**** OUTPUT **********

********Welcome to Online IDE!! Happy Coding :)**********

NOTE: Precision is limited to 19 digits wide...

NOTE: Any decimal with recurring single digit(19 times) will be assumed to be periodic number

NOTE: Since computational issues, reduction on decimal component EXCEEDING 9(9 digits) will lose its precision by 1 or more digit depending on the decimal.

NOTE: Value of acceptedNumeratorLimit = 9 digits. This can be changed to speed up most reduced form calculations

NOTE: Also to AVOID any Session kills or hanging

To create sample of decimals and fractions, original numerator will be divided numerator/prime number from: 2 => 5

DENOMINATOR AND NUMERATOR WILL BOTH BE TRUNCATED INLINE WITH INTEGRITY ORIGINAL INPUTS

****

is a prime number: 2

is a prime number: 3

is not a prime number: 4

is a prime number: 5

*** The Decimal presented: 5.2

The fraction part: 2

this is numerator: 2

this is denominator: 10

This is potential Greatest Common Factor(2) between Numerator(2) Denominator(10)

Reduction numerator: (2,1 divisor: 2)

Reduction denominator: (10,5 divisor: 2)

Reduction in fraction and decimal: 2/10(0.2) => 1/5(0.2)

Elapsed time in seconds to reduce fraction: 0.87903818

reduced numerator: 1

reduced denominator: 5

Fraction presented is: 26/5

No whole number before fraction

Original whole number from fraction (numerator/denominator):5

remaining numerator: 1

remaining fraction: 1/5

Fraction provided in challenge is improper: 26/5

Mixed number fraction is: 5-1/5

1Mixed number fraction(5) and original decimal(5) have same whole number

fraction provided in challenge is fully reduced: 1/5

Original fraction(1/5) provided is exact representation of the initial decimal conversion: 5.2

****

This will just be final part of the code...

Since several prime numbers have been generated, I will use the existing numerator

And store the results in a String array: Numerator/prime number AS fraction and Numerator/prime number AS decimal and store results. IT IS TO GET A FEEL OF DIFFERENT PATTERNS AS EXPLORED IN DOCUMENTATION!

This will provide a sample should I wish to populate initial array also!

It will also show examples in which repetends have occurred two or more times in the decimal precision (19 digits wide)

THIS IS ONLY circumstances available to enhance my code to utilize mapping to fractional equivalent....

****

Original numerator / PRIME number AS FRACTION (not reduced) => 26/2

Original numerator / PRIME number AS DECIMAL => 13.0

Original numerator / PRIME number AS FRACTION (not reduced) => 26/3

Original numerator / PRIME number AS DECIMAL => 8.666666666666666

Original numerator / PRIME number AS FRACTION (not reduced) => 26/5

Original numerator / PRIME number AS DECIMAL => 5.2

****

CHECK TO SEE IF ORIGINAL FRACTION IS TERMINATING:

Terminating fraction since reduced denominator(5) has prime factor (divisible by prime number of two or five)

Original decimal:5.2 is a periodic number

*** The Decimal presented: 7.1111111

The fraction part: 1111111

this is numerator: 1111111

this is denominator: 10000000

This is potential Greatest Common Factor(1111111) between Numerator(1111111) Denominator(10000000)

Elapsed time in seconds to reduce fraction: 1.202937871

reduced numerator: 1111111

reduced denominator: 10000000

Fraction presented is: 22/1999

No whole number before fraction

1This is the difference(0.10010559724862432) between original(0.1111111) decimal and associated fraction(22/1999) in decimal(0.011005502751375688)

2Original fraction (22/1999) is non- representation of the initial decimal conversion: 0.1111111(1111111/10000000)

****

This will just be final part of the code...

Since several prime numbers have been generated, I will use the existing numerator

And store the results in a String array: Numerator/prime number AS fraction and Numerator/prime number AS decimal and store results. IT IS TO GET A FEEL OF DIFFERENT PATTERNS AS EXPLORED IN DOCUMENTATION!

This will provide a sample should I wish to populate initial array also!

It will also show examples in which repetends have occurred two or more times in the decimal precision (19 digits wide)

THIS IS ONLY circumstances available to enhance my code to utilize mapping to fractional equivalent....

****

Original numerator / PRIME number AS FRACTION (not reduced) => 22/2

Original numerator / PRIME number AS DECIMAL => 11.0

Original numerator / PRIME number AS FRACTION (not reduced) => 22/3

Original numerator / PRIME number AS DECIMAL => 7.333333333333333

Original numerator / PRIME number AS FRACTION (not reduced) => 22/5

Original numerator / PRIME number AS DECIMAL => 4.4

****

CHECK TO SEE IF ORIGINAL FRACTION IS TERMINATING:

Terminating fraction since reduced denominator(10000000) has prime factor (divisible by prime number of two or five)

Original decimal:7.1111111 is a periodic number

*** The Decimal presented: 0.192367

The fraction part: 192367

this is numerator: 192367

this is denominator: 1000000

This is potential Greatest Common Factor(192367) between Numerator(192367) Denominator(1000000)

Elapsed time in seconds to reduce fraction: 0.303816441

reduced numerator: 192367

reduced denominator: 1000000

Fraction presented is: 64/9

No whole number before fraction

Original whole number from fraction (numerator/denominator):7

remaining numerator: 1

remaining fraction: 1/9

Fraction provided in challenge is improper: 64/9

Mixed number fraction is: 7-1/9

Mismatch in whole numbers*

Mixed number fraction has following whole number:7

Original decimal has following whole number:0

****

1Original fraction(64/9)has been changed to mixed number fraction(7-1/9)

1This is the difference(0.0812558888888889) between original(0.192367) decimal and associated fraction(1/9) in decimal(0.1111111111111111)

2Original fraction (1/9) is non- representation of the initial decimal conversion: 0.192367(192367/1000000)

Original fraction(1/9) is NOT reduced identically to the converted decimal (most reduced form): (192367/1000000)

1Remaining fraction(1/9) is non- representation of the initial decimal conversion: 0.192367

1Remaining fraction (1/9) in decimal is 0.1111111111111111 and NOT:0.192367(original)

2This is the difference(0.0812558888888889) between original(0.192367) decimal and associated mixed number remaining fraction(1/9)

****

This will just be final part of the code...

Since several prime numbers have been generated, I will use the existing numerator

And store the results in a String array: Numerator/prime number AS fraction and Numerator/prime number AS decimal and store results. IT IS TO GET A FEEL OF DIFFERENT PATTERNS AS EXPLORED IN DOCUMENTATION!

This will provide a sample should I wish to populate initial array also!

It will also show examples in which repetends have occurred two or more times in the decimal precision (19 digits wide)

THIS IS ONLY circumstances available to enhance my code to utilize mapping to fractional equivalent....

****

Original numerator / PRIME number AS FRACTION (not reduced) => 64/2

Original numerator / PRIME number AS DECIMAL => 32.0

Original numerator / PRIME number AS FRACTION (not reduced) => 64/3

Original numerator / PRIME number AS DECIMAL => 21.333333333333332

Original numerator / PRIME number AS FRACTION (not reduced) => 64/5

Original numerator / PRIME number AS DECIMAL => 12.8

****

CHECK TO SEE IF ORIGINAL FRACTION IS TERMINATING:

Terminating fraction since reduced denominator(1000000) has prime factor (divisible by prime number of two or five)

Original decimal:0.192367 is a periodic number

*** The Decimal presented: 0.2228

The fraction part: 2228

this is numerator: 2228

this is denominator: 10000

This is potential Greatest Common Factor(2228) between Numerator(2228) Denominator(10000)

Reduction numerator: (2228,1114 divisor: 2)

Reduction denominator: (10000,5000 divisor: 2)

Reduction in fraction and decimal: 2228/10000(0.2228) => 1114/5000(0.2228)

Reduction numerator: (1114,557 divisor: 2)

Reduction denominator: (5000,2500 divisor: 2)

Reduction in fraction and decimal: 1114/5000(0.2228) => 557/2500(0.2228)

Elapsed time in seconds to reduce fraction: 4.002E-4

reduced numerator: 557

reduced denominator: 2500

Fraction presented is: 5343/27775

No whole number before fraction

1This is the difference(0.03043276327632763) between original(0.2228) decimal and associated fraction(5343/27775) in decimal(0.19236723672367237)

2Original fraction (5343/27775) is non- representation of the initial decimal conversion: 0.2228(557/2500)

****

This will just be final part of the code...

Since several prime numbers have been generated, I will use the existing numerator

And store the results in a String array: Numerator/prime number AS fraction and Numerator/prime number AS decimal and store results. IT IS TO GET A FEEL OF DIFFERENT PATTERNS AS EXPLORED IN DOCUMENTATION!

This will provide a sample should I wish to populate initial array also!

It will also show examples in which repetends have occurred two or more times in the decimal precision (19 digits wide)

THIS IS ONLY circumstances available to enhance my code to utilize mapping to fractional equivalent....

****

Original numerator / PRIME number AS FRACTION (not reduced) => 5343/2

Original numerator / PRIME number AS DECIMAL => 2671.5

Original numerator / PRIME number AS FRACTION (not reduced) => 5343/3

Original numerator / PRIME number AS DECIMAL => 1781.0

Original numerator / PRIME number AS FRACTION (not reduced) => 5343/5

Original numerator / PRIME number AS DECIMAL => 1068.6

****

CHECK TO SEE IF ORIGINAL FRACTION IS TERMINATING:

Terminating fraction since reduced denominator(2500) has prime factor (divisible by prime number of two or five)

Original decimal:0.2228 is a periodic number

*** The Decimal presented: 0.777777777777778777

The fraction part: 777777777777778777

this is numerator: 777777777777778777

this is denominator: 1000000000000000000

6Numerator truncated from: 18 to 9 digits:

7Denominator truncated from 19 to 10 digits

Truncated denominator: 1000000000

Truncated numerator: 777777777

2Single digit recurring:false

2THIS IS THE REGEX TO CHECK FOR POORLY DIVISIBLE NUMERATORS :777777777

THIS IS THE numerator:777777777

3Numerator truncated to: 8 digits (77777777) to mitigate resource limitations of recurrence 9 times (1,3,7,9)

4Denominator truncated to: 9 digits(100000000)inline with numerator

This is potential Greatest Common Factor(77777777) between Numerator(77777777) Denominator(100000000)

Elapsed time in seconds to reduce fraction: 14.55571082

reduced numerator: 77777777

reduced denominator: 100000000

Fraction presented is: 222/100

No whole number before fraction

Original whole number from fraction (numerator/denominator):2

remaining numerator: 22

remaining fraction: 22/100

Fraction provided in challenge is improper: 222/100

Mixed number fraction is: 2-22/100

Mismatch in whole numbers*

Mixed number fraction has following whole number:2

Original decimal has following whole number:0

****

1Original fraction(222/100)has been changed to mixed number fraction(2-22/100)

1This is the difference(0.5577777777777788) between original(0.777777777777778777) decimal and associated fraction(22/100) in decimal(0.22)

2Original fraction (22/100) is non- representation of the initial decimal conversion: 0.77777777777778777(77777777/100000000)

Original fraction(22/100) is NOT reduced identically to the converted decimal (most reduced form): (77777777/100000000)

1Remaining fraction(22/100) is non- representation of the initial decimal conversion: 0.777777777777778777

1Remaining fraction (22/100) in decimal is 0.22 and NOT:0.77777777777778777(original)

2This is the difference(0.5577777777777788) between original(0.777777777777778777) decimal and associated mixed number remaining fraction(22/100)

****

This will just be final part of the code...

Since several prime numbers have been generated, I will use the existing numerator

And store the results in a String array: Numerator/prime number AS fraction and Numerator/prime number AS decimal and store results. IT IS TO GET A FEEL OF DIFFERENT PATTERNS AS EXPLORED IN DOCUMENTATION!

This will provide a sample should I wish to populate initial array also!

It will also show examples in which repetends have occurred two or more times in the decimal precision (19 digits wide)

THIS IS ONLY circumstances available to enhance my code to utilize mapping to fractional equivalent....

****

Original numerator / PRIME number AS FRACTION (not reduced) => 222/2

Original numerator / PRIME number AS DECIMAL => 111.0

Original numerator / PRIME number AS FRACTION (not reduced) => 222/3

Original numerator / PRIME number AS DECIMAL => 74.0

Original numerator / PRIME number AS FRACTION (not reduced) => 222/5

Original numerator / PRIME number AS DECIMAL => 44.4

****

CHECK TO SEE IF ORIGINAL FRACTION IS TERMINATING:

Terminating fraction since reduced denominator(100000000) has prime factor (divisible by prime number of two or five)

Original decimal:0.777777777777778777 is a periodic number

*** The Decimal presented: 3.14857

The fraction part: 14857

this is numerator: 14857

this is denominator: 100000

This is potential Greatest Common Factor(14857) between Numerator(14857) Denominator(100000)

Elapsed time in seconds to reduce fraction: 4.88239E-4

reduced numerator: 14857

reduced denominator: 100000

Fraction presented is: 987000200/100000020

No whole number before fraction

Original whole number from fraction (numerator/denominator):9

remaining numerator: 87000020

remaining fraction: 87000020/100000020

Fraction provided in challenge is improper: 987000200/100000020

Mixed number fraction is: 9-87000020/100000020

Mismatch in whole numbers*

Mixed number fraction has following whole number:9

Original decimal has following whole number:3

****

1Original fraction(987000200/100000020)has been changed to mixed number fraction(9-87000020/100000020)

1This is the difference(0.7214300259999948) between original(0.14857) decimal and associated fraction(87000020/100000020) in decimal(0.8700000259999948)

2Original fraction (87000020/100000020) is non- representation of the initial decimal conversion: 0.14857(14857/100000)

1Remaining fraction(87000020/100000020) is non- representation of the initial decimal conversion: 0.14857

1Remaining fraction (87000020/100000020) in decimal is 0.8700000259999948 and NOT:0.14857(original)

2This is the difference(0.7214300259999948) between original(0.14857) decimal and associated mixed number remaining fraction(87000020/100000020)

****

This will just be final part of the code...

Since several prime numbers have been generated, I will use the existing numerator

And store the results in a String array: Numerator/prime number AS fraction and Numerator/prime number AS decimal and store results. IT IS TO GET A FEEL OF DIFFERENT PATTERNS AS EXPLORED IN DOCUMENTATION!

This will provide a sample should I wish to populate initial array also!

It will also show examples in which repetends have occurred two or more times in the decimal precision (19 digits wide)

THIS IS ONLY circumstances available to enhance my code to utilize mapping to fractional equivalent....

****

Original numerator / PRIME number AS FRACTION (not reduced) => 987000200/2

Original numerator / PRIME number AS DECIMAL => 4.935001E8

Original numerator / PRIME number AS FRACTION (not reduced) => 987000200/3

Original numerator / PRIME number AS DECIMAL => 3.290000666666667E8

Original numerator / PRIME number AS FRACTION (not reduced) => 987000200/5

Original numerator / PRIME number AS DECIMAL => 1.9740004E8

****

CHECK TO SEE IF ORIGINAL FRACTION IS TERMINATING:

Terminating fraction since reduced denominator(100000) has prime factor (divisible by prime number of two or five)

Original decimal:3.14857 is a periodic number

*** The Decimal presented: 0.6

The fraction part: 6

this is numerator: 6

this is denominator: 10

This is potential Greatest Common Factor(6) between Numerator(6) Denominator(10)

Reduction numerator: (6,3 divisor: 2)

Reduction denominator: (10,5 divisor: 2)

Reduction in fraction and decimal: 6/10(0.6) => 3/5(0.6)

Elapsed time in seconds to reduce fraction: 0.004974285

reduced numerator: 3

reduced denominator: 5

Fraction presented is: 22/7

No whole number before fraction

Original whole number from fraction (numerator/denominator):3

remaining numerator: 1

remaining fraction: 1/7

Fraction provided in challenge is improper: 22/7

Mixed number fraction is: 3-1/7

Mismatch in whole numbers*

Mixed number fraction has following whole number:3

Original decimal has following whole number:0

****

1Original fraction(22/7)has been changed to mixed number fraction(3-1/7)

1This is the difference(0.45714285714285713) between original(0.6) decimal and associated fraction(1/7) in decimal(0.14285714285714285)

2Original fraction (1/7) is non- representation of the initial decimal conversion: 0.6(3/5)

Original fraction(1/7) is NOT reduced identically to the converted decimal (most reduced form): (3/5)

1Remaining fraction(1/7) is non- representation of the initial decimal conversion: 0.6

1Remaining fraction (1/7) in decimal is 0.14285714285714285 and NOT:0.6(original)

2This is the difference(0.45714285714285713) between original(0.6) decimal and associated mixed number remaining fraction(1/7)

****

This will just be final part of the code...

Since several prime numbers have been generated, I will use the existing numerator

And store the results in a String array: Numerator/prime number AS fraction and Numerator/prime number AS decimal and store results. IT IS TO GET A FEEL OF DIFFERENT PATTERNS AS EXPLORED IN DOCUMENTATION!

This will provide a sample should I wish to populate initial array also!

It will also show examples in which repetends have occurred two or more times in the decimal precision (19 digits wide)

THIS IS ONLY circumstances available to enhance my code to utilize mapping to fractional equivalent....

****

Original numerator / PRIME number AS FRACTION (not reduced) => 22/2

Original numerator / PRIME number AS DECIMAL => 11.0

Original numerator / PRIME number AS FRACTION (not reduced) => 22/3

Original numerator / PRIME number AS DECIMAL => 7.333333333333333

Original numerator / PRIME number AS FRACTION (not reduced) => 22/5

Original numerator / PRIME number AS DECIMAL => 4.4

****

CHECK TO SEE IF ORIGINAL FRACTION IS TERMINATING:

Terminating fraction since reduced denominator(5) has prime factor (divisible by prime number of two or five)

Original decimal:0.6 is a periodic number

*** The Decimal presented: 1.1

The fraction part: 1

this is numerator: 1

this is denominator: 10

This is potential Greatest Common Factor(1) between Numerator(1) Denominator(10)

Elapsed time in seconds to reduce fraction: 2.5362E-5

reduced numerator: 1

reduced denominator: 10

Fraction presented is: 2/3

No whole number before fraction

1This is the difference(0.5666666666666667) between original(0.1) decimal and associated fraction(2/3) in decimal(0.6666666666666666)

2Original fraction (2/3) is non- representation of the initial decimal conversion: 0.1(1/10)

****

This will just be final part of the code...

Since several prime numbers have been generated, I will use the existing numerator

And store the results in a String array: Numerator/prime number AS fraction and Numerator/prime number AS decimal and store results. IT IS TO GET A FEEL OF DIFFERENT PATTERNS AS EXPLORED IN DOCUMENTATION!

This will provide a sample should I wish to populate initial array also!

It will also show examples in which repetends have occurred two or more times in the decimal precision (19 digits wide)

THIS IS ONLY circumstances available to enhance my code to utilize mapping to fractional equivalent....

****

Original numerator / PRIME number AS FRACTION (not reduced) => 2/2

Original numerator / PRIME number AS DECIMAL => 1.0

Original numerator / PRIME number AS FRACTION (not reduced) => 2/3

Original numerator / PRIME number AS DECIMAL => 0.6666666666666666

Original numerator / PRIME number AS FRACTION (not reduced) => 2/5

Original numerator / PRIME number AS DECIMAL => 0.4

****

CHECK TO SEE IF ORIGINAL FRACTION IS TERMINATING:

Terminating fraction since reduced denominator(10) has prime factor (divisible by prime number of two or five)

Original decimal:1.1 is a periodic number

*** The Decimal presented: 6.1

The fraction part: 1

this is numerator: 1

this is denominator: 10

This is potential Greatest Common Factor(1) between Numerator(1) Denominator(10)

Elapsed time in seconds to reduce fraction: 2.276E-5

reduced numerator: 1

reduced denominator: 10

Fraction presented is: 10/9

No whole number before fraction

Original whole number from fraction (numerator/denominator):1

remaining numerator: 1

remaining fraction: 1/9

Fraction provided in challenge is improper: 10/9

Mixed number fraction is: 1-1/9

Mismatch in whole numbers*

Mixed number fraction has following whole number:1

Original decimal has following whole number:6

****

1Original fraction(10/9)has been changed to mixed number fraction(1-1/9)

1This is the difference(0.0111111111111111) between original(0.1) decimal and associated fraction(1/9) in decimal(0.1111111111111111)

2Original fraction (1/9) is non- representation of the initial decimal conversion: 0.1(1/10)

1Remaining fraction(1/9) is non- representation of the initial decimal conversion: 0.1

1Remaining fraction (1/9) in decimal is 0.1111111111111111 and NOT:0.1(original)

2This is the difference(0.0111111111111111) between original(0.1) decimal and associated mixed number remaining fraction(1/9)

****

This will just be final part of the code...

Since several prime numbers have been generated, I will use the existing numerator

And store the results in a String array: Numerator/prime number AS fraction and Numerator/prime number AS decimal and store results. IT IS TO GET A FEEL OF DIFFERENT PATTERNS AS EXPLORED IN DOCUMENTATION!

This will provide a sample should I wish to populate initial array also!

It will also show examples in which repetends have occurred two or more times in the decimal precision (19 digits wide)

THIS IS ONLY circumstances available to enhance my code to utilize mapping to fractional equivalent....

****

Original numerator / PRIME number AS FRACTION (not reduced) => 10/2

Original numerator / PRIME number AS DECIMAL => 5.0

Original numerator / PRIME number AS FRACTION (not reduced) => 10/3

Original numerator / PRIME number AS DECIMAL => 3.3333333333333335

Original numerator / PRIME number AS FRACTION (not reduced) => 10/5

Original numerator / PRIME number AS DECIMAL => 2.0

****

CHECK TO SEE IF ORIGINAL FRACTION IS TERMINATING:

Terminating fraction since reduced denominator(10) has prime factor (divisible by prime number of two or five)

Original decimal:6.1 is a periodic number

*** The Decimal presented: 6.1

The fraction part: 1

this is numerator: 1

this is denominator: 10

This is potential Greatest Common Factor(1) between Numerator(1) Denominator(10)

Elapsed time in seconds to reduce fraction: 1.9524E-5

reduced numerator: 1

reduced denominator: 10

Fraction presented is: 1-3/9

Whole number before fraction: 1

Fraction after whole number: 3/9

2Mismatch in whole numbers*

Mixed number fraction has following whole number:1

Original decimal has following whole number:6

****

1This is the difference(0.2333333333333333) between original(0.1) decimal and associated fraction(1-3/9) in decimal(0.3333333333333333)

2Original fraction (1-3/9) is non- representation of the initial decimal conversion: 0.1(1/10)

****

This will just be final part of the code...

Since several prime numbers have been generated, I will use the existing numerator

And store the results in a String array: Numerator/prime number AS fraction and Numerator/prime number AS decimal and store results. IT IS TO GET A FEEL OF DIFFERENT PATTERNS AS EXPLORED IN DOCUMENTATION!

This will provide a sample should I wish to populate initial array also!

It will also show examples in which repetends have occurred two or more times in the decimal precision (19 digits wide)

THIS IS ONLY circumstances available to enhance my code to utilize mapping to fractional equivalent....

****

Original numerator / PRIME number AS FRACTION (not reduced) => 3/2

Original numerator / PRIME number AS DECIMAL => 1.5

Original numerator / PRIME number AS FRACTION (not reduced) => 3/3

Original numerator / PRIME number AS DECIMAL => 1.0

Original numerator / PRIME number AS FRACTION (not reduced) => 3/5

Original numerator / PRIME number AS DECIMAL => 0.6

****

CHECK TO SEE IF ORIGINAL FRACTION IS TERMINATING:

Terminating fraction since reduced denominator(10) has prime factor (divisible by prime number of two or five)

Original decimal:6.1 is a periodic number

*** The Decimal presented: 0.10973

The fraction part: 10973

this is numerator: 10973

this is denominator: 100000

This is potential Greatest Common Factor(10973) between Numerator(10973) Denominator(100000)

Elapsed time in seconds to reduce fraction: 0.082441117

reduced numerator: 10973

reduced denominator: 100000

Fraction presented is: 5-11/10

Whole number before fraction: 5

Fraction after whole number: 11/10

Original whole number from fraction (numerator/denominator):1

remaining numerator: 1

remaining fraction: 1/10

Fraction provided in challenge is improper: 5-11/10

Mixed number fraction is: 6-1/10

Mismatch in whole numbers*

Mixed number fraction has following whole number:6

Original decimal has following whole number:0

****

1Original fraction(5-11/10)has been changed to mixed number fraction(6-1/10)

1This is the difference(0.009729999999999989) between original(0.10973) decimal and associated fraction(1/10) in decimal(0.1)

2Original fraction (1/10) is non- representation of the initial decimal conversion: 0.10973(10973/100000)

1Remaining fraction(1/10) is non- representation of the initial decimal conversion: 0.10973

1Remaining fraction (1/10) in decimal is 0.1 and NOT:0.10973(original)

2This is the difference(0.009729999999999989) between original(0.10973) decimal and associated mixed number remaining fraction(1/10)

****

This will just be final part of the code...

Since several prime numbers have been generated, I will use the existing numerator

And store the results in a String array: Numerator/prime number AS fraction and Numerator/prime number AS decimal and store results. IT IS TO GET A FEEL OF DIFFERENT PATTERNS AS EXPLORED IN DOCUMENTATION!

This will provide a sample should I wish to populate initial array also!

It will also show examples in which repetends have occurred two or more times in the decimal precision (19 digits wide)

THIS IS ONLY circumstances available to enhance my code to utilize mapping to fractional equivalent....

****

Original numerator / PRIME number AS FRACTION (not reduced) => 11/2

Original numerator / PRIME number AS DECIMAL => 5.5

Original numerator / PRIME number AS FRACTION (not reduced) => 11/3

Original numerator / PRIME number AS DECIMAL => 3.6666666666666665

Original numerator / PRIME number AS FRACTION (not reduced) => 11/5

Original numerator / PRIME number AS DECIMAL => 2.2

****

CHECK TO SEE IF ORIGINAL FRACTION IS TERMINATING:

Terminating fraction since reduced denominator(100000) has prime factor (divisible by prime number of two or five)

Original decimal:0.10973 is a periodic number

*** The Decimal presented: 0.343

The fraction part: 343

this is numerator: 343

this is denominator: 1000

This is potential Greatest Common Factor(343) between Numerator(343) Denominator(1000)

Elapsed time in seconds to reduce fraction: 1.17242E-4

reduced numerator: 343

reduced denominator: 1000

Fraction presented is: 823/7500

No whole number before fraction

1This is the difference(0.23326666666666668) between original(0.343) decimal and associated fraction(823/7500) in decimal(0.10973333333333334)

2Original fraction (823/7500) is non- representation of the initial decimal conversion: 0.343(343/1000)

****

This will just be final part of the code...

Since several prime numbers have been generated, I will use the existing numerator

And store the results in a String array: Numerator/prime number AS fraction and Numerator/prime number AS decimal and store results. IT IS TO GET A FEEL OF DIFFERENT PATTERNS AS EXPLORED IN DOCUMENTATION!

This will provide a sample should I wish to populate initial array also!

It will also show examples in which repetends have occurred two or more times in the decimal precision (19 digits wide)

THIS IS ONLY circumstances available to enhance my code to utilize mapping to fractional equivalent....

****

Original numerator / PRIME number AS FRACTION (not reduced) => 823/2

Original numerator / PRIME number AS DECIMAL => 411.5

Original numerator / PRIME number AS FRACTION (not reduced) => 823/3

Original numerator / PRIME number AS DECIMAL => 274.3333333333333

Original numerator / PRIME number AS FRACTION (not reduced) => 823/5

Original numerator / PRIME number AS DECIMAL => 164.6

****

CHECK TO SEE IF ORIGINAL FRACTION IS TERMINATING:

Terminating fraction since reduced denominator(1000) has prime factor (divisible by prime number of two or five)

Original decimal:0.343 is a periodic number

*** The Decimal presented: 0.242424

The fraction part: 242424

this is numerator: 242424

this is denominator: 1000000

This is potential Greatest Common Factor(242424) between Numerator(242424) Denominator(1000000)

Reduction numerator: (242424,121212 divisor: 2)

Reduction denominator: (1000000,500000 divisor: 2)

Reduction in fraction and decimal: 242424/1000000(0.242424) => 121212/500000(0.242424)

Reduction numerator: (121212,60606 divisor: 2)

Reduction denominator: (500000,250000 divisor: 2)

Reduction in fraction and decimal: 121212/500000(0.242424) => 60606/250000(0.242424)

Reduction numerator: (60606,30303 divisor: 2)

Reduction denominator: (250000,125000 divisor: 2)

Reduction in fraction and decimal: 60606/250000(0.242424) => 30303/125000(0.242424)

Elapsed time in seconds to reduce fraction: 0.522619279

reduced numerator: 30303

reduced denominator: 125000

Fraction presented is: 1/3

No whole number before fraction

1This is the difference(0.09090933333333331) between original(0.242424) decimal and associated fraction(1/3) in decimal(0.3333333333333333)

2Original fraction (1/3) is non- representation of the initial decimal conversion: 0.242424(30303/125000)

****

This will just be final part of the code...

Since several prime numbers have been generated, I will use the existing numerator

And store the results in a String array: Numerator/prime number AS fraction and Numerator/prime number AS decimal and store results. IT IS TO GET A FEEL OF DIFFERENT PATTERNS AS EXPLORED IN DOCUMENTATION!

This will provide a sample should I wish to populate initial array also!

It will also show examples in which repetends have occurred two or more times in the decimal precision (19 digits wide)

THIS IS ONLY circumstances available to enhance my code to utilize mapping to fractional equivalent....

****

Original numerator / PRIME number AS FRACTION (not reduced) => 1/2

Original numerator / PRIME number AS DECIMAL => 0.5

Original numerator / PRIME number AS FRACTION (not reduced) => 1/3

Original numerator / PRIME number AS DECIMAL => 0.3333333333333333

Original numerator / PRIME number AS FRACTION (not reduced) => 1/5

Original numerator / PRIME number AS DECIMAL => 0.2

****

CHECK TO SEE IF ORIGINAL FRACTION IS TERMINATING:

Terminating fraction since reduced denominator(125000) has prime factor (divisible by prime number of two or five)

Original decimal:0.242424 is a periodic number

*** The Decimal presented: 0.1875325

The fraction part: 1875325

this is numerator: 1875325

this is denominator: 10000000

This is potential Greatest Common Factor(1875325) between Numerator(1875325) Denominator(10000000)

Reduction numerator: (1875325,375065 divisor: 5)

Reduction denominator: (10000000,2000000 divisor: 5)

Reduction in fraction and decimal: 1875325/10000000(0.1875325) => 375065/2000000(0.1875325)

Reduction numerator: (375065,75013 divisor: 5)

Reduction denominator: (2000000,400000 divisor: 5)

Reduction in fraction and decimal: 375065/2000000(0.1875325) => 75013/400000(0.1875325)

Elapsed time in seconds to reduce fraction: 1.748738957

reduced numerator: 75013

reduced denominator: 400000

Fraction presented is: 30303/125000

No whole number before fraction

1This is the difference(0.05489150000000001) between original(0.1875325) decimal and associated fraction(30303/125000) in decimal(0.242424)

2Original fraction (30303/125000) is non- representation of the initial decimal conversion: 0.1875325(75013/400000)

****

This will just be final part of the code...

Since several prime numbers have been generated, I will use the existing numerator

And store the results in a String array: Numerator/prime number AS fraction and Numerator/prime number AS decimal and store results. IT IS TO GET A FEEL OF DIFFERENT PATTERNS AS EXPLORED IN DOCUMENTATION!

This will provide a sample should I wish to populate initial array also!

It will also show examples in which repetends have occurred two or more times in the decimal precision (19 digits wide)

THIS IS ONLY circumstances available to enhance my code to utilize mapping to fractional equivalent....

****

Original numerator / PRIME number AS FRACTION (not reduced) => 30303/2

Original numerator / PRIME number AS DECIMAL => 15151.5

Original numerator / PRIME number AS FRACTION (not reduced) => 30303/3

Original numerator / PRIME number AS DECIMAL => 10101.0

Original numerator / PRIME number AS FRACTION (not reduced) => 30303/5

Original numerator / PRIME number AS DECIMAL => 6060.6

****

CHECK TO SEE IF ORIGINAL FRACTION IS TERMINATING:

Terminating fraction since reduced denominator(400000) has prime factor (divisible by prime number of two or five)

Original decimal:0.1875325 is a periodic number

*** The Decimal presented: 0.5

The fraction part: 5

this is numerator: 5

this is denominator: 10

This is potential Greatest Common Factor(5) between Numerator(5) Denominator(10)

Reduction numerator: (5,1 divisor: 5)

Reduction denominator: (10,2 divisor: 5)

Reduction in fraction and decimal: 5/10(0.5) => 1/2(0.5)

Elapsed time in seconds to reduce fraction: 0.068905001

reduced numerator: 1

reduced denominator: 2

Fraction presented is: 75013/400000

No whole number before fraction

1This is the difference(0.3124675) between original(0.5) decimal and associated fraction(75013/400000) in decimal(0.1875325)

2Original fraction (75013/400000) is non- representation of the initial decimal conversion: 0.5(1/2)

****

This will just be final part of the code...

Since several prime numbers have been generated, I will use the existing numerator

And store the results in a String array: Numerator/prime number AS fraction and Numerator/prime number AS decimal and store results. IT IS TO GET A FEEL OF DIFFERENT PATTERNS AS EXPLORED IN DOCUMENTATION!

This will provide a sample should I wish to populate initial array also!

It will also show examples in which repetends have occurred two or more times in the decimal precision (19 digits wide)

THIS IS ONLY circumstances available to enhance my code to utilize mapping to fractional equivalent....

****

Original numerator / PRIME number AS FRACTION (not reduced) => 75013/2

Original numerator / PRIME number AS DECIMAL => 37506.5

Original numerator / PRIME number AS FRACTION (not reduced) => 75013/3

Original numerator / PRIME number AS DECIMAL => 25004.333333333332

Original numerator / PRIME number AS FRACTION (not reduced) => 75013/5

Original numerator / PRIME number AS DECIMAL => 15002.6

****

CHECK TO SEE IF ORIGINAL FRACTION IS TERMINATING:

Terminating fraction since reduced denominator(2) has prime factor (divisible by prime number of two or five)

Original decimal:0.5 is a periodic number

*** The Decimal presented: 1.25

The fraction part: 25

this is numerator: 25

this is denominator: 100

This is potential Greatest Common Factor(25) between Numerator(25) Denominator(100)

Reduction numerator: (25,5 divisor: 5)

Reduction denominator: (100,20 divisor: 5)

Reduction in fraction and decimal: 25/100(0.25) => 5/20(0.25)

Reduction numerator: (5,1 divisor: 5)

Reduction denominator: (20,4 divisor: 5)

Reduction in fraction and decimal: 5/20(0.25) => 1/4(0.25)

Elapsed time in seconds to reduce fraction: 1.69907E-4

reduced numerator: 1

reduced denominator: 4

Fraction presented is: 1/2

No whole number before fraction

1This is the difference(0.25) between original(0.25) decimal and associated fraction(1/2) in decimal(0.5)

2Original fraction (1/2) is non- representation of the initial decimal conversion: 0.25(1/4)

****

This will just be final part of the code...

Since several prime numbers have been generated, I will use the existing numerator

And store the results in a String array: Numerator/prime number AS fraction and Numerator/prime number AS decimal and store results. IT IS TO GET A FEEL OF DIFFERENT PATTERNS AS EXPLORED IN DOCUMENTATION!

This will provide a sample should I wish to populate initial array also!

It will also show examples in which repetends have occurred two or more times in the decimal precision (19 digits wide)

THIS IS ONLY circumstances available to enhance my code to utilize mapping to fractional equivalent....

****

Original numerator / PRIME number AS FRACTION (not reduced) => 1/2

Original numerator / PRIME number AS DECIMAL => 0.5

Original numerator / PRIME number AS FRACTION (not reduced) => 1/3

Original numerator / PRIME number AS DECIMAL => 0.3333333333333333

Original numerator / PRIME number AS FRACTION (not reduced) => 1/5

Original numerator / PRIME number AS DECIMAL => 0.2

****

CHECK TO SEE IF ORIGINAL FRACTION IS TERMINATING:

Terminating fraction since reduced denominator(4) has prime factor (divisible by prime number of two or five)

Original decimal:1.25 is a periodic number

*** The Decimal presented: 0.0123

The fraction part: 0123

this is numerator: 123

this is denominator: 10000

This is potential Greatest Common Factor(123) between Numerator(123) Denominator(10000)

Elapsed time in seconds to reduce fraction: 1.9818E-5

reduced numerator: 123

reduced denominator: 10000

Fraction presented is: 1-1/4

Whole number before fraction: 1

Fraction after whole number: 1/4

1This is the difference(0.2377) between original(0.0123) decimal and associated fraction(1-1/4) in decimal(0.25)

2Original fraction (1-1/4) is non- representation of the initial decimal conversion: 0.0123(123/10000)

****

This will just be final part of the code...

Since several prime numbers have been generated, I will use the existing numerator

And store the results in a String array: Numerator/prime number AS fraction and Numerator/prime number AS decimal and store results. IT IS TO GET A FEEL OF DIFFERENT PATTERNS AS EXPLORED IN DOCUMENTATION!

This will provide a sample should I wish to populate initial array also!

It will also show examples in which repetends have occurred two or more times in the decimal precision (19 digits wide)

THIS IS ONLY circumstances available to enhance my code to utilize mapping to fractional equivalent....

****

Original numerator / PRIME number AS FRACTION (not reduced) => 1/2

Original numerator / PRIME number AS DECIMAL => 0.5

Original numerator / PRIME number AS FRACTION (not reduced) => 1/3

Original numerator / PRIME number AS DECIMAL => 0.3333333333333333

Original numerator / PRIME number AS FRACTION (not reduced) => 1/5

Original numerator / PRIME number AS DECIMAL => 0.2

****

CHECK TO SEE IF ORIGINAL FRACTION IS TERMINATING:

Terminating fraction since reduced denominator(10000) has prime factor (divisible by prime number of two or five)

Original decimal:0.0123 is a periodic number

*** The Decimal presented: 0.0001

The fraction part: 0001

this is numerator: 1

this is denominator: 10000

This is potential Greatest Common Factor(1) between Numerator(1) Denominator(10000)

Elapsed time in seconds to reduce fraction: 2.2104E-5

reduced numerator: 1

reduced denominator: 10000

Fraction presented is: 123/10000

No whole number before fraction

1This is the difference(0.0122) between original(0.0001) decimal and associated fraction(123/10000) in decimal(0.0123)

2Original fraction (123/10000) is non- representation of the initial decimal conversion: 0.0001(1/10000)

****

This will just be final part of the code...

Since several prime numbers have been generated, I will use the existing numerator

And store the results in a String array: Numerator/prime number AS fraction and Numerator/prime number AS decimal and store results. IT IS TO GET A FEEL OF DIFFERENT PATTERNS AS EXPLORED IN DOCUMENTATION!

This will provide a sample should I wish to populate initial array also!

It will also show examples in which repetends have occurred two or more times in the decimal precision (19 digits wide)

THIS IS ONLY circumstances available to enhance my code to utilize mapping to fractional equivalent....

****

Original numerator / PRIME number AS FRACTION (not reduced) => 123/2

Original numerator / PRIME number AS DECIMAL => 61.5

Original numerator / PRIME number AS FRACTION (not reduced) => 123/3

Original numerator / PRIME number AS DECIMAL => 41.0

Original numerator / PRIME number AS FRACTION (not reduced) => 123/5

Original numerator / PRIME number AS DECIMAL => 24.6

****

CHECK TO SEE IF ORIGINAL FRACTION IS TERMINATING:

Terminating fraction since reduced denominator(10000) has prime factor (divisible by prime number of two or five)

Original decimal:0.0001 is a periodic number

*** The Decimal presented: 0.999

The fraction part: 999

this is numerator: 999

this is denominator: 1000

This is potential Greatest Common Factor(999) between Numerator(999) Denominator(1000)

Elapsed time in seconds to reduce fraction: 1.32108E-4

reduced numerator: 999

reduced denominator: 1000

Fraction presented is: 1/10000

No whole number before fraction

1This is the difference(0.9989) between original(0.999) decimal and associated fraction(1/10000) in decimal(1.0E-4)

2Original fraction (1/10000) is non- representation of the initial decimal conversion: 0.999(999/1000)

****

This will just be final part of the code...

Since several prime numbers have been generated, I will use the existing numerator

And store the results in a String array: Numerator/prime number AS fraction and Numerator/prime number AS decimal and store results. IT IS TO GET A FEEL OF DIFFERENT PATTERNS AS EXPLORED IN DOCUMENTATION!

This will provide a sample should I wish to populate initial array also!

It will also show examples in which repetends have occurred two or more times in the decimal precision (19 digits wide)

THIS IS ONLY circumstances available to enhance my code to utilize mapping to fractional equivalent....

****

Original numerator / PRIME number AS FRACTION (not reduced) => 1/2

Original numerator / PRIME number AS DECIMAL => 0.5

Original numerator / PRIME number AS FRACTION (not reduced) => 1/3

Original numerator / PRIME number AS DECIMAL => 0.3333333333333333

Original numerator / PRIME number AS FRACTION (not reduced) => 1/5

Original numerator / PRIME number AS DECIMAL => 0.2

****

CHECK TO SEE IF ORIGINAL FRACTION IS TERMINATING:

Terminating fraction since reduced denominator(1000) has prime factor (divisible by prime number of two or five)

Original decimal:0.999 is a periodic number

*** The Decimal presented: 0.15

The fraction part: 15

this is numerator: 15

this is denominator: 100

This is potential Greatest Common Factor(15) between Numerator(15) Denominator(100)

Reduction numerator: (15,3 divisor: 5)

Reduction denominator: (100,20 divisor: 5)

Reduction in fraction and decimal: 15/100(0.15) => 3/20(0.15)

Elapsed time in seconds to reduce fraction: 0.059610496

reduced numerator: 3

reduced denominator: 20

Fraction presented is: 999/1000

No whole number before fraction

1This is the difference(0.849) between original(0.15) decimal and associated fraction(999/1000) in decimal(0.999)

2Original fraction (999/1000) is non- representation of the initial decimal conversion: 0.15(3/20)

****

This will just be final part of the code...

Since several prime numbers have been generated, I will use the existing numerator

And store the results in a String array: Numerator/prime number AS fraction and Numerator/prime number AS decimal and store results. IT IS TO GET A FEEL OF DIFFERENT PATTERNS AS EXPLORED IN DOCUMENTATION!

This will provide a sample should I wish to populate initial array also!

It will also show examples in which repetends have occurred two or more times in the decimal precision (19 digits wide)

THIS IS ONLY circumstances available to enhance my code to utilize mapping to fractional equivalent....

****

Original numerator / PRIME number AS FRACTION (not reduced) => 999/2

Original numerator / PRIME number AS DECIMAL => 499.5

Original numerator / PRIME number AS FRACTION (not reduced) => 999/3

Original numerator / PRIME number AS DECIMAL => 333.0

Original numerator / PRIME number AS FRACTION (not reduced) => 999/5

Original numerator / PRIME number AS DECIMAL => 199.8

****

CHECK TO SEE IF ORIGINAL FRACTION IS TERMINATING:

Terminating fraction since reduced denominator(20) has prime factor (divisible by prime number of two or five)

Original decimal:0.15 is a periodic number

*** The Decimal presented: 0.86

The fraction part: 86

this is numerator: 86

this is denominator: 100

This is potential Greatest Common Factor(86) between Numerator(86) Denominator(100)

Reduction numerator: (86,43 divisor: 2)

Reduction denominator: (100,50 divisor: 2)

Reduction in fraction and decimal: 86/100(0.86) => 43/50(0.86)

Elapsed time in seconds to reduce fraction: 1.05129E-4

reduced numerator: 43

reduced denominator: 50

Fraction presented is: 15/100

No whole number before fraction

1This is the difference(0.71) between original(0.86) decimal and associated fraction(15/100) in decimal(0.15)

2Original fraction (15/100) is non- representation of the initial decimal conversion: 0.86(43/50)

****

This will just be final part of the code...

Since several prime numbers have been generated, I will use the existing numerator

And store the results in a String array: Numerator/prime number AS fraction and Numerator/prime number AS decimal and store results. IT IS TO GET A FEEL OF DIFFERENT PATTERNS AS EXPLORED IN DOCUMENTATION!

This will provide a sample should I wish to populate initial array also!

It will also show examples in which repetends have occurred two or more times in the decimal precision (19 digits wide)

THIS IS ONLY circumstances available to enhance my code to utilize mapping to fractional equivalent....

****

Original numerator / PRIME number AS FRACTION (not reduced) => 15/2

Original numerator / PRIME number AS DECIMAL => 7.5

Original numerator / PRIME number AS FRACTION (not reduced) => 15/3

Original numerator / PRIME number AS DECIMAL => 5.0

Original numerator / PRIME number AS FRACTION (not reduced) => 15/5

Original numerator / PRIME number AS DECIMAL => 3.0

****

CHECK TO SEE IF ORIGINAL FRACTION IS TERMINATING:

Terminating fraction since reduced denominator(50) has prime factor (divisible by prime number of two or five)

Original decimal:0.86 is a periodic number

*** The Decimal presented: 10.4

The fraction part: 4

this is numerator: 4

this is denominator: 10

This is potential Greatest Common Factor(4) between Numerator(4) Denominator(10)

Reduction numerator: (4,2 divisor: 2)

Reduction denominator: (10,5 divisor: 2)

Reduction in fraction and decimal: 4/10(0.4) => 2/5(0.4)

Elapsed time in seconds to reduce fraction: 8.1666E-5

reduced numerator: 2

reduced denominator: 5

Fraction presented is: 86/100

No whole number before fraction

1This is the difference(0.45999999999999996) between original(0.4) decimal and associated fraction(86/100) in decimal(0.86)

2Original fraction (86/100) is non- representation of the initial decimal conversion: 0.4(2/5)

****

This will just be final part of the code...

Since several prime numbers have been generated, I will use the existing numerator

And store the results in a String array: Numerator/prime number AS fraction and Numerator/prime number AS decimal and store results. IT IS TO GET A FEEL OF DIFFERENT PATTERNS AS EXPLORED IN DOCUMENTATION!

This will provide a sample should I wish to populate initial array also!

It will also show examples in which repetends have occurred two or more times in the decimal precision (19 digits wide)

THIS IS ONLY circumstances available to enhance my code to utilize mapping to fractional equivalent....

****

Original numerator / PRIME number AS FRACTION (not reduced) => 86/2

Original numerator / PRIME number AS DECIMAL => 43.0

Original numerator / PRIME number AS FRACTION (not reduced) => 86/3

Original numerator / PRIME number AS DECIMAL => 28.666666666666668

Original numerator / PRIME number AS FRACTION (not reduced) => 86/5

Original numerator / PRIME number AS DECIMAL => 17.2

****

CHECK TO SEE IF ORIGINAL FRACTION IS TERMINATING:

Terminating fraction since reduced denominator(5) has prime factor (divisible by prime number of two or five)

Original decimal:10.4 is a periodic number

*** The Decimal presented: 23.5

The fraction part: 5

this is numerator: 5

this is denominator: 10

This is potential Greatest Common Factor(5) between Numerator(5) Denominator(10)

Reduction numerator: (5,1 divisor: 5)

Reduction denominator: (10,2 divisor: 5)

Reduction in fraction and decimal: 5/10(0.5) => 1/2(0.5)

Elapsed time in seconds to reduce fraction: 8.0041E-5

reduced numerator: 1

reduced denominator: 2

Fraction presented is: 10-2/5

Whole number before fraction: 10

Fraction after whole number: 2/5

2Mismatch in whole numbers*

Mixed number fraction has following whole number:10

Original decimal has following whole number:23

****

1This is the difference(0.09999999999999998) between original(0.5) decimal and associated fraction(10-2/5) in decimal(0.4)

2Original fraction (10-2/5) is non- representation of the initial decimal conversion: 0.5(1/2)

****

This will just be final part of the code...

Since several prime numbers have been generated, I will use the existing numerator

And store the results in a String array: Numerator/prime number AS fraction and Numerator/prime number AS decimal and store results. IT IS TO GET A FEEL OF DIFFERENT PATTERNS AS EXPLORED IN DOCUMENTATION!

This will provide a sample should I wish to populate initial array also!

It will also show examples in which repetends have occurred two or more times in the decimal precision (19 digits wide)

THIS IS ONLY circumstances available to enhance my code to utilize mapping to fractional equivalent....

****

Original numerator / PRIME number AS FRACTION (not reduced) => 2/2

Original numerator / PRIME number AS DECIMAL => 1.0

Original numerator / PRIME number AS FRACTION (not reduced) => 2/3

Original numerator / PRIME number AS DECIMAL => 0.6666666666666666

Original numerator / PRIME number AS FRACTION (not reduced) => 2/5

Original numerator / PRIME number AS DECIMAL => 0.4

****

CHECK TO SEE IF ORIGINAL FRACTION IS TERMINATING:

Terminating fraction since reduced denominator(2) has prime factor (divisible by prime number of two or five)

Original decimal:23.5 is a periodic number

*** The Decimal presented: 0.75

The fraction part: 75

this is numerator: 75

this is denominator: 100

This is potential Greatest Common Factor(75) between Numerator(75) Denominator(100)

Reduction numerator: (75,15 divisor: 5)

Reduction denominator: (100,20 divisor: 5)

Reduction in fraction and decimal: 75/100(0.75) => 15/20(0.75)

Reduction numerator: (15,3 divisor: 5)

Reduction denominator: (20,4 divisor: 5)

Reduction in fraction and decimal: 15/20(0.75) => 3/4(0.75)

Elapsed time in seconds to reduce fraction: 1.69385E-4

reduced numerator: 3

reduced denominator: 4

Fraction presented is: 23-1/2

Whole number before fraction: 23

Fraction after whole number: 1/2

1This is the difference(0.25) between original(0.75) decimal and associated fraction(23-1/2) in decimal(0.5)

2Original fraction (23-1/2) is non- representation of the initial decimal conversion: 0.75(3/4)

****

This will just be final part of the code...

Since several prime numbers have been generated, I will use the existing numerator

And store the results in a String array: Numerator/prime number AS fraction and Numerator/prime number AS decimal and store results. IT IS TO GET A FEEL OF DIFFERENT PATTERNS AS EXPLORED IN DOCUMENTATION!

This will provide a sample should I wish to populate initial array also!

It will also show examples in which repetends have occurred two or more times in the decimal precision (19 digits wide)

THIS IS ONLY circumstances available to enhance my code to utilize mapping to fractional equivalent....

****

Original numerator / PRIME number AS FRACTION (not reduced) => 1/2

Original numerator / PRIME number AS DECIMAL => 0.5

Original numerator / PRIME number AS FRACTION (not reduced) => 1/3

Original numerator / PRIME number AS DECIMAL => 0.3333333333333333

Original numerator / PRIME number AS FRACTION (not reduced) => 1/5

Original numerator / PRIME number AS DECIMAL => 0.2

****

CHECK TO SEE IF ORIGINAL FRACTION IS TERMINATING:

Terminating fraction since reduced denominator(4) has prime factor (divisible by prime number of two or five)

Original decimal:0.75 is a periodic number

*** The Decimal presented: 0.1111111111111111111

The fraction part: 1111111111111111111

this is numerator: 1111111111111111111

this is denominator: 10000000000000000000

Denominator(10000000000000000000) considered too large for computation

Denominator reduced to 10 digits wide: (1000000000) due to recurrence numerator

Truncated denominator: 1000000000

Numerator (1111111111111111111) considered too large for computation

Numerator truncated to 9 digits wide: (111111111) due to recurrence numerator

Truncated numerator: 111111111

BACK TO HERE

2THIS IS THE REGEX TO CHECK FOR POORLY DIVISIBLE NUMERATORS :111111111

THIS IS THE numerator:111111111

2Single digit recurring:true

1Numerator truncated to: 8 digits (11111111) to mitigate resource limitations of recurrence 9 times (1,3,7,9)

2Denominator truncated to: 9 digits(100000000) inline with numerator

Truncated numerator: 11111111

Truncated numerator: 100000000

This is potential Greatest Common Factor(11111111) between Numerator(11111111) Denominator(100000000)

11Numerator truncated to: 7 digits to mitigate idle hanging of numerator: (1111111)

12Denominator truncated to: 8 digits(10000000) inline with numerator

Truncated numerator: 1111111

Truncated denominator: 10000000

Elapsed time in seconds to reduce fraction: 1.995957231

reduced numerator: 1111111

reduced denominator: 10000000

Fraction presented is: 3/4

No whole number before fraction

1This is the difference(0.6388888888888888) between original(0.1111111111111111111) decimal and associated fraction(3/4) in decimal(0.75)

2Original fraction (3/4) is non- representation of the initial decimal conversion: 0.11111111111111111111(1111111/10000000)

****

This will just be final part of the code...

Since several prime numbers have been generated, I will use the existing numerator

And store the results in a String array: Numerator/prime number AS fraction and Numerator/prime number AS decimal and store results. IT IS TO GET A FEEL OF DIFFERENT PATTERNS AS EXPLORED IN DOCUMENTATION!

This will provide a sample should I wish to populate initial array also!

It will also show examples in which repetends have occurred two or more times in the decimal precision (19 digits wide)

THIS IS ONLY circumstances available to enhance my code to utilize mapping to fractional equivalent....

****

Original numerator / PRIME number AS FRACTION (not reduced) => 3/2

Original numerator / PRIME number AS DECIMAL => 1.5

Original numerator / PRIME number AS FRACTION (not reduced) => 3/3

Original numerator / PRIME number AS DECIMAL => 1.0

Original numerator / PRIME number AS FRACTION (not reduced) => 3/5

Original numerator / PRIME number AS DECIMAL => 0.6

****

CHECK TO SEE IF ORIGINAL FRACTION IS TERMINATING:

Terminating fraction since reduced denominator(10000000) has prime factor (divisible by prime number of two or five)

Original decimal:0.1111111111111111111 is a periodic number

*** The Decimal presented: 0.1111111111111111111

The fraction part: 1111111111111111111

this is numerator: 1111111111111111111

this is denominator: 10000000000000000000

Denominator(10000000000000000000) considered too large for computation

Denominator reduced to 10 digits wide: (1000000000) due to recurrence numerator

Truncated denominator: 1000000000

Numerator (1111111111111111111) considered too large for computation

Numerator truncated to 9 digits wide: (111111111) due to recurrence numerator

Truncated numerator: 111111111

BACK TO HERE

2THIS IS THE REGEX TO CHECK FOR POORLY DIVISIBLE NUMERATORS :111111111

THIS IS THE numerator:111111111

2Single digit recurring:true

1Numerator truncated to: 8 digits (11111111) to mitigate resource limitations of recurrence 9 times (1,3,7,9)

2Denominator truncated to: 9 digits(100000000) inline with numerator

Truncated numerator: 11111111

Truncated numerator: 100000000

This is potential Greatest Common Factor(11111111) between Numerator(11111111) Denominator(100000000)

11Numerator truncated to: 7 digits to mitigate idle hanging of numerator: (1111111)

12Denominator truncated to: 8 digits(10000000) inline with numerator

Truncated numerator: 1111111

Truncated denominator: 10000000

Elapsed time in seconds to reduce fraction: 1.912321787

reduced numerator: 1111111

reduced denominator: 10000000

Fraction presented is: 3/27

No whole number before fraction

1This is the difference(0.0) between original(0.1111111111111111111) decimal and associated fraction(3/27) in decimal(0.1111111111111111)

2Original fraction (3/27) is exact representation of the initial decimal conversion: 0.111111111111111111(1111111/10000000)

****

This will just be final part of the code...

Since several prime numbers have been generated, I will use the existing numerator

And store the results in a String array: Numerator/prime number AS fraction and Numerator/prime number AS decimal and store results. IT IS TO GET A FEEL OF DIFFERENT PATTERNS AS EXPLORED IN DOCUMENTATION!

This will provide a sample should I wish to populate initial array also!

It will also show examples in which repetends have occurred two or more times in the decimal precision (19 digits wide)

THIS IS ONLY circumstances available to enhance my code to utilize mapping to fractional equivalent....

****

Original numerator / PRIME number AS FRACTION (not reduced) => 3/2

Original numerator / PRIME number AS DECIMAL => 1.5

Original numerator / PRIME number AS FRACTION (not reduced) => 3/3

Original numerator / PRIME number AS DECIMAL => 1.0

Original numerator / PRIME number AS FRACTION (not reduced) => 3/5

Original numerator / PRIME number AS DECIMAL => 0.6

****

CHECK TO SEE IF ORIGINAL FRACTION IS TERMINATING:

Terminating fraction since reduced denominator(10000000) has prime factor (divisible by prime number of two or five)

Original decimal:0.1111111111111111111 is a periodic number

*** The Decimal presented: 0.33533

The fraction part: 33533

this is numerator: 33533

this is denominator: 100000

This is potential Greatest Common Factor(33533) between Numerator(33533) Denominator(100000)

Elapsed time in seconds to reduce fraction: 9.44989E-4

reduced numerator: 33533

reduced denominator: 100000

Fraction presented is: 1/9

No whole number before fraction

1This is the difference(0.2242188888888889) between original(0.33533) decimal and associated fraction(1/9) in decimal(0.1111111111111111)

2Original fraction (1/9) is non- representation of the initial decimal conversion: 0.33533(33533/100000)

****

This will just be final part of the code...

Since several prime numbers have been generated, I will use the existing numerator

And store the results in a String array: Numerator/prime number AS fraction and Numerator/prime number AS decimal and store results. IT IS TO GET A FEEL OF DIFFERENT PATTERNS AS EXPLORED IN DOCUMENTATION!

This will provide a sample should I wish to populate initial array also!

It will also show examples in which repetends have occurred two or more times in the decimal precision (19 digits wide)

THIS IS ONLY circumstances available to enhance my code to utilize mapping to fractional equivalent....

****

Original numerator / PRIME number AS FRACTION (not reduced) => 1/2

Original numerator / PRIME number AS DECIMAL => 0.5

Original numerator / PRIME number AS FRACTION (not reduced) => 1/3

Original numerator / PRIME number AS DECIMAL => 0.3333333333333333

Original numerator / PRIME number AS FRACTION (not reduced) => 1/5

Original numerator / PRIME number AS DECIMAL => 0.2

****

CHECK TO SEE IF ORIGINAL FRACTION IS TERMINATING:

Terminating fraction since reduced denominator(100000) has prime factor (divisible by prime number of two or five)

Original decimal:0.33533 is a periodic number

*** The Decimal presented: 5.1111111111111111111

The fraction part: 111111111111111111

this is numerator: 111111111111111111

this is denominator: 10000000000000000000

Denominator(100000000000000000) considered too large for computation

Denominator reduced to 10 digits wide: (1000000000) due to recurrence numerator

Truncated denominator: 1000000000

Numerator (1111111111111111111) considered too large for computation

Numerator truncated to 9 digits wide: (111111111) due to recurrence numerator

Truncated numerator: 111111111

BACK TO HERE

2THIS IS THE REGEX TO CHECK FOR POORLY DIVISIBLE NUMERATORS :111111111

THIS IS THE numerator:111111111

2Single digit recurring:true

1Numerator truncated to: 8 digits (11111111) to mitigate resource limitations of recurrence 9 times (1,3,7,9)

2Denominator truncated to: 9 digits(100000000) inline with numerator

Truncated numerator: 11111111

Truncated numerator: 100000000

This is potential Greatest Common Factor(11111111) between Numerator(11111111) Denominator(100000000)

11Numerator truncated to: 7 digits to mitigate idle hanging of numerator: (1111111)

12Denominator truncated to: 8 digits(10000000) inline with numerator

Truncated numerator: 1111111

Truncated denominator: 10000000

Elapsed time in seconds to reduce fraction: 1.607480814

reduced numerator: 1111111

reduced denominator: 10000000

Fraction presented is: 33533/100000

No whole number before fraction

1This is the difference(0.2242188888888889) between original(0.1111111111111111111) decimal and associated fraction(33533/100000) in decimal(0.33533)

2Original fraction (33533/100000) is non- representation of the initial decimal conversion: 0.111111111111111111(1111111/10000000)

****

This will just be final part of the code...

Since several prime numbers have been generated, I will use the existing numerator

And store the results in a String array: Numerator/prime number AS fraction and Numerator/prime number AS decimal and store results. IT IS TO GET A FEEL OF DIFFERENT PATTERNS AS EXPLORED IN DOCUMENTATION!

This will provide a sample should I wish to populate initial array also!

It will also show examples in which repetends have occurred two or more times in the decimal precision (19 digits wide)

THIS IS ONLY circumstances available to enhance my code to utilize mapping to fractional equivalent....

****

Original numerator / PRIME number AS FRACTION (not reduced) => 33533/2

Original numerator / PRIME number AS DECIMAL => 16766.5

Original numerator / PRIME number AS FRACTION (not reduced) => 33533/3

Original numerator / PRIME number AS DECIMAL => 11177.666666666666

Original numerator / PRIME number AS FRACTION (not reduced) => 33533/5

Original numerator / PRIME number AS DECIMAL => 6706.6

****

CHECK TO SEE IF ORIGINAL FRACTION IS TERMINATING:

Terminating fraction since reduced denominator(10000000) has prime factor (divisible by prime number of two or five)

Original decimal:5.1111111111111111111 is a periodic number

*** The Decimal presented: 5.111111111111111111

The fraction part: 111111111111111111

this is numerator: 111111111111111111

this is denominator: 10000000000000000000

Denominator(10000000000000000000) considered too large for computation

Denominator reduced to 10 digits wide: (1000000000) due to recurrence numerator

Truncated denominator: 1000000000

Numerator (111111111111111111) considered too large for computation

Numerator truncated to 9 digits wide: (111111111) due to recurrence numerator

Truncated numerator: 111111111

BACK TO HERE

2THIS IS THE REGEX TO CHECK FOR POORLY DIVISIBLE NUMERATORS :111111111

THIS IS THE numerator:111111111

2Single digit recurring:true

1Numerator truncated to: 8 digits (11111111) to mitigate resource limitations of recurrence 9 times (1,3,7,9)

2Denominator truncated to: 9 digits(100000000) inline with numerator

Truncated numerator: 11111111

Truncated numerator: 100000000

This is potential Greatest Common Factor(11111111) between Numerator(11111111) Denominator(100000000)

11Numerator truncated to: 7 digits to mitigate idle hanging of numerator: (1111111)

12Denominator truncated to: 8 digits(10000000) inline with numerator

Truncated numerator: 1111111

Truncated denominator: 10000000

Elapsed time in seconds to reduce fraction: 1.897284791

reduced numerator: 1111111

reduced denominator: 10000000

Fraction presented is: 4-64/9

Whole number before fraction: 4

Fraction after whole number: 64/9

Original whole number from fraction (numerator/denominator):7

remaining numerator: 1

remaining fraction: 1/9

Fraction provided in challenge is improper: 4-64/9

Mixed number fraction is: 11-1/9

Mismatch in whole numbers*

Mixed number fraction has following whole number:11

Original decimal has following whole number:5

****

1Original fraction(4-64/9)has been changed to mixed number fraction(11-1/9)

1This is the difference(0.0) between original(0.1111111111111111111) decimal and associated fraction(1/9) in decimal(0.1111111111111111)

2Original fraction (1/9) is exact representation of the initial decimal conversion: 0.11111111111111111111(1111111/10000000)

****

This will just be final part of the code...

Since several prime numbers have been generated, I will use the existing numerator

And store the results in a String array: Numerator/prime number AS fraction and Numerator/prime number AS decimal and store results. IT IS TO GET A FEEL OF DIFFERENT PATTERNS AS EXPLORED IN DOCUMENTATION!

This will provide a sample should I wish to populate initial array also!

It will also show examples in which repetends have occurred two or more times in the decimal precision (19 digits wide)

THIS IS ONLY circumstances available to enhance my code to utilize mapping to fractional equivalent....

****

Original numerator / PRIME number AS FRACTION (not reduced) => 64/2

Original numerator / PRIME number AS DECIMAL => 32.0

Original numerator / PRIME number AS FRACTION (not reduced) => 64/3

Original numerator / PRIME number AS DECIMAL => 21.333333333333332

Original numerator / PRIME number AS FRACTION (not reduced) => 64/5

Original numerator / PRIME number AS DECIMAL => 12.8

****

CHECK TO SEE IF ORIGINAL FRACTION IS TERMINATING:

Terminating fraction since reduced denominator(10000000) has prime factor (divisible by prime number of two or five)

Original decimal:5.1111111111111111111 is a periodic number

*** The Decimal presented: 0.555

The fraction part: 555

this is numerator: 555

this is denominator: 1000

This is potential Greatest Common Factor(555) between Numerator(555) Denominator(1000)

Reduction numerator: (555,111 divisor: 5)

Reduction denominator: (1000,200 divisor: 5)

Reduction in fraction and decimal: 555/1000(0.555) => 111/200(0.555)

Elapsed time in seconds to reduce fraction: 2.00263E-4

reduced numerator: 111

reduced denominator: 200

Fraction presented is: 64/9

No whole number before fraction

Original whole number from fraction (numerator/denominator):7

remaining numerator: 1

remaining fraction: 1/9

Fraction provided in challenge is improper: 64/9

Mixed number fraction is: 7-1/9

Mismatch in whole numbers*

Mixed number fraction has following whole number:7

Original decimal has following whole number:0

****

1Original fraction(64/9)has been changed to mixed number fraction(7-1/9)

1This is the difference(0.44388888888888894) between original(0.555) decimal and associated fraction(1/9) in decimal(0.1111111111111111)

2Original fraction (1/9) is non- representation of the initial decimal conversion: 0.555(111/200)

****

This will just be final part of the code...

Since several prime numbers have been generated, I will use the existing numerator

And store the results in a String array: Numerator/prime number AS fraction and Numerator/prime number AS decimal and store results. IT IS TO GET A FEEL OF DIFFERENT PATTERNS AS EXPLORED IN DOCUMENTATION!

This will provide a sample should I wish to populate initial array also!

It will also show examples in which repetends have occurred two or more times in the decimal precision (19 digits wide)

THIS IS ONLY circumstances available to enhance my code to utilize mapping to fractional equivalent....

****

Original numerator / PRIME number AS FRACTION (not reduced) => 64/2

Original numerator / PRIME number AS DECIMAL => 32.0

Original numerator / PRIME number AS FRACTION (not reduced) => 64/3

Original numerator / PRIME number AS DECIMAL => 21.333333333333332

Original numerator / PRIME number AS FRACTION (not reduced) => 64/5

Original numerator / PRIME number AS DECIMAL => 12.8

****

CHECK TO SEE IF ORIGINAL FRACTION IS TERMINATING:

Terminating fraction since reduced denominator(200) has prime factor (divisible by prime number of two or five)

Original decimal:0.555 is a periodic number

*** The Decimal presented: 10.0

The fraction part: 0

this is numerator: 0

this is denominator: 10

This is potential Greatest Common Factor(0) between Numerator(0) Denominator(10)

Elapsed time in seconds to reduce fraction: 1.9181E-5

reduced numerator: 0

reduced denominator: 10

Fraction presented is: 64/9

No whole number before fraction

Original whole number from fraction (numerator/denominator):7

remaining numerator: 1

remaining fraction: 1/9

Fraction provided in challenge is improper: 64/9

Mixed number fraction is: 7-1/9

Mismatch in whole numbers*

Mixed number fraction has following whole number:7

Original decimal has following whole number:10

****

1Original fraction(64/9)has been changed to mixed number fraction(7-1/9)

1This is the difference(0.1111111111111111) between original(0.0) decimal and associated fraction(1/9) in decimal(0.1111111111111111)

2Original fraction (1/9) is non- representation of the initial decimal conversion: 0.0(0/10)

1Remaining fraction(1/9) is non- representation of the initial decimal conversion: 0.0

1Remaining fraction (1/9) in decimal is 0.1111111111111111 and NOT:0.0(original)

2This is the difference(0.1111111111111111) between original(0.0) decimal and associated mixed number remaining fraction(1/9)

****

This will just be final part of the code...

Since several prime numbers have been generated, I will use the existing numerator

And store the results in a String array: Numerator/prime number AS fraction and Numerator/prime number AS decimal and store results. IT IS TO GET A FEEL OF DIFFERENT PATTERNS AS EXPLORED IN DOCUMENTATION!

This will provide a sample should I wish to populate initial array also!

It will also show examples in which repetends have occurred two or more times in the decimal precision (19 digits wide)

THIS IS ONLY circumstances available to enhance my code to utilize mapping to fractional equivalent....

****

Original numerator / PRIME number AS FRACTION (not reduced) => 64/2

Original numerator / PRIME number AS DECIMAL => 32.0

Original numerator / PRIME number AS FRACTION (not reduced) => 64/3

Original numerator / PRIME number AS DECIMAL => 21.333333333333332

Original numerator / PRIME number AS FRACTION (not reduced) => 64/5

Original numerator / PRIME number AS DECIMAL => 12.8

****

CHECK TO SEE IF ORIGINAL FRACTION IS TERMINATING:

Terminating fraction since reduced denominator(10) has prime factor (divisible by prime number of two or five)

Original decimal:10.0 is a periodic number

*** The Decimal presented: 0.011005502751375

The fraction part: 011005502751375

this is numerator: 11005502751375

this is denominator: 1000000000000000

6Numerator truncated from: 14 to 9 digits:

7Denominator truncated from 16 to 11 digits

Truncated denominator: 10000000000

Truncated numerator: 110055027

2Single digit recurring:false

2THIS IS THE REGEX TO CHECK FOR POORLY DIVISIBLE NUMERATORS :111111111

THIS IS THE numerator:110055027

This is potential Greatest Common Factor(110055027) between Numerator(110055027) Denominator(10000000000)

Elapsed time in seconds to reduce fraction: 26.179539804

reduced numerator: 110055027

reduced denominator: 10000000000

Fraction presented is: 20/2

No whole number before fraction

Original whole number from fraction (numerator/denominator):10

remaining numerator: 0

remaining fraction: 0/2

Fraction provided in challenge is improper: 20/2

Mixed number fraction is: 10-0/2

Mismatch in whole numbers*

Mixed number fraction has following whole number:10

Original decimal has following whole number:0

****

1Original fraction(20/2)has been changed to mixed number fraction(10-0/2)

1This is the difference(9.988994497248624) between original(0.011005502751375) decimal and associated fraction(0/2) in decimal(10.0)

2Original fraction (0/2) is non- representation of the initial decimal conversion: 0.011005502751375(110055027/10000000000)

Original fraction(0/2) is NOT reduced identically to the converted decimal (most reduced form): (110055027/10000000000)

1Remaining fraction(0/2) is non- representation of the initial decimal conversion: 0.011005502751375

1Remaining fraction (0/2) in decimal is 10.0 and NOT:0.011005502751375(original)

2This is the difference(0.011005502751375) between original(0.011005502751375) decimal and associated mixed number remaining fraction(0/2)

****

This will just be final part of the code...

Since several prime numbers have been generated, I will use the existing numerator

And store the results in a String array: Numerator/prime number AS fraction and Numerator/prime number AS decimal and store results. IT IS TO GET A FEEL OF DIFFERENT PATTERNS AS EXPLORED IN DOCUMENTATION!

This will provide a sample should I wish to populate initial array also!

It will also show examples in which repetends have occurred two or more times in the decimal precision (19 digits wide)

THIS IS ONLY circumstances available to enhance my code to utilize mapping to fractional equivalent....

****

Original numerator / PRIME number AS FRACTION (not reduced) => 20/2

Original numerator / PRIME number AS DECIMAL => 10.0

Original numerator / PRIME number AS FRACTION (not reduced) => 20/3

Original numerator / PRIME number AS DECIMAL => 6.666666666666667

Original numerator / PRIME number AS FRACTION (not reduced) => 20/5

Original numerator / PRIME number AS DECIMAL => 4.0

****

CHECK TO SEE IF ORIGINAL FRACTION IS TERMINATING:

Terminating fraction since reduced denominator(10000000000) has prime factor (divisible by prime number of two or five)

Original decimal:0.011005502751375 is a periodic number

*** The Decimal presented: 7.111111111111111111

The fraction part: 111111111111111111

this is numerator: 111111111111111111

this is denominator: 10000000000000000000

Denominator(10000000000000000000) considered too large for computation

Denominator reduced to 10 digits wide: (1000000000) due to recurrence numerator

Truncated denominator: 1000000000

Numerator (1111111111111111111) considered too large for computation

Numerator truncated to 9 digits wide: (111111111) due to recurrence numerator

Truncated numerator: 111111111

BACK TO HERE

2THIS IS THE REGEX TO CHECK FOR POORLY DIVISIBLE NUMERATORS :111111111

THIS IS THE numerator:111111111

2Single digit recurring:true

1Numerator truncated to: 8 digits (11111111) to mitigate resource limitations of recurrence 9 times (1,3,7,9)

2Denominator truncated to: 9 digits(100000000) inline with numerator

Truncated numerator: 11111111

Truncated numerator: 100000000

This is potential Greatest Common Factor(11111111) between Numerator(11111111) Denominator(100000000)

11Numerator truncated to: 7 digits to mitigate idle hanging of numerator: (1111111)

12Denominator truncated to: 8 digits(10000000) inline with numerator

Truncated numerator: 1111111

Truncated denominator: 10000000

Elapsed time in seconds to reduce fraction: 12.401305246

reduced numerator: 1111111

reduced denominator: 10000000

Fraction presented is: 22/1999

No whole number before fraction

1This is the difference(0.10010560835973542) between original(0.1111111111111111111) decimal and associated fraction(22/1999) in decimal(0.011005502751375688)

2Original fraction (22/1999) is non- representation of the initial decimal conversion: 0.111111111111111111(1111111/10000000)

****

This will just be final part of the code...

Since several prime numbers have been generated, I will use the existing numerator

And store the results in a String array: Numerator/prime number AS fraction and Numerator/prime number AS decimal and store results. IT IS TO GET A FEEL OF DIFFERENT PATTERNS AS EXPLORED IN DOCUMENTATION!

This will provide a sample should I wish to populate initial array also!

It will also show examples in which repetends have occurred two or more times in the decimal precision (19 digits wide)

THIS IS ONLY circumstances available to enhance my code to utilize mapping to fractional equivalent....

****

Original numerator / PRIME number AS FRACTION (not reduced) => 22/2

Original numerator / PRIME number AS DECIMAL => 11.0

Original numerator / PRIME number AS FRACTION (not reduced) => 22/3

Original numerator / PRIME number AS DECIMAL => 7.333333333333333

Original numerator / PRIME number AS FRACTION (not reduced) => 22/5

Original numerator / PRIME number AS DECIMAL => 4.4

****

CHECK TO SEE IF ORIGINAL FRACTION IS TERMINATING:

Terminating fraction since reduced denominator(10000000) has prime factor (divisible by prime number of two or five)

Original decimal:7.1111111111111111111 is a periodic number

** Process exited - Return Code: 0 **