# ユークリッド互除法 Euclidean Algorithm

```
/* ユークリッド互除法 */
/* 入力した2つの自然数の最大公約数 G.C.D.を求める */
#include <stdio.h>
#include <math.h>
int main(void)
{
  int a, b, r, tmp;
  printf("2つの自然数のGCDをもとめる (ユークリッド互除法)\n");
  scanf("%d", &a);
  printf("\n自然数1 = %d", a);
  scanf("%d", &b);
  printf("\n自然数2 = %d", b);
  /* 自然数 a > b を確認・入替 */
  if(a<b){
    tmp = a;
    a = b;
    b = tmp;
  }
  /* ユークリッドの互除法 */
  r = a % b;
  while(r!=0){
    a = b;
    b = r;
    r = a % b;
  }
  /* 最大公約数を出力 */
  printf("\n最大公約数 = %d", b);
  return 0;
}
```

計算結果:

2つの自然数のGCDをもとめる (ユークリッド互除法)

自然数1 = 63

自然数2 = 21

最大公約数 = 21

```
/* ユークリッド互除法. */
/* 2つの自然数の最小公倍数 L.C.M. を求める */
```

```c
#include <stdio.h>
#include <math.h>

int main(void)
{
  int a, b, r, x, tmp;
  printf("2つの自然数のLCMをもとめる (ユークリッド互除法)\n");
  scanf("%d", &a);
  printf("\n自然数1 = %d", a);
  scanf("%d", &b);
  printf("\n自然数2 = %d", b);
  x = a * b;

  /* 自然数 a > b を確認・入替 */
  if(a<b){
    tmp = a;
    a = b;
    b = tmp;
  }

  /* ユークリッドの互除法 */
  r = a % b;
  while(r!=0){
    a = b;
    b = r;
    r = a % b;
  }

  /* 最小公倍数を出力 */
  printf("\n最小公倍数 = %d", x/b);
  return 0;
}
```

```
2つの自然数のLCMをもとめる (ユークリッド互除法)

自然数1 = 19
自然数2 = 7
最小公倍数 = 133
```

## ユークリッドの互除法：

自然数 $a, b$ ($a \geq b$) に対して，$a$ を $b$ で割った余りを $r$ とおくと、
$\gcd(a, b) = \gcd(b, r)$が成立し、
これを繰り返し用いると $a$ と $b$ の最大公約数 GCDが求まる.

証明;

$a$ を $b$ で割った商を $p$、余りを $r$ とおく.
$a = b \times p + r$ より、
$a$、$b$がともに $m$の倍数→ $r = a - b \times p$ も $m$の倍数.
$\gcd(a, b) \leq \gcd(b, r)$

$b, r$ がともに $m$の倍数→ $a = b \times p + r$ も $m$の倍数.
$\gcd(a, b) \geq \gcd(b, r)$

以上から，$\gcd(a, b) = \gcd(b, r)$

これを繰りかえすと、
余りの定義より $b > r$ なのでこの操作は有限回で終わり、
最後は必ず余りが 0になって停止する.
そのときの割った数が、$\gcd(a, b)$になっている.　　q.e.d.

## 美しい補題たち

**最大公約数と最小公倍数の関係**：
正の整数 $a, b$ に対して，それらの最大公約数を $g$，最小公倍数を $l$ とおくと $a \times b = g \times l$

（最大公約数と最小公倍数の積がもとの二つの数の積に等しい）

**ユークリッドの補題１**
自然数 $a, b$ に対してその最大公約数を$d$と置くと、整数 $s, t$ があり、$d = a \times s + b \times t$　と書ける