Euklidischer Algorithmus

Der **Euklidische Algorithmus** ermittelt den größten gemeinsamen Teiler von zwei **natürlichen** Zahlen.

Hintergrund des **Euklidischen Algorithmus** ist stets der **Satz von der Division mit Rest**:

Zu je zwei **natürlichen Zahlen a** und **b** (mit **b** > **0**) gibt es stets **eindeutig** bestimmte nichtnegative ganze Zahlen **q** und **r** mit der Eigenschaft **a = q * b + r** und **0 <= r < b**.
(kann mit vollständiger Induktion bewiesen werden)

Beispiel: Aus **a = 17**, **b = 5** folgt **17 = 3 * 5 + 2** , d.h. **q = 5** , **r = 2**

Wenn **Div(a, b)** die **ganzzahlige Division a / b ohne Rest** und **Mod(a, b)** der **Divisionsrest** von **a / b** ist, dann gilt allgemein immer:

a = Div(a, b) *b + Mod(a, b) , d.h. **17 = Div(17, 5) * 5 + Mod(17, 5) = 3 * 5 + 2** ,

d.h. **q = Div(a, b)** und **Mod(a, b) = r**

Die entscheidende **Idee des Euklidischen Algorithmus** besteht nun darin, den **Satz von der Division mit Rest** nach dem Prinzip der "Wechselwegnahme" zu iterieren.

Dazu ersetzt man nach der Durchführung der Division mit Rest die ursprünglich größere Zahl **a durch die ursprünglich kleinere Zahl b** und **b durch den Rest r**.

Mit diesen **neuen Zahlen a und b** führt man wiederum das **Verfahren der Division mit Rest** durch und erhält ein **neues q** und ein **neues r**.

Mit diesen Zahlen verfährt man wiederum nach dem Prinzip der Wechselwegnahme und nimmt die kleinere so lange von der größeren weg wie es geht.

Bei der Erweiterung des Euklidischen Algorithmus von den natürlichen Zahlen auf die ganzen Zahlen kann mit Absolutbeträgen gearbeitet werden, da der **ggT** immer eine positive Zahl ist.

```
// Rekursiver Algorithmus von Euklid
long ggT(long n, long m){
      if(abs(m)>abs(n)) return abs(ggT(m,n)); // vertauschen von n und m
      if(!m) return abs(n);                   // if(m == 0) return abs(n), Abbruch
      return abs(ggT(m,n%m));                 // rekursiver Schritt
}

// Iterativer Algorithmus von Euklid
long ggT(long a, long b){
      a = abs(a); b=abs(b);
      while(a*b){
                  if(a >= b) a = a % b;
                  else b = b % a;
     }
     return a + b;
}
```

Rechnen Sie beide Algorithmen mit dem Parameterpaar 6 , 15 manuell aus.