

Package analyze

```
import "analyze"
```

[Overview](#)[Index](#)[Subdirectories](#)

Overview ▼

数値解析パッケージ

Index ▼

```
func GetGreatestCommonDivisor(x, y int)
func GetLeastCommonMultiple(x, y int)
func IsAmicableNumbers(x, y int)
func IsBetrothedNumber(x, y int)
func IsCousinPrimes(x, y int)
func IsEvenNumber(number int)
func IsFibonacciNumber(number int)
func IsFriedmanNumber(x, y int)
func IsGrothendieckPrime(number int)
func IsKaprekarNumber(number int)
func IsMersenneNumber(number int)
func IsOddNumber(number int)
func IsPerfectNumber(number int)
func IsPrimeNumber(number int)
func IsPythagoreanTriple(x, y, z int)
func IsRepunitNumber(number int)
func IsSafePrime(number int)
func IsSexyPrimes(x, y int)
func IsSophieGermainPrime(number int)
func IsSquareNumber(number int)
func IsTaxicabNumber(number int)
func IsTwinPrime(x, y int)
```

Package files

analyze.go

func GetGreatestCommonDivisor

```
func GetGreatestCommonDivisor(x, y int)
```

最大公約数判定関数。与えられた引数が最大公約数の場合、その趣旨を表示する 最大公約数とは「2つ以上の正の整数に共通な約数（公約数）のうち最大のもの」 ex) 12と9の最大公約数は3, ...

func GetLeastCommonMultiple

```
func GetLeastCommonMultiple(x, y int)
```

最小公倍数判定関数。与えられた引数が最小公倍数の場合、その趣旨を表示する 最小公倍数とは「0ではない複数の整数の公倍数のうち共通する最小の自然数を指す」 ex) 12と9の最小公倍数は36, ...

func IsAmicableNumbers

```
func IsAmicableNumbers(x, y int)
```

親和数判定関数。与えられた引数が親和数の場合、その趣旨を表示する 親和数とは「自身を除く約数の和が、他方に等しい自然数の組」 ex) 220と284, 1184と1210, 2620と2924,...

func IsBetrothedNumber

```
func IsBetrothedNumber(x, y int)
```

婚約数判定関数。与えられた引数が婚約数の場合、その趣旨を表示する 婚約数とは「1と自身を除いた約数の和が互いに他方に等しい自然数の組」 ex) 48と75, 140と195, 1050と1925, ...

func IsCousinPrimes

```
func IsCousinPrimes(x, y int)
```

いとこ素数判定関数。与えられた引数がいとこ素数の場合、その趣旨を表示する いとこ素数とは「差が4である素数の組」 ex) 3と7, 7と11, 13と17, ...

func IsEvenNumber

```
func IsEvenNumber(number int)
```

偶数判定関数。与えられた引数が偶数の場合、その趣旨を表示する 偶数とは、「2で割り切れる整数」 ex) 2, 4, 6, 8, ...

func IsFibonacciNumber

```
func IsFibonacciNumber(number int)
```

フィボナッチ数判定関数。与えられた引数がフィボナッチ数の場合、その趣旨を表示する フィボナッチ数とは「フィボナッチ数列の項に該当する数」 ex) 0, 1, 1, 2, 3, 5, 8, 13, ...

func IsFriedmanNumber

```
func IsFriedmanNumber(x, y int)
```

フリードマン数判定関数。与えられた引数がフリードマン数の場合、その趣旨を表示する フリードマン数とは「自然数のうち、その数に使われている数字を全て用いて、(I) 四則演算、(II) 累乗、(III) 複数個の数字を合わせて2桁以上の数にする、という3つの方法のうち少なくとも1つを用いて数式を作ることによって元の数に一致させられる数のこと。ただし(III)の方法だけでフリードマン数を作ることはできないものとする。」 ex) $25 = 5^2$, $121 = 11^2$, $125 = 5^{(1+2)}$

func IsGrothendieckPrime

```
func IsGrothendieckPrime(number int)
```

グロタンディーク素数判定関数。与えられた引数がグロタンディーク素数の場合、その趣旨を表示する グロタンディーク素数とは「57の数のこと」

func IsKaprekarNumber

```
func IsKaprekarNumber(number int)
```

カプレカ数一判定関数。与えられた引数がカプレカ数一の場合、その趣旨を表示する カプレカ数一とは「桁を並べ替えて最大にした数と最小にした数との差を取ったとき、元の値に等しくなる自然数」 ex) 1, 9, 45, 55, 99, 297, ...

func IsMersenneNumber

```
func IsMersenneNumber(number int)
```

メルセンヌ数判定関数。与えられた引数がメルセンス数の場合、その趣旨を表示する メルセンヌ数とは「 $2^n - 1$ で表される数」 ex) 1, 3, 7, 31, ...

func IsOddNumber

```
func IsOddNumber(number int)
```

奇数判定関数。与えられた引数が奇数の場合、その趣旨を表示する 奇数とは、「2で割り切れない整数」 ex) 1, 3, 5, 7, ...

func IsPerfectNumber

```
func IsPerfectNumber(number int)
```

完全数判定関数。与えられた引数が完全数の場合、その趣旨を表示する 完全数とは「自身を除く約数の和が自身に等しい数の自然数」 ex) 6, 28, 496, ...

func IsPrimeNumber

```
func IsPrimeNumber(number int)
```

素数判定関数。与えられた引数が素数の場合、その趣旨を表示する 素数とは「1を除く約数が1とその数自身だけである自然数」 ex) 2, 3, 5, 7, ...

func IsPythagoreanTriple

```
func IsPythagoreanTriple(x, y, z int)
```

ピタゴラス数判定関数。与えられた引数がピタゴラス数の場合、その趣旨を表示する ピタゴラス数とは「 $a^2 + b^2 = c^2$ を満たす3つの自然数の組 (a, b, c) のこと」 ex) 3と4と5, 5と12と13, ...

func IsRepunitNumber

```
func IsRepunitNumber(number int)
```

レピュニット数判定関数。与えられた引数がレピュニット数の場合、その趣旨を表示する レピュニット数とは「全ての桁の数字が1である自然数のこと」 ex) 1, 11, 111, ...

func IsSafePrime

```
func IsSafePrime(number int)
```

安全素数判定関数。与えられた引数が安全素数の場合、その趣旨を表示する 安全素数とは「 p と $2p+1$ が素数である時の $2p+1$ のこと」 ex) 5, 11, 23, 83, ...

func IsSexyPrimes

```
func IsSexyPrimes(x, y int)
```

セクシー素数判定関数。与えられた引数がセクシー素数の場合、その趣旨を表示する セクシー素数とは「差が6の素数の組($p, p + 6$)」 ex) 5と11, 7と13, 11と17, ...

func IsSophieGermainPrime

```
func IsSophieGermainPrime(number int)
```

ソフィー・ジェルマン素数判定関数。与えられた引数がソフィー・ジェルマン素数の場合、その趣旨を表示する ソフィー・ジェルマン素数とは「 p と $2p+1$ が共に素数である時の p のこと」 ex) 2, 3, 5, 11, ...

func IsSquareNumber

```
func IsSquareNumber(number int)
```

平方数。与えられた引数が平方数の場合、その趣旨を表示する 平方数とは「整数の自乗（二乗）で表される数である」 ex) 0, 1, 4, 9, 16, 25, ...

func IsTaxicabNumber

```
func IsTaxicabNumber(number int)
```

タクシー数判定関数。与えられた引数がタクシー数の場合、その趣旨を表示する タクシー数とは「2つの立方数の和として表すことができる数($x^3 + y^3 = N$)」 ex) 2 ($= 1^3 + 1^3$), 1792 ($= 1^3 + 12^3$, $= 9^3 + 10^3$), ...

func IsTwinPrime

```
func IsTwinPrime(x, y int)
```

双子素数判定関数。与えられた引数が双子素数の場合、その趣旨を表示する 双子素数とは「差が2である二つの素数の組」 ex) 3と5, 5と7, 11と13, ...

Subdirectories

Name

..

[cmd](#)

Build version go1.21.0.

Except as [noted](#), the content of this page is licensed under the Creative Commons Attribution 3.0 License, and code is licensed under a [BSD license](#).

[Terms of Service](#) | [Privacy Policy](#)