

(greatest common divisor) of numbers a, b

12 and 15 -> 3

min(12,15)->12


12	12%12	15%12 no
11	12%11	15%11 no
10		
...		
3	12%3	15%3 yes break

```
void GCD(int number, int secondNum) { // 12 and 15
--> 3
// need to keep track of divisor for each number
// need to compare divisor of each number
// need to find greatest divisor for both numbers
int min;
min = (number > secondNum) ? (secondNum):(number);
for (int i = min; i >= 1; i--) {
    if (number % i == 0 && secondNum % i == 0) {
        std::cout << i << std::endl;
        break;
    }
}
}
```

GCD(18,12)=2*3
 18=2*9=2*3*3
 12=2*6=2*2*3
 LCM(18,12)=2*2*3*3

least common multiple LCM
 $5/12+7/18=(5*3+7*2)/36$
 GCD(18,12)*LCM(18,12)=18*12
 GCD(a,b)*LCM(a,b)=a*b
 LCM(a,b)=a*b/GCD(a,b)

Первый алгоритм



Как-то раз древнегреческого геометра Евклида спросили: - Что бы ты предпочел - два целых яблока или четыре половинки? - Конечно, четыре половинки. - А почему? Это ведь одно и то же. - Отнюдь. Выбирая два целых яблока, как я узнаю, червивые они или нет?

Алгоритм Евклида – это алгоритм нахождения НОД двух целых положительных чисел

$\text{min} = (\text{number} > \text{secondNum}) ? (\text{secondNum}):(\text{number});$

```
if(number>secondNum)
{
    min = secondNum;
}
else
{
    min=number;
}
```

fundamental theorem of arithmetic
 exist single factorisation on prime numbers

$12=2*2*3$
 $12737812491=117*111^2=17^5*13$

7462
6279

homework

7462|6279
6279|1
1183

6279|1183
5915|5
364

1183|364
1092|3
91

364|91
364|4
0

hw create program to calculate GCD using Euclid's algorithm

7462%6279->1183
6279%1183- > 364

```
void euclidGCD(int number, int secondNum) {
int min;
int max;
int temp;
if (number < secondNum) {
    max = secondNum;
    min = number;
}
else {
    max = number;
    min = secondNum;
}
while (1) {
    temp = min;
    min = max % min;
    max = temp;
    if (min == 0) {
        std::cout << max << std::endl;
        break;
    }
}
}
```