
                        000,000,000,000,000,000,000,000,000
------------------------------------------------------------------------------
************************   Farey Neighbor Index -1    ************************
------------------------------------------------------------------------------
               h(-1):                               103,993   (    6 digits)
------------------------------------------------------------------------------
               k(-1):                                33,102   (    5 digits)
------------------------------------------------------------------------------
           DAP_N(-1):                                     3,  (  109 digits)
                        141,592,653,011,902,604,072,261,494,
                        773,729,684,007,008,639,961,331,641,
                        592,653,011,902,604,072,261,494,773,
                        729,684,007,008,639,961,331,641,592
------------------------------------------------------------------------------
           DAP_D(-1):                                     1,  (  109 digits)
                        000,000,000,000,000,000,000,000,000,
                        000,000,000,000,000,000,000,000,000,
                        000,000,000,000,000,000,000,000,000,
                        000,000,000,000,000,000,000,000,000
```

```
------------------------------------------------------------------------------
         error_h(-1): -          95,646,678,897,952,106,366,  (   74 digits)
                        590,522,363,473,507,290,573,764,429,
                        563,079,472,866,693,404,430,435,039
------------------------------------------------------------------------------
         error_k(-1):                                   165,  (   84 digits)
                        510,000,000,000,000,000,000,000,000,
                        000,000,000,000,000,000,000,000,000,
                        000,000,000,000,000,000,000,000,000
------------------------------------------------------------------------------
```

This is the last approximation on the left of the number of interest (note the subscript of -1). The error is about 5.78×10^{-10}.

```
     ERROR_DAP_N(-1): -             577,890,634,390,381,888,  (   99 digits)
                        505,773,200,190,160,759,413,774,179,
                        382,291,580,405,212,334,024,714,126,
                        270,315,992,991,360,038,668,358,407
------------------------------------------------------------------------------
     ERROR_DAP_D(-1):                                     1,  (  109 digits)
                        000,000,000,000,000,000,000,000,000,
                        000,000,000,000,000,000,000,000,000,
                        000,000,000,000,000,000,000,000,000,
                        000,000,000,000,000,000,000,000,000
------------------------------------------------------------------------------
*************************   Farey Neighbor Index 1   *************************
------------------------------------------------------------------------------
                h(1):                               104,348   (    6 digits)
------------------------------------------------------------------------------
                k(1):                                33,215   (    5 digits)
------------------------------------------------------------------------------
            DAP_N(1):                                     3,  (  109 digits)
                        141,592,653,921,421,044,708,715,941,
                        592,653,921,421,044,708,715,941,592,
                        653,921,421,044,708,715,941,592,653,
                        921,421,044,708,715,941,592,653,921
------------------------------------------------------------------------------
            DAP_D(1):                                     1,  (  109 digits)
                        000,000,000,000,000,000,000,000,000,
                        000,000,000,000,000,000,000,000,000,
                        000,000,000,000,000,000,000,000,000,
                        000,000,000,000,000,000,000,000,000
------------------------------------------------------------------------------
          error_h(1):            22,030,035,168,926,600,048,  (   74 digits)
                        742,623,402,782,036,799,511,720,312,
                        634,430,732,991,756,130,407,142,773
------------------------------------------------------------------------------
          error_k(1):                                    66,  (   83 digits)
                        430,000,000,000,000,000,000,000,000,
                        000,000,000,000,000,000,000,000,000,
                        000,000,000,000,000,000,000,000,000
------------------------------------------------------------------------------
```

This is the first approximation on the right of the number of interest (note the subscript of 1). The error is about 3.32×10^{-10}.

```
      ERROR_DAP_N(1):               331,627,806,246,072,558,  (   99 digits)
                        313,151,037,223,875,309,340,835,771,
                        678,976,828,736,892,309,655,383,753,
                        921,421,044,708,715,941,592,653,921
------------------------------------------------------------------------------
      ERROR_DAP_D(1):                                     1,  (  109 digits)
                        000,000,000,000,000,000,000,000,000,
                        000,000,000,000,000,000,000,000,000,
                        000,000,000,000,000,000,000,000,000,
                        000,000,000,000,000,000,000,000,000
------------------------------------------------------------------------------
*************************   Farey Neighbor Index 2   *************************
------------------------------------------------------------------------------
                h(2):                               104,703   (    6 digits)
------------------------------------------------------------------------------
                k(2):                                33,328   (    5 digits)
------------------------------------------------------------------------------
            DAP_N(2):                                     3,  (  109 digits)
                        141,592,654,824,771,963,514,162,265,
                        962,554,008,641,382,621,219,395,103,
                        216,514,642,342,774,843,975,036,005,
                        760,921,747,479,596,735,477,676,428
------------------------------------------------------------------------------
            DAP_D(2):                                     1,  (  109 digits)
                        000,000,000,000,000,000,000,000,000,
                        000,000,000,000,000,000,000,000,000,
                        000,000,000,000,000,000,000,000,000,
                        000,000,000,000,000,000,000,000,000
------------------------------------------------------------------------------
          error_h(2):            25,724,606,842,823,138,326,  (   74 digits)
                        287,954,922,172,961,411,016,545,749,
                        091,904,142,228,184,257,058,268,613
------------------------------------------------------------------------------
          error_k(2):                                    20,  (   83 digits)
                        830,000,000,000,000,000,000,000,000,
                        000,000,000,000,000,000,000,000,000,
                        000,000,000,000,000,000,000,000,000
------------------------------------------------------------------------------
      ERROR_DAP_N(2):             1,234,978,725,051,518,882,  (  100 digits)
                        683,051,124,444,213,221,844,289,282,
                        241,570,050,034,958,437,688,827,105,
                        760,921,747,479,596,735,477,676,428
------------------------------------------------------------------------------
      ERROR_DAP_D(2):                                     1,  (  109 digits)
                        000,000,000,000,000,000,000,000,000,
                        000,000,000,000,000,000,000,000,000,
                        000,000,000,000,000,000,000,000,000,
                        000,000,000,000,000,000,000,000,000
------------------------------------------------------------------------------
*************************   Farey Neighbor Index 3   *************************
------------------------------------------------------------------------------
                h(3):                               105,058   (    6 digits)
------------------------------------------------------------------------------
                k(3):                                33,441   (    5 digits)
------------------------------------------------------------------------------
            DAP_N(3):                                     3,  (  109 digits)
                        141,592,655,722,017,882,240,363,625,
                        489,668,371,161,149,487,156,484,554,
                        887,712,688,017,702,819,891,749,648,
                        634,909,243,144,642,803,743,907,179
------------------------------------------------------------------------------
            DAP_D(3):                                     1,  (  109 digits)
                        000,000,000,000,000,000,000,000,000,
                        000,000,000,000,000,000,000,000,000,
                        000,000,000,000,000,000,000,000,000,
                        000,000,000,000,000,000,000,000,000
------------------------------------------------------------------------------
          error_h(3):           713,037,243,125,707,426,197,  (   75 digits)
                        501,440,495,624,581,154,970,862,407,
                        768,778,886,343,115,573,828,881,751
------------------------------------------------------------------------------
          error_k(3):                                   334,  (   84 digits)
                        410,000,000,000,000,000,000,000,000,
                        000,000,000,000,000,000,000,000,000,
                        000,000,000,000,000,000,000,000,000
------------------------------------------------------------------------------
      ERROR_DAP_N(3):             2,132,224,643,777,720,242,  (  100 digits)
                        210,165,486,963,980,087,781,378,733,
                        912,768,095,709,886,413,605,540,748,
                        634,909,243,144,642,803,743,907,179
------------------------------------------------------------------------------
      ERROR_DAP_D(3):                                     1,  (  109 digits)
                        000,000,000,000,000,000,000,000,000,
                        000,000,000,000,000,000,000,000,000,
                        000,000,000,000,000,000,000,000,000,
                        000,000,000,000,000,000,000,000,000
------------------------------------------------------------------------------
*************************   Farey Neighbor Index 4   *************************
------------------------------------------------------------------------------
                h(4):                               105,413   (    6 digits)
------------------------------------------------------------------------------
                k(4):                                33,554   (    5 digits)
------------------------------------------------------------------------------
            DAP_N(4):                                     3,  (  109 digits)
                        141,592,656,613,220,480,419,622,101,
                        686,833,164,451,332,180,962,031,352,
                        446,802,169,637,003,039,876,020,742,
                        683,435,655,957,560,946,533,945,282
------------------------------------------------------------------------------
            DAP_D(4):                                     1,  (  109 digits)
                        000,000,000,000,000,000,000,000,000,
                        000,000,000,000,000,000,000,000,000,
                        000,000,000,000,000,000,000,000,000,
                        000,000,000,000,000,000,000,000,000
------------------------------------------------------------------------------
          error_h(4):           507,240,388,383,122,319,587,  (   75 digits)
                        197,801,118,240,889,866,838,496,415,
                        033,545,748,517,641,517,362,732,847
------------------------------------------------------------------------------
          error_k(4):                                   167,  (   84 digits)
                        770,000,000,000,000,000,000,000,000,
                        000,000,000,000,000,000,000,000,000,
                        000,000,000,000,000,000,000,000,000
------------------------------------------------------------------------------
      ERROR_DAP_N(4):             3,023,427,241,956,978,718,  (  100 digits)
                        407,330,280,254,162,781,586,925,531,
                        471,857,577,329,186,633,589,811,842,
                        683,435,655,957,560,946,533,945,282
------------------------------------------------------------------------------
      ERROR_DAP_D(4):                                     1,  (  109 digits)
                        000,000,000,000,000,000,000,000,000,
                        000,000,000,000,000,000,000,000,000,
                        000,000,000,000,000,000,000,000,000,
                        000,000,000,000,000,000,000,000,000
------------------------------------------------------------------------------
*************************   Farey Neighbor Index 5   *************************
------------------------------------------------------------------------------
                h(5):                               105,768   (    6 digits)
------------------------------------------------------------------------------
                k(5):                                33,667   (    5 digits)
------------------------------------------------------------------------------
            DAP_N(5):                                     3,  (  109 digits)
                        141,592,657,498,440,609,498,915,852,
                        318,293,878,278,432,886,803,100,959,
                        396,441,619,389,907,030,623,459,173,
                        671,547,806,457,361,808,298,927,733
------------------------------------------------------------------------------
            DAP_D(5):                                     1,  (  109 digits)
                        000,000,000,000,000,000,000,000,000,
                        000,000,000,000,000,000,000,000,000,
                        000,000,000,000,000,000,000,000,000,
                        000,000,000,000,000,000,000,000,000
------------------------------------------------------------------------------
          error_h(5):         1,315,924,310,406,781,852,151,  (   76 digits)
                        289,763,977,338,978,312,383,123,252,
                        365,404,107,727,450,495,622,049,637
------------------------------------------------------------------------------
          error_k(5):                                   336,  (   84 digits)
                        670,000,000,000,000,000,000,000,000,
                        000,000,000,000,000,000,000,000,000,
                        000,000,000,000,000,000,000,000,000
------------------------------------------------------------------------------
      ERROR_DAP_N(5):             3,908,647,371,036,272,469,  (  100 digits)
                        038,790,994,081,263,487,427,995,138,
                        421,497,027,082,090,624,337,250,273,
                        671,547,806,457,361,808,298,927,733
------------------------------------------------------------------------------
      ERROR_DAP_D(5):                                     1,  (  109 digits)
                        000,000,000,000,000,000,000,000,000,
                        000,000,000,000,000,000,000,000,000,
                        000,000,000,000,000,000,000,000,000,
                        000,000,000,000,000,000,000,000,000
------------------------------------------------------------------------------
RAP execution ends.
------------------------------------------------------------------------------
```

Practical Note: For Farey series of large order, the terms become quite dense. If an irrational number to be approximated isn't specified precisely enough, it is easy to accidentally generate Farey neighbors which enclose the rational approximation specified, but do not enclose the irrational number.

For example, using 3.141592654 as a rational approximation to π to obtain the best approximations to π in $F_{65,535}$ yields 104,348/33,215 and 104,703/33,328 as the two best approximations to π. However, it is easy to show that

$$p < \frac{104{,}348}{33{,}215} < 3.141592654 < \frac{104{,}703}{33{,}328}.$$

In other words, the rational numbers in $F_{65,535}$ generated enclose the rational approximation of π used, but they do not enclose π.

There are three approaches to mitigate this type of problem.

The first approach is analytic and practical. It can be shown that the distance between two consecutive terms of the Farey series of any order is 1/ab, where a and b are the denominators of the neighboring terms. It follows immediately that in a Farey series of order N, the minimum distance between terms is $1/(N^2-N)$. Although it won't be done here, this information can be used to derive the required precision of the rational approximation to the irrational number of interest used to apply the CF algorithms to obtain Farey neighbors, so that at most one Farey term appears between the irrational and the rational approximation used to apply the algorithms. In other words, one could derive a rule of thumb for how many decimal places are appropriate as a function of the order of the Farey series.

The second (analytic and usually impractical) approach would be to calculate the continued fraction partial quotients required symbolically rather than numerically. Unfortunately, doing this by hand is generally not practical (although it can be done for many algebraic numbers relatively easily). (Commercial software such as *Mathematica* will do this automatically, even for transcendentals, but as of this writing *Mathematica* costs about $1,500.) Every case where the wrong Farey neighbors are located corresponds to an error in the partial quotients and convergents used—or to put it another way, the continued fraction representation of the rational approximation used differs from the true continued fraction representation of the irrational relatively [too] early in the partial quotients and convergents. (This gets into technical detail and won't be discussed here.)

The third approach is practical. If the irrational number can be bounded by rational numbers, and if the CF algorithms give the same result at both of the rational bounds, then the rational approximation of the irrational is specified precisely enough. The case of 3.141592654 is atypical because it is rounded rather than truncated (it came from a pocket calculator). The actual inequality at this number of decimal places which confines π is:

$$3.141592653 < p < 3.141592654$$

If the CF algorithms presented give the same results for both 3.141592653 and 3.141592654, then the correct Farey neighbors of π are identified.

Using 3.141592653 as a rational approximation gives 103,638/32,989 and 103,993/33,102 as the best approximations in F_{65535}. Using 3.141592654 as a rational approximation gives 104,348/33,215 and 104,703/33,328 as the best approximations in $F_{65,535}$. The results are not identical, so neither pair of neighbors can be trusted to be the correct pair.

4.17 Upper Bound On Distance Between Farey Terms In An Interval [FNDMAX]

4.17.1 Command Line Invocation Forms

rap fndmax LL UL N

Calculates an upper bound on the maximum distance between terms of F_N in the interval [LL,UL] using the techniques developed in the TOMS paper. LL and UL must both be in F_N, and LL and UL must be non-negative.

4.17.2 Detailed Algorithm Description

Applies the continued fraction algorithms described in the TOMS paper to find the rational number with the the smallest denominator in the interval [LL, UL], and to place an upper bound on the distance between terms in F_N.

The results in the TOMS paper state the distance in terms of an upper bound on the distance between r_A and r_I, i.e.

$$\left|\frac{h}{k}\right| - r_I < \frac{1}{2q_{MIN}\max(q_{MIN}, k_{MAX} - q_{MIN})}$$

However, this command calculates the distance as an upper bound on the distance between consecutive Farey terms, which is twice the result above. The formula used by this command to obtain a result is:

$$result = \frac{1}{q_{MIN}\max(q_{MIN}, k_{MAX} - q_{MIN})}$$

4.17.3 Example Invocation

The example invocation below is the same example as at the end of the TOMS paper. Note that the upper bound obtained is twice the value in the TOMS paper, for the reasons discussed above.

```
c:\>rap fndmax 0.385 0.386 500
------------------------------------------------------------------------------
RAP ($Revision: 15 $ $Date: 11/13/00 3:59a $) execution begins.
------------------------------------------------------------------------------
                 l_h:                                    77   (    2 digits)
------------------------------------------------------------------------------
                 l_k:                                   200   (    3 digits)
------------------------------------------------------------------------------
                 r_h:                                   193   (    3 digits)
------------------------------------------------------------------------------
                 r_k:                                   500   (    3 digits)
------------------------------------------------------------------------------
               k_max:                                   500   (    3 digits)
------------------------------------------------------------------------------
               dap_D:                                     1,  (  109 digits)
                        000,000,000,000,000,000,000,000,000,
                        000,000,000,000,000,000,000,000,000,
                        000,000,000,000,000,000,000,000,000,
                        000,000,000,000,000,000,000,000,000
------------------------------------------------------------------------------
          midpoint_h:                                   771   (    3 digits)
```

```
------------------------------------------------------------------------------
          midpoint_k:                                 2,000   (    4 digits)
------------------------------------------------------------------------------
*****************   CF Representation Of Interval Midpoint   *****************
************************   Inputs To CF Calculation   ************************
------------------------------------------------------------------------------
                h_in:                                   771   (    3 digits)
------------------------------------------------------------------------------
                k_in:                                 2,000   (    4 digits)
------------------------------------------------------------------------------
**************************   CF Partial Quotients   **************************
------------------------------------------------------------------------------
                a(0):                                     0   (    1 digit)
------------------------------------------------------------------------------
                a(1):                                     2   (    1 digit)
------------------------------------------------------------------------------
                a(2):                                     1   (    1 digit)
------------------------------------------------------------------------------
                a(3):                                     1   (    1 digit)
------------------------------------------------------------------------------
                a(4):                                     2   (    1 digit)
------------------------------------------------------------------------------
                a(5):                                     6   (    1 digit)
------------------------------------------------------------------------------
                a(6):                                     3   (    1 digit)
------------------------------------------------------------------------------
                a(7):                                     3   (    1 digit)
------------------------------------------------------------------------------
                a(8):                                     2   (    1 digit)
------------------------------------------------------------------------------
*****************************   CF Convergents   *****************************
------------------------------------------------------------------------------
                p(0):                                     0   (    1 digit)
                q(0):                                     1   (    1 digit)
------------------------------------------------------------------------------
                p(1):                                     1   (    1 digit)
                q(1):                                     2   (    1 digit)
------------------------------------------------------------------------------
                p(2):                                     1   (    1 digit)
                q(2):                                     3   (    1 digit)
------------------------------------------------------------------------------
                p(3):                                     2   (    1 digit)
                q(3):                                     5   (    1 digit)
------------------------------------------------------------------------------
                p(4):                                     5   (    1 digit)
                q(4):                                    13   (    2 digits)
------------------------------------------------------------------------------
                p(5):                                    32   (    2 digits)
                q(5):                                    83   (    2 digits)
------------------------------------------------------------------------------
                p(6):                                   101   (    3 digits)
                q(6):                                   262   (    3 digits)
------------------------------------------------------------------------------
                p(7):                                   335   (    3 digits)
                q(7):                                   869   (    3 digits)
------------------------------------------------------------------------------
                p(8):                                   771   (    3 digits)
                q(8):                                 2,000   (    4 digits)
------------------------------------------------------------------------------
********   A Rational Number With Smallest Denominator In Interval    ********
------------------------------------------------------------------------------
            result_h:                                    22   (    2 digits)
------------------------------------------------------------------------------
            result_k:                                    57   (    2 digits)
------------------------------------------------------------------------------
*****   Upper Bound On Distance Between Farey Terms As Rational Number   *****
------------------------------------------------------------------------------
          error_ub_h:                                     1   (    1 digit)
------------------------------------------------------------------------------
          error_up_k:                                25,251   (    5 digits)
------------------------------------------------------------------------------
**   Upper Bound On Distance Between Farey Terms As Decimal Approximation   **
------------------------------------------------------------------------------
               dap_h:        39,602,391,984,475,862,342,085,  (  104 digits)
                        461,961,902,498,910,934,220,426,913,
                        785,592,649,796,047,681,279,949,308,
                        938,259,870,896,202,130,608,688,764
------------------------------------------------------------------------------
               dap_k:                                     1,  (  109 digits)
                        000,000,000,000,000,000,000,000,000,
                        000,000,000,000,000,000,000,000,000,
                        000,000,000,000,000,000,000,000,000,
                        000,000,000,000,000,000,000,000,000
------------------------------------------------------------------------------
RAP execution ends.
------------------------------------------------------------------------------
```

4.18 Enclosing Rational Numbers In A Rectangular Area Of The Integer Lattice [FAB]

4.18.1 Command Line Invocation Forms

rap fab R1 HMAX KMAX

Finds the two rational numbers in $F_{KMAX,\overline{HMAX}}$ which enclose the non-negative rational number R1.

rap fab R1 HMAX KMAX NNEIGHBORS D

Same as above but provides NNEIGHBORS terms in $F_{KMAX,HMAX}$ on both the left and right, and will emit decimal approximations with denominator D.

4.18.2 Detailed Algorithm Description

Applies the continued fraction algorithms described in the TOMS paper to obtain neighbors to an arbitrary non-negative rational number R1 in $F_{KMAX,HMAX}$. There are two cases to consider: either the supplied rational number R1 is in $F_{KMAX,HMAX}$, or it is not. The algorithm will announce clearly which case applies. In either case, the algorithm applied is nearly identical.

4.18.3 Example Invocation

The invocation below shows RAP used to find the neighbors to 0.31830989 (approximately 1/π) in the rectangular area of the integer lattice bounded by h≤193 and k≤500. Although it isn't shown below, RAP handles all of the boundary cases at the "corner" near h/k=193/500. The output below includes narrative explanations in a different font and with shading.

```
c:\>rap fab 0.31830989 193 500 2 1e54
```

In the command-line invocation above, "2" tells RAP to display 2 neighbors on each side of the rational number to be approximated, and "1e54" is the DAP denominator, which indicates that two lines of digits after the decimal point are desired (see the DAP command).

```
------------------------------------------------------------------------------
RAP ($Revision: 15 $ $Date: 11/13/00 3:59a $) execution begins.
------------------------------------------------------------------------------
****************   Rational Number h_in/k_in To Approximate   ****************
------------------------------------------------------------------------------
                h_in:                            31,830,989   (    8 digits)
------------------------------------------------------------------------------
                k_in:                           100,000,000   (    9 digits)
------------------------------------------------------------------------------
***********************   Other Solution Parameters    ***********************
------------------------------------------------------------------------------
                hmax:                                   193   (    3 digits)
------------------------------------------------------------------------------
                kmax:                                   500   (    3 digits)
------------------------------------------------------------------------------
          NNEIGHBORS:                                     2   (    1 digit)
------------------------------------------------------------------------------
     DAP Denominator:                                     1,  (   55 digits)
                        000,000,000,000,000,000,000,000,000,
                        000,000,000,000,000,000,000,000,000
```

Below are the continued fraction partial quotients and convergents of the rational number to be approximated. These are necessary to economically locate the immediate Farey neighbors along the "right wall" of the rectangular area of the integer lattice, if this proves necessary.

```
------------------------------------------------------------------------------
***************   Continued Fraction Expansion Of h_in/k_in    ***************
------------------------------------------------------------------------------
```

```
************************   Inputs To CF Calculation   ************************
------------------------------------------------------------------------------
                h_in:                            31,830,989   (    8 digits)
------------------------------------------------------------------------------
                k_in:                           100,000,000   (    9 digits)
------------------------------------------------------------------------------
**************************   CF Partial Quotients   **************************
------------------------------------------------------------------------------
                a(0):                                     0   (    1 digit)
------------------------------------------------------------------------------
                a(1):                                     3   (    1 digit)
------------------------------------------------------------------------------
                a(2):                                     7   (    1 digit)
------------------------------------------------------------------------------
                a(3):                                    15   (    2 digits)
------------------------------------------------------------------------------
                a(4):                                     1   (    1 digit)
------------------------------------------------------------------------------
                a(5):                                   256   (    3 digits)
------------------------------------------------------------------------------
                a(6):                                     3   (    1 digit)
------------------------------------------------------------------------------
                a(7):                                     5   (    1 digit)
------------------------------------------------------------------------------
                a(8):                                     5   (    1 digit)
------------------------------------------------------------------------------
                a(9):                                    13   (    2 digits)
------------------------------------------------------------------------------
*****************************   CF Convergents   *****************************
------------------------------------------------------------------------------
                p(0):                                     0   (    1 digit)
                q(0):                                     1   (    1 digit)
------------------------------------------------------------------------------
                p(1):                                     1   (    1 digit)
                q(1):                                     3   (    1 digit)
------------------------------------------------------------------------------
                p(2):                                     7   (    1 digit)
                q(2):                                    22   (    2 digits)
------------------------------------------------------------------------------
                p(3):                                   106   (    3 digits)
                q(3):                                   333   (    3 digits)
------------------------------------------------------------------------------
                p(4):                                   113   (    3 digits)
                q(4):                                   355   (    3 digits)
------------------------------------------------------------------------------
                p(5):                                29,034   (    5 digits)
                q(5):                                91,213   (    5 digits)
------------------------------------------------------------------------------
                p(6):                                87,215   (    5 digits)
                q(6):                               273,994   (    6 digits)
------------------------------------------------------------------------------
                p(7):                               465,109   (    6 digits)
                q(7):                             1,461,183   (    7 digits)
------------------------------------------------------------------------------
                p(8):                             2,412,760   (    7 digits)
                q(8):                             7,579,909   (    7 digits)
------------------------------------------------------------------------------
                p(9):                            31,830,989   (    8 digits)
                q(9):                           100,000,000   (    9 digits)
```

Below are the continued fraction partial quotients and convergents of the reciprocal of the rational number to be approximated. These are necessary to economically locate the immediate Farey neighbors along the “top wall” of the rectangular area of the integer lattice, if this proves necessary.

```
------------------------------------------------------------------------------
***************   Continued Fraction Expansion Of k_in/h_in    ***************
------------------------------------------------------------------------------
************************   Inputs To CF Calculation   ************************
------------------------------------------------------------------------------
                h_in:                           100,000,000   (    9 digits)
------------------------------------------------------------------------------
                k_in:                            31,830,989   (    8 digits)
------------------------------------------------------------------------------
**************************   CF Partial Quotients   **************************
------------------------------------------------------------------------------
```

```
                a(0):                                     3   (    1 digit)
------------------------------------------------------------------------------
                a(1):                                     7   (    1 digit)
------------------------------------------------------------------------------
                a(2):                                    15   (    2 digits)
------------------------------------------------------------------------------
                a(3):                                     1   (    1 digit)
------------------------------------------------------------------------------
                a(4):                                   256   (    3 digits)
------------------------------------------------------------------------------
                a(5):                                     3   (    1 digit)
------------------------------------------------------------------------------
                a(6):                                     5   (    1 digit)
------------------------------------------------------------------------------
                a(7):                                     5   (    1 digit)
------------------------------------------------------------------------------
                a(8):                                    13   (    2 digits)
------------------------------------------------------------------------------
*****************************   CF Convergents   *****************************
------------------------------------------------------------------------------
                p(0):                                     3   (    1 digit)
                q(0):                                     1   (    1 digit)
------------------------------------------------------------------------------
                p(1):                                    22   (    2 digits)
                q(1):                                     7   (    1 digit)
------------------------------------------------------------------------------
                p(2):                                   333   (    3 digits)
                q(2):                                   106   (    3 digits)
------------------------------------------------------------------------------
                p(3):                                   355   (    3 digits)
                q(3):                                   113   (    3 digits)
------------------------------------------------------------------------------
                p(4):                                91,213   (    5 digits)
                q(4):                                29,034   (    5 digits)
------------------------------------------------------------------------------
                p(5):                               273,994   (    6 digits)
                q(5):                                87,215   (    5 digits)
------------------------------------------------------------------------------
                p(6):                             1,461,183   (    7 digits)
                q(6):                               465,109   (    6 digits)
------------------------------------------------------------------------------
                p(7):                             7,579,909   (    7 digits)
                q(7):                             2,412,760   (    7 digits)
------------------------------------------------------------------------------
                p(8):                           100,000,000   (    9 digits)
                q(8):                            31,830,989   (    8 digits)
```

Below are the continued fraction partial quotients and convergents of the corner point [i.e. (k_{MAX}, h_{MAX})]. These are necessary to economically locate the immediate Farey neighbor just to the left on the number line, along the “right wall” of the rectangular region of the integer lattice.

```
------------------------------------------------------------------------------
**************   Continued Fraction Expansion Of Corner Point   **************
------------------------------------------------------------------------------
************************   Inputs To CF Calculation   ************************
------------------------------------------------------------------------------
                h_in:                                   193   (    3 digits)
------------------------------------------------------------------------------
                k_in:                                   500   (    3 digits)
------------------------------------------------------------------------------
**************************   CF Partial Quotients   **************************
------------------------------------------------------------------------------
                a(0):                                     0   (    1 digit)
------------------------------------------------------------------------------
                a(1):                                     2   (    1 digit)
------------------------------------------------------------------------------
                a(2):                                     1   (    1 digit)
------------------------------------------------------------------------------
                a(3):                                     1   (    1 digit)
------------------------------------------------------------------------------
                a(4):                                     2   (    1 digit)
------------------------------------------------------------------------------
                a(5):                                     3   (    1 digit)
------------------------------------------------------------------------------
                a(6):                                     1   (    1 digit)
```

```
-------------------------------------------------------------------------------
                a(7):                                      8   (    1 digit)
-------------------------------------------------------------------------------
*****************************   CF Convergents   *****************************
-------------------------------------------------------------------------------
                p(0):                                      0   (    1 digit)
                q(0):                                      1   (    1 digit)
-------------------------------------------------------------------------------
                p(1):                                      1   (    1 digit)
                q(1):                                      2   (    1 digit)
-------------------------------------------------------------------------------
                p(2):                                      1   (    1 digit)
                q(2):                                      3   (    1 digit)
-------------------------------------------------------------------------------
                p(3):                                      2   (    1 digit)
                q(3):                                      5   (    1 digit)
-------------------------------------------------------------------------------
                p(4):                                      5   (    1 digit)
                q(4):                                     13   (    2 digits)
-------------------------------------------------------------------------------
                p(5):                                     17   (    2 digits)
                q(5):                                     44   (    2 digits)
-------------------------------------------------------------------------------
                p(6):                                     22   (    2 digits)
                q(6):                                     57   (    2 digits)
-------------------------------------------------------------------------------
                p(7):                                    193   (    3 digits)
                q(7):                                    500   (    3 digits)
```

Below are the continued fraction partial quotients and convergents of the reciprocal of the corner point [i.e. (k_{MAX}, h_{MAX})]. These are necessary to economically locate the immediate Farey neighbor just to the right on the number line, along the “top wall” of the rectangular region of the integer lattice.

```
-------------------------------------------------------------------------------
********   Continued Fraction Expansion Of Corner Point Reciprocal    ********
-------------------------------------------------------------------------------
************************   Inputs To CF Calculation   ************************
-------------------------------------------------------------------------------
                h_in:                                    500   (    3 digits)
-------------------------------------------------------------------------------
                k_in:                                    193   (    3 digits)
-------------------------------------------------------------------------------
**************************   CF Partial Quotients   **************************
-------------------------------------------------------------------------------
                a(0):                                      2   (    1 digit)
-------------------------------------------------------------------------------
                a(1):                                      1   (    1 digit)
-------------------------------------------------------------------------------
                a(2):                                      1   (    1 digit)
-------------------------------------------------------------------------------
                a(3):                                      2   (    1 digit)
-------------------------------------------------------------------------------
                a(4):                                      3   (    1 digit)
-------------------------------------------------------------------------------
                a(5):                                      1   (    1 digit)
-------------------------------------------------------------------------------
                a(6):                                      8   (    1 digit)
-------------------------------------------------------------------------------
*****************************   CF Convergents   *****************************
-------------------------------------------------------------------------------
                p(0):                                      2   (    1 digit)
                q(0):                                      1   (    1 digit)
-------------------------------------------------------------------------------
                p(1):                                      3   (    1 digit)
                q(1):                                      1   (    1 digit)
-------------------------------------------------------------------------------
                p(2):                                      5   (    1 digit)
                q(2):                                      2   (    1 digit)
-------------------------------------------------------------------------------
                p(3):                                     13   (    2 digits)
                q(3):                                      5   (    1 digit)
-------------------------------------------------------------------------------
                p(4):                                     44   (    2 digits)
                q(4):                                     17   (    2 digits)
-------------------------------------------------------------------------------
```

```
                p(5):                                    57   (    2 digits)
                q(5):                                    22   (    2 digits)
------------------------------------------------------------------------------
                p(6):                                   500   (    3 digits)
                q(6):                                   193   (    3 digits)
------------------------------------------------------------------------------
*************************   Corner Point Neighbors   *************************
------------------------------------------------------------------------------
       corner_pred_h:                                    22   (    2 digits)
       corner_pred_k:                                    57   (    2 digits)
------------------------------------------------------------------------------
       corner_succ_h:                                   171   (    3 digits)
       corner_succ_k:                                   443   (    3 digits)
------------------------------------------------------------------------------
```

The lines below advise that the rational number to approximated is not in the rectangular area of the integer lattice (i.e. can't be represented subject to the constraints). (This is the most typical case, as we seldom seek neighbors to a number we can represent exactly.) This means that none of the neighbors will be subscripted "0".

```
******************************************************************************
********   h_in/k_in IS NOT In Rectangular Farey Series Of Interest   ********
******************************************************************************
```

The statement below advises that the rational number to be approximated is less than h_{MAX}/k_{MAX} (i.e. to the left of the corner point) in the rectangular area of the integer lattice formed by the constraints. All boundary cases are covered when the neighbors span the corner.

```
------------------------------------------------------------------------------
***********************   0 <= h_in/k_in < hmax/kmax   ***********************
------------------------------------------------------------------------------
*****************   Highest-Order Convergent With q(i)<=N    *****************
------------------------------------------------------------------------------
                p(4):                                   113   (    3 digits)
                q(4):                                   355   (    3 digits)
------------------------------------------------------------------------------
*******************   Accompanying Intermediate Fraction   *******************
------------------------------------------------------------------------------
      intermediate_h:                                   106   (    3 digits)
      intermediate_k:                                   333   (    3 digits)
```

Below is a typical neighbor. The subscript of "-2" indicates it is the second neighbor to the left on the number line from the number to be approximated, subject to the constraints.

```
------------------------------------------------------------------------------
******************   Rectangular Farey Neighbor Index -2    ******************
------------------------------------------------------------------------------
               h(-2):                                   120   (    3 digits)
------------------------------------------------------------------------------
               k(-2):                                   377   (    3 digits)
```

Below, the number 120/377 is rephrased as a rational number with a larger denominator which is a power of ten. This says that this number is 0.318302…

```
------------------------------------------------------------------------------
           DAP_N(-2):   318,302,387,267,904,509,283,819,628,  (   54 digits)
                        647,214,854,111,405,835,543,766,578
------------------------------------------------------------------------------
           DAP_D(-2):                                     1,  (   55 digits)
                        000,000,000,000,000,000,000,000,000,
                        000,000,000,000,000,000,000,000,000
```

The error (the difference between the approximation and the number to be approximated) is supplied as a rational number.

```
------------------------------------------------------------------------------
         error_h(-2): -                             282,853   (    6 digits)
------------------------------------------------------------------------------
         error_k(-2):                        37,700,000,000   (   11 digits)
```

The error just above is restated with a power-of-ten denominator. This says that the error is about -0.0000075…

```
------------------------------------------------------------------------------
     ERROR_DAP_N(-2): -       7,502,732,095,490,716,180,371,  (   49 digits)
                        352,785,145,888,594,164,456,233,421
------------------------------------------------------------------------------
```

```
     ERROR_DAP_D(-2):                                     1,  (   55 digits)
                        000,000,000,000,000,000,000,000,000,
                        000,000,000,000,000,000,000,000,000
```

The same information described above is repeated for each neighbor.

```
-------------------------------------------------------------------------------
******************   Rectangular Farey Neighbor Index -1    *******************
-------------------------------------------------------------------------------
               h(-1):                                   113   (    3 digits)
-------------------------------------------------------------------------------
               k(-1):                                   355   (    3 digits)
-------------------------------------------------------------------------------
           DAP_N(-1):   318,309,859,154,929,577,464,788,732,  (   54 digits)
                        394,366,197,183,098,591,549,295,774
-------------------------------------------------------------------------------
           DAP_D(-1):                                     1,  (   55 digits)
                        000,000,000,000,000,000,000,000,000,
                        000,000,000,000,000,000,000,000,000
-------------------------------------------------------------------------------
         error_h(-1): -                                 219   (    3 digits)
-------------------------------------------------------------------------------
         error_k(-1):                         7,100,000,000   (   10 digits)
-------------------------------------------------------------------------------
     ERROR_DAP_N(-1): -          30,845,070,422,535,211,267,  (   47 digits)
                        605,633,802,816,901,408,450,704,225
-------------------------------------------------------------------------------
     ERROR_DAP_D(-1):                                     1,  (   55 digits)
                        000,000,000,000,000,000,000,000,000,
                        000,000,000,000,000,000,000,000,000
-------------------------------------------------------------------------------
*******************   Rectangular Farey Neighbor Index 1   ********************
-------------------------------------------------------------------------------
                h(1):                                   106   (    3 digits)
-------------------------------------------------------------------------------
                k(1):                                   333   (    3 digits)
-------------------------------------------------------------------------------
            DAP_N(1):   318,318,318,318,318,318,318,318,318,  (   54 digits)
                        318,318,318,318,318,318,318,318,318
-------------------------------------------------------------------------------
            DAP_D(1):                                     1,  (   55 digits)
                        000,000,000,000,000,000,000,000,000,
                        000,000,000,000,000,000,000,000,000
-------------------------------------------------------------------------------
          error_h(1):                               280,663   (    6 digits)
-------------------------------------------------------------------------------
          error_k(1):                        33,300,000,000   (   11 digits)
-------------------------------------------------------------------------------
      ERROR_DAP_N(1):         8,428,318,318,318,318,318,318,  (   49 digits)
                        318,318,318,318,318,318,318,318,318
-------------------------------------------------------------------------------
      ERROR_DAP_D(1):                                     1,  (   55 digits)
                        000,000,000,000,000,000,000,000,000,
                        000,000,000,000,000,000,000,000,000
-------------------------------------------------------------------------------
*******************   Rectangular Farey Neighbor Index 2   ********************
-------------------------------------------------------------------------------
                h(2):                                    99   (    2 digits)
-------------------------------------------------------------------------------
                k(2):                                   311   (    3 digits)
-------------------------------------------------------------------------------
            DAP_N(2):   318,327,974,276,527,331,189,710,610,  (   54 digits)
                        932,475,884,244,372,990,353,697,749
-------------------------------------------------------------------------------
            DAP_D(2):                                     1,  (   55 digits)
                        000,000,000,000,000,000,000,000,000,
                        000,000,000,000,000,000,000,000,000
-------------------------------------------------------------------------------
          error_h(2):                               562,421   (    6 digits)
-------------------------------------------------------------------------------
          error_k(2):                        31,100,000,000   (   11 digits)
-------------------------------------------------------------------------------
      ERROR_DAP_N(2):        18,084,276,527,331,189,710,610,  (   50 digits)
                        932,475,884,244,372,990,353,697,749
-------------------------------------------------------------------------------
      ERROR_DAP_D(2):                                     1,  (   55 digits)
```

```
                        000,000,000,000,000,000,000,000,000,
                        000,000,000,000,000,000,000,000,000
------------------------------------------------------------------------------
RAP execution ends.
------------------------------------------------------------------------------
```

4.19 Upper Bound On Distance Between Members Of $F_{A,B}$ In An Interval [FABDMAX]

4.19.1 Command Line Invocation Forms

rap fab LL UL HMAX KMAX

Calculates an upper bound on the maximum distance between terms of $F_{KMAX,HMAX}$ in the interval [LL,UL] using the techniques developed in the TOMS paper. LL and UL must be non-negative, and must both be in $F_{KMAX,HMAX}$. HMAX and KMAX must both be positive.

4.19.2 Detailed Algorithm Description

The TOMS paper develops several ways of bounding the maximum distance between terms in $F_{KMAX,HMAX}$, and which is applicable depends on whether HMAX is the dominant constraint, or KMAX is the dominant constraint. The RAP program output clearly explains how it calculates an upper bound on the distance between terms in $F_{KMAX,HMAX}$.

4.19.3 Example Invocation

The example invocation below demonstrates error bounds over the interval [0.385, 2.160] with $h_{MAX} \leq 193$ and $k_{MAX} \leq 500$. This is the same example used in the TOMS paper. Explanatory remarks are included in a different font and with shading.

```
c:\>rap fabdmax 0.385 2.160 193 500
------------------------------------------------------------------------------
RAP ($Revision: 15 $ $Date: 11/13/00 3:59a $) execution begins.
------------------------------------------------------------------------------
```

The section below just echoes the command-line or batch inputs.

```
******************************************************************************
******************************************************************************
***************************    Input Parameters   ****************************
******************************************************************************
******************************************************************************
------------------------------------------------------------------------------
                 l_h:                                    77   (    2 digits)
------------------------------------------------------------------------------
                 l_k:                                   200   (    3 digits)
------------------------------------------------------------------------------
                 r_h:                                    54   (    2 digits)
------------------------------------------------------------------------------
                 r_k:                                    25   (    2 digits)
------------------------------------------------------------------------------
               h_max:                                   193   (    3 digits)
------------------------------------------------------------------------------
               k_max:                                   500   (    3 digits)
------------------------------------------------------------------------------
               dap_D:                                     1,  (  109 digits)
                        000,000,000,000,000,000,000,000,000,
                        000,000,000,000,000,000,000,000,000,
                        000,000,000,000,000,000,000,000,000,
                        000,000,000,000,000,000,000,000,000
------------------------------------------------------------------------------
******************************************************************************
******************************************************************************
*************************   Case Selection Results   *************************
******************************************************************************
******************************************************************************
------------------------------------------------------------------------------
```

The program decides which cases should be evaluated to give an error bound. Even though the highest-numbered case will always give the larger bounds, unnecessary cases which apply are evaluated in case the user is interested in different upper bounds over different portions of the interval.

```
Case I applies and will be evaluated.
Case II applies and will be evaluated.
Case III applies and will be evaluated.
------------------------------------------------------------------------------
******************************************************************************
******************************************************************************
************************   Start Of Case I Results    ************************
******************************************************************************
******************************************************************************
```

These are the Case I results. Note that the right endpoint of the interval is truncated at HMAX/KMAX.

```
------------------------------------------------------------------------------
*****************   Possibly Truncated Interval For Case I   *****************
------------------------------------------------------------------------------
                 l_h:                                    77   (    2 digits)
------------------------------------------------------------------------------
                 l_k:                                   200   (    3 digits)
------------------------------------------------------------------------------
                 r_h:                                   193   (    3 digits)
------------------------------------------------------------------------------
                 r_k:                                   500   (    3 digits)
------------------------------------------------------------------------------
          midpoint_h:                                   771   (    3 digits)
------------------------------------------------------------------------------
          midpoint_k:                                 2,000   (    4 digits)
------------------------------------------------------------------------------
*****************   CF Representation Of Interval Midpoint   *****************
************************   Inputs To CF Calculation   ************************
------------------------------------------------------------------------------
                h_in:                                   771   (    3 digits)
------------------------------------------------------------------------------
                k_in:                                 2,000   (    4 digits)
------------------------------------------------------------------------------
**************************   CF Partial Quotients   **************************
------------------------------------------------------------------------------
                a(0):                                     0   (    1 digit)
------------------------------------------------------------------------------
                a(1):                                     2   (    1 digit)
------------------------------------------------------------------------------
                a(2):                                     1   (    1 digit)
------------------------------------------------------------------------------
                a(3):                                     1   (    1 digit)
------------------------------------------------------------------------------
                a(4):                                     2   (    1 digit)
------------------------------------------------------------------------------
                a(5):                                     6   (    1 digit)
------------------------------------------------------------------------------
                a(6):                                     3   (    1 digit)
------------------------------------------------------------------------------
                a(7):                                     3   (    1 digit)
------------------------------------------------------------------------------
                a(8):                                     2   (    1 digit)
------------------------------------------------------------------------------
*****************************   CF Convergents   *****************************
------------------------------------------------------------------------------
                p(0):                                     0   (    1 digit)
                q(0):                                     1   (    1 digit)
------------------------------------------------------------------------------
                p(1):                                     1   (    1 digit)
                q(1):                                     2   (    1 digit)
------------------------------------------------------------------------------
                p(2):                                     1   (    1 digit)
                q(2):                                     3   (    1 digit)
------------------------------------------------------------------------------
                p(3):                                     2   (    1 digit)
                q(3):                                     5   (    1 digit)
------------------------------------------------------------------------------
                p(4):                                     5   (    1 digit)
                q(4):                                    13   (    2 digits)
------------------------------------------------------------------------------
```

```
                p(5):                                    32   (    2 digits)
                q(5):                                    83   (    2 digits)
------------------------------------------------------------------------------
                p(6):                                   101   (    3 digits)
                q(6):                                   262   (    3 digits)
------------------------------------------------------------------------------
                p(7):                                   335   (    3 digits)
                q(7):                                   869   (    3 digits)
------------------------------------------------------------------------------
                p(8):                                   771   (    3 digits)
                q(8):                                 2,000   (    4 digits)
------------------------------------------------------------------------------
********   A Rational Number With Smallest Denominator In Interval    ********
------------------------------------------------------------------------------
            result_h:                                    22   (    2 digits)
------------------------------------------------------------------------------
            result_k:                                    57   (    2 digits)
------------------------------------------------------------------------------
*****   Upper Bound On Distance Between Farey Terms As Rational Number   *****
------------------------------------------------------------------------------
          error_ub_h:                                     1   (    1 digit)
------------------------------------------------------------------------------
          error_up_k:                                25,251   (    5 digits)
------------------------------------------------------------------------------
**   Upper Bound On Distance Between Farey Terms As Decimal Approximation   **
------------------------------------------------------------------------------
               dap_h:        39,602,391,984,475,862,342,085,  (  104 digits)
                        461,961,902,498,910,934,220,426,913,
                        785,592,649,796,047,681,279,949,308,
                        938,259,870,896,202,130,608,688,764
------------------------------------------------------------------------------
               dap_k:                                     1,  (  109 digits)
                        000,000,000,000,000,000,000,000,000,
                        000,000,000,000,000,000,000,000,000,
                        000,000,000,000,000,000,000,000,000,
                        000,000,000,000,000,000,000,000,000
------------------------------------------------------------------------------
******************************************************************************
******************************************************************************
************************   Start Of Case II Results   ************************
******************************************************************************
******************************************************************************
```

These are the Case II results.

```
------------------------------------------------------------------------------
**************************   Case II Right Limit    **************************
------------------------------------------------------------------------------
      case_2_right_h:                                     1   (    1 digit)
------------------------------------------------------------------------------
      case_2_right_k:                                     1   (    1 digit)
------------------------------------------------------------------------------
*****   Upper Bound On Distance Between Farey Terms As Rational Number   *****
------------------------------------------------------------------------------
          error_ub_h:                                     1   (    1 digit)
------------------------------------------------------------------------------
          error_up_k:                                   194   (    3 digits)
------------------------------------------------------------------------------
**   Upper Bound On Distance Between Farey Terms As Decimal Approximation   **
------------------------------------------------------------------------------
               dap_h:     5,154,639,175,257,731,958,762,886,  (  106 digits)
                        597,938,144,329,896,907,216,494,845,
                        360,824,742,268,041,237,113,402,061,
                        855,670,103,092,783,505,154,639,175
------------------------------------------------------------------------------
               dap_k:                                     1,  (  109 digits)
                        000,000,000,000,000,000,000,000,000,
                        000,000,000,000,000,000,000,000,000,
                        000,000,000,000,000,000,000,000,000,
                        000,000,000,000,000,000,000,000,000
------------------------------------------------------------------------------
******************************************************************************
******************************************************************************
***********************   Start Of Case III Results    ***********************
******************************************************************************
******************************************************************************
```

These are the Case III results.

```
------------------------------------------------------------------------------
*****   Upper Bound On Distance Between Farey Terms As Rational Number   *****
------------------------------------------------------------------------------
          error_ub_h:                                     1   (    1 digit)
------------------------------------------------------------------------------
          error_up_k:                                    89   (    2 digits)
------------------------------------------------------------------------------
**   Upper Bound On Distance Between Farey Terms As Decimal Approximation   **
------------------------------------------------------------------------------
               dap_h:    11,235,955,056,179,775,280,898,876,  (  107 digits)
                        404,494,382,022,471,910,112,359,550,
                        561,797,752,808,988,764,044,943,820,
                        224,719,101,123,595,505,617,977,528
------------------------------------------------------------------------------
               dap_k:                                     1,  (  109 digits)
                        000,000,000,000,000,000,000,000,000,
                        000,000,000,000,000,000,000,000,000,
                        000,000,000,000,000,000,000,000,000,
                        000,000,000,000,000,000,000,000,000
------------------------------------------------------------------------------
******************************************************************************
******************************************************************************
**********************   Start Of Cumulative Results    **********************
******************************************************************************
******************************************************************************
------------------------------------------------------------------------------
```

These are the cumulative results. The largest error upper bound anywhere over the interval is supplied.

```
******************************************************************************
******************************************************************************
************************   Largest Error (Case III)   ************************
******************************************************************************
******************************************************************************
------------------------------------------------------------------------------
*****   Upper Bound On Distance Between Farey Terms As Rational Number   *****
------------------------------------------------------------------------------
          error_ub_h:                                     1   (    1 digit)
------------------------------------------------------------------------------
          error_up_k:                                    89   (    2 digits)
------------------------------------------------------------------------------
**   Upper Bound On Distance Between Farey Terms As Decimal Approximation   **
------------------------------------------------------------------------------
               dap_h:    11,235,955,056,179,775,280,898,876,  (  107 digits)
                        404,494,382,022,471,910,112,359,550,
                        561,797,752,808,988,764,044,943,820,
                        224,719,101,123,595,505,617,977,528
------------------------------------------------------------------------------
               dap_k:                                     1,  (  109 digits)
                        000,000,000,000,000,000,000,000,000,
                        000,000,000,000,000,000,000,000,000,
                        000,000,000,000,000,000,000,000,000,
                        000,000,000,000,000,000,000,000,000
------------------------------------------------------------------------------
RAP execution ends.
------------------------------------------------------------------------------
```

5. Digits Of Useful Constants

This section contains base-10 rational approximations of useful constants to many decimal places. These can be manually entered or pasted into files used as RAP input.

5.1 Digits Of p

The first 1,000 digits of π were obtained from the web site *http://www.ex.ac.uk/cimt/general/pi10000.htm.* The numbers in bold and in brackets show how many digits have been presented up to that point. This information has not been verified for accuracy.

```
3.1415926535   8979323846   2643383279   5028841971   6939937510
  5820974944   5923078164   0628620899   8628034825   3421170679
  8214808651   3282306647   0938446095   5058223172   5359408128
  4811174502   8410270193   8521105559   6446229489   5493038196
  4428810975   6659334461   2847564823   3786783165   2712019091
  4564856692   3460348610   4543266482   1339360726   0249141273
  7245870066   0631558817   4881520920   9628292540   9171536436
  7892590360   0113305305   4882046652   1384146951   9415116094
  3305727036   5759591953   0921861173   8193261179   3105118548
  0744623799   6274956735   1885752724   8912279381   8301194912
[500]
  9833673362   4406566430   8602139494   6395224737   1907021798
  6094370277   0539217176   2931767523   8467481846   7669405132
  0005681271   4526356082   7785771342   7577896091   7363717872
  1468440901   2249534301   4654958537   1050792279   6892589235
  4201995611   2129021960   8640344181   5981362977   4771309960
  5187072113   4999999837   2978049951   0597317328   1609631859
  5024459455   3469083026   4252230825   3344685035   2619311881
  7101000313   7838752886   5875332083   8142061717   7669147303
  5982534904   2875546873   1159562863   8823537875   9375195778
  1857780532   1712268066   1300192787   6611195909   2164201989
[1000]
```

5.2 Digits Of e

The first 1,000 digits of *e* were obtained from the web site *http://fermi.udw.ac.za/physics/e.html*. This information has not been verified for accuracy.

```
2.7182818284590452353602874713526624977572470936999595749669676277240766303535
475945713821785251664274274663919320030599218174135966290435729003342952605956
307381323286279434907632338298807531952510190115738341879307021540891499348841
675092447614606680822648001684774118537423454424371075390777449920695517027618
386062613313845830007520449338265602976067371132007093287091274437470472306969
772093101416928368190255151086574637721112523897844250569536967707854499699679
468644549059879316368892300987931277361782154249992295763514822082698951936680
331825288693984964651058209392398294887933203625094431173012381970684161403970
198376793206832823764648042953118023287825098194558153017567173613320698112509
961818815930416903515988885193458072738667385894228792284998920868058257492796
104841984443634632449684875602336248270419786232090021609902353043699418491463
140934317381436405462531520961836908887070167683964243781405927145635490613031
07208510383750510115747704171898610687396965521267154688957035035
```