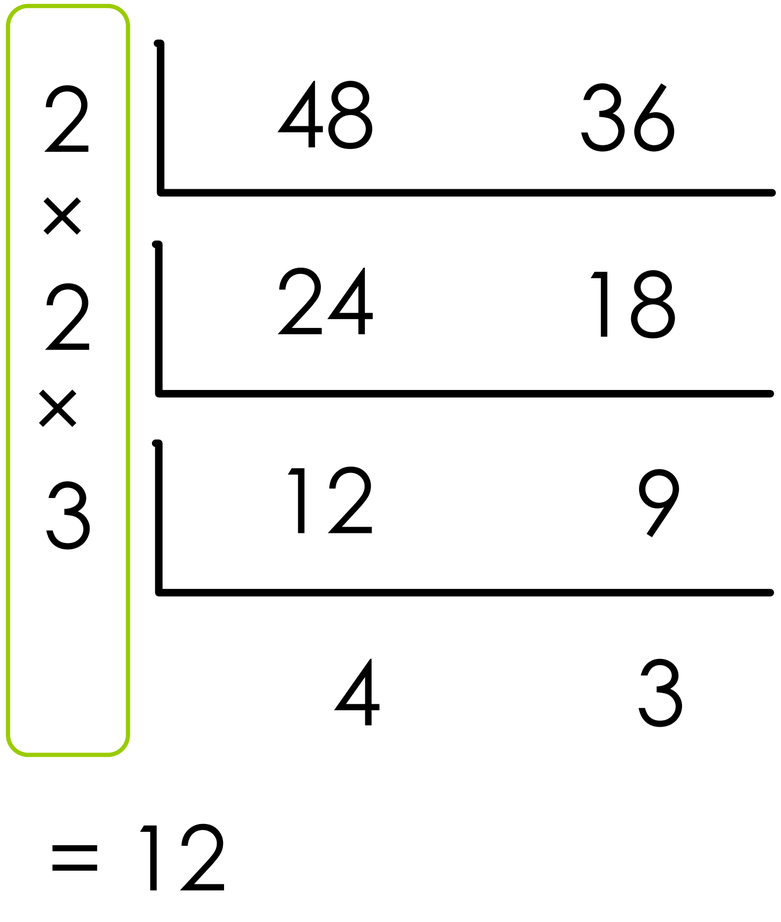
例題：最大公約数を求めるには？

最大公約数の求め方

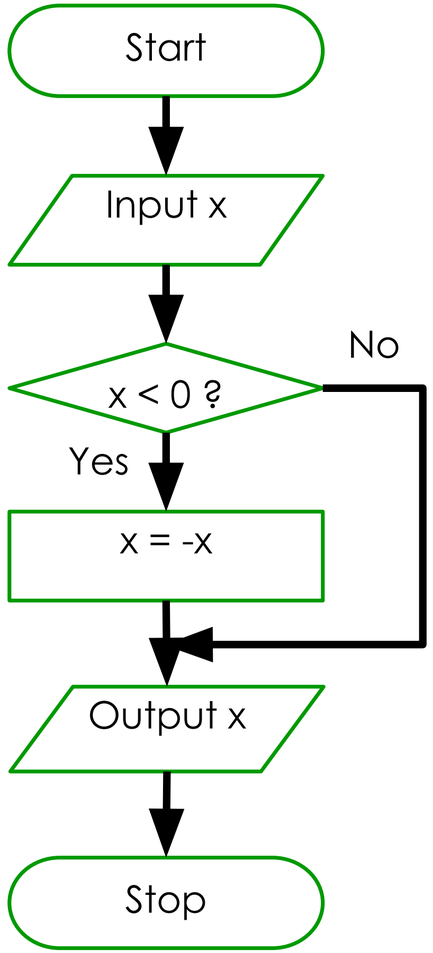
- 2つの自然数が与えられた時に、それらの最大公約数を求めるには、どのようにすれば良いでしょうか？
- 小学校では、次のような手順を教わりましたね？
  1. 両方の数を割り切れる数を探し、両方の数を割る
  2. 1で割り切れる数がなくなるまで繰り返す
  3. 1で探し当てた数を、すべて掛け合わせる



フローチャート

- アルゴリズムを記述する際には、しばしば「フローチャート」というものも用いられます
- 以下の部品を、矢印でつなぎ合わせて組み合わせて記述します
  - 端点（角丸の長方形）：「開始」や「終了」を表す
  - 処理（長方形）：具体的な手順を表す
  - 判断（ひし形）：条件分岐を表す
  - 入出力（平行四辺形）
- 例えば、絶対値を返す処理をフローチャート で表すと、右のようになります

絶対値を返す処理の記述例

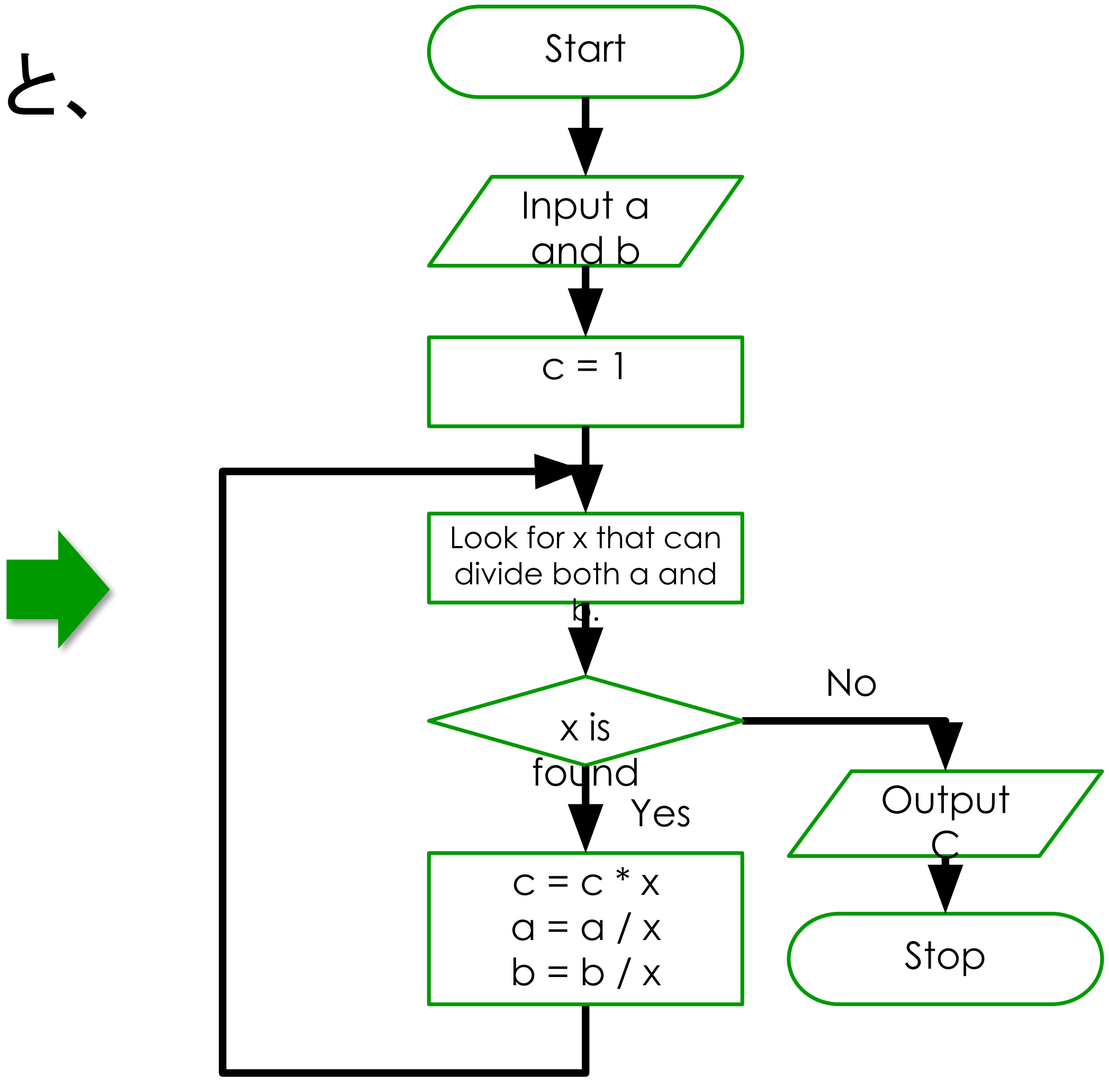


最大公約数のフローチャート

- 先程の手順をフローチャートで記述すると、右図のようになります

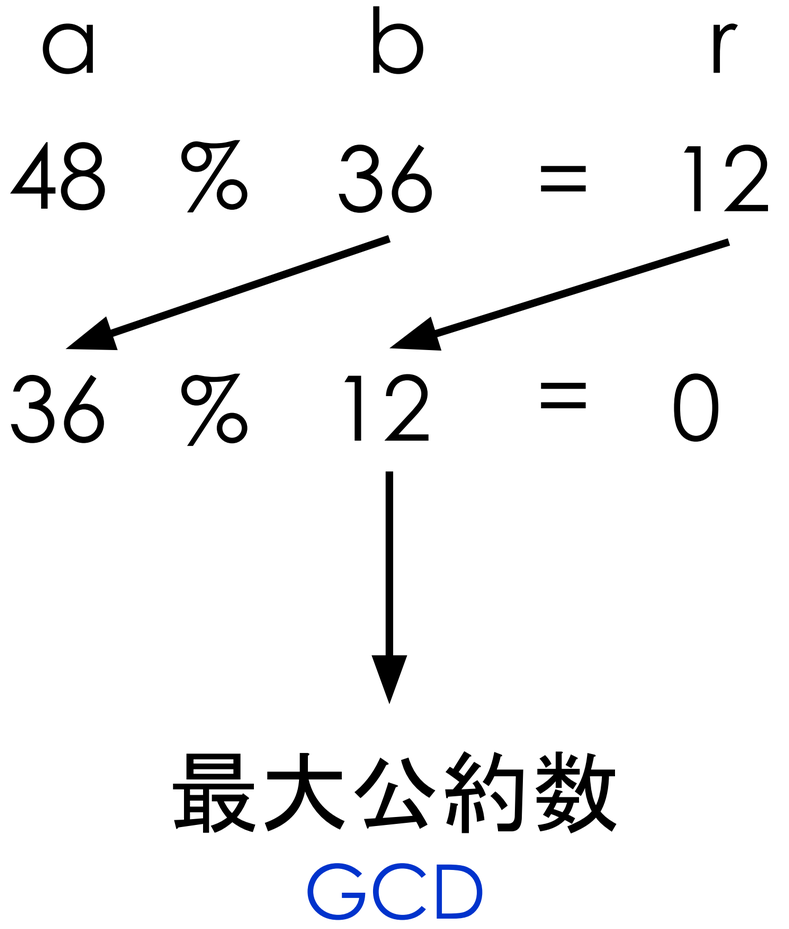
2	48	36
×		
2	24	18
×		
3	12	9
	4	3

= 12



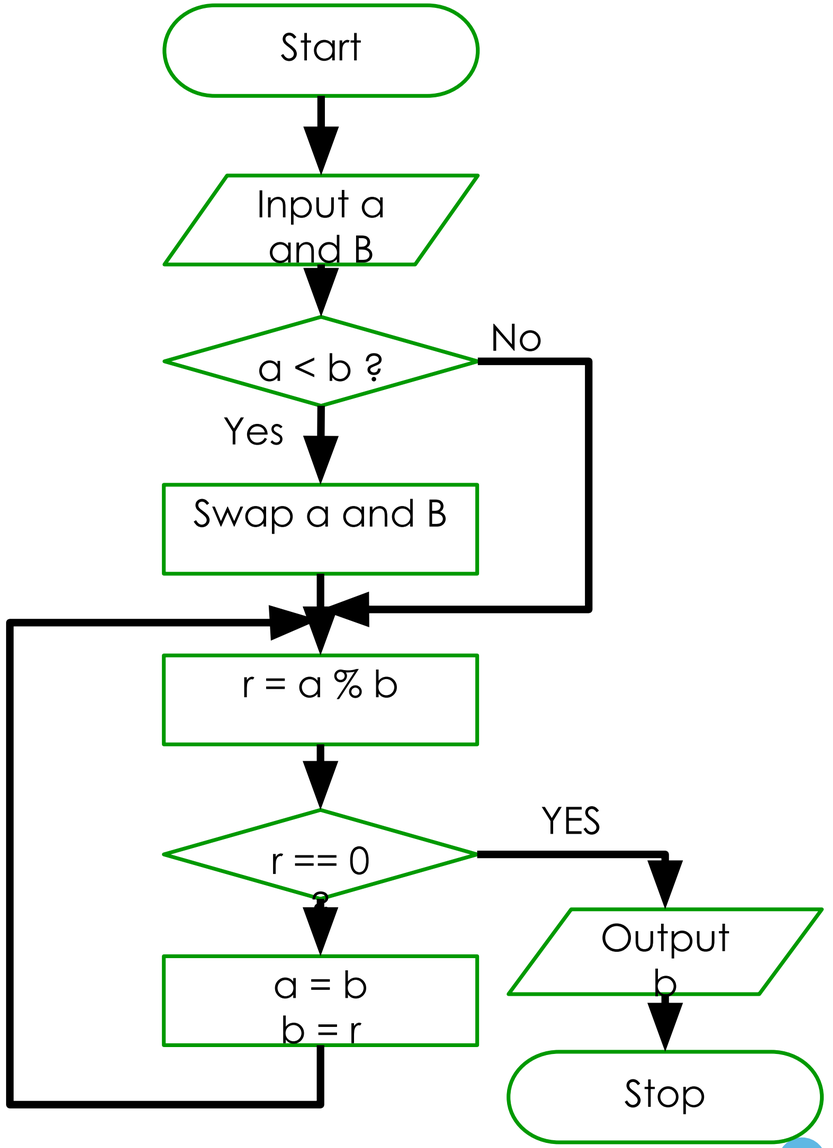
ユークリッドの互除法

- 実際には、もっと効率的な手法として、次のような手順が知られています
  1. 大きい方の数 a を、小さい方の数 b で割り、余り r を求める
  2. 余り r が0の場合は、小さい方 b の数が最大公約数となる
  3. それ以外の場合は、bを大きい方の数、rを小さい方の数として、1から繰り返す



ユークリッドの互除法のフローチャート

- このアルゴリズムをフローチャートで記述すると右図のようになります
- 「ユークリッドの互除法」と呼ばれています
  - 紀元前からある最古の「アルゴリズム」の1つとして知られています
  - ※当たり前ですが、コンピュータはありませんね

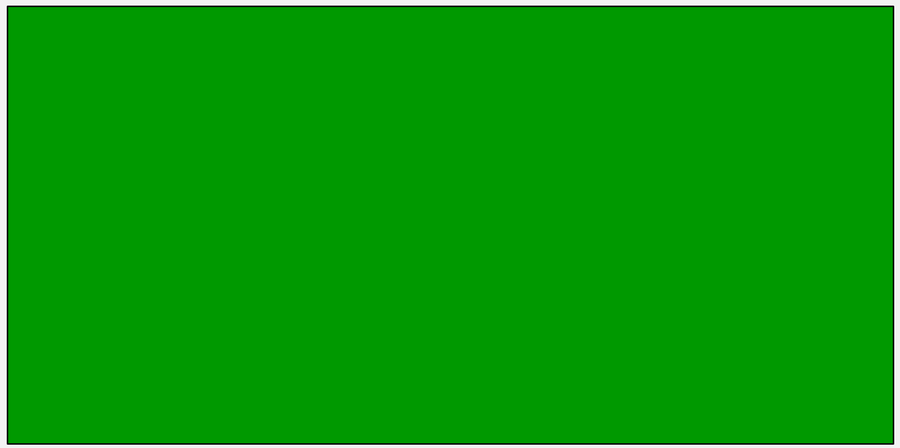


ユークリッドの互除法をPythonで書いてみよう

- 正の整数 a と b を引数にとり、最大公約数を返す関数 gcd(a,b) を定義しなさい

ユークリッドの互除法

[2]:



```
[3]: gcd(24, 32)
```

```
[3]: 8
```

手順がわかれば、プログラムに直すのは簡単ですね？

```
def gcd(a, b):
    if a < b:
        tmp = a
        a = b
        b = tmp
    while True:
        r = a % b
        if r == 0:
            return b
        a = b
        b = r
```