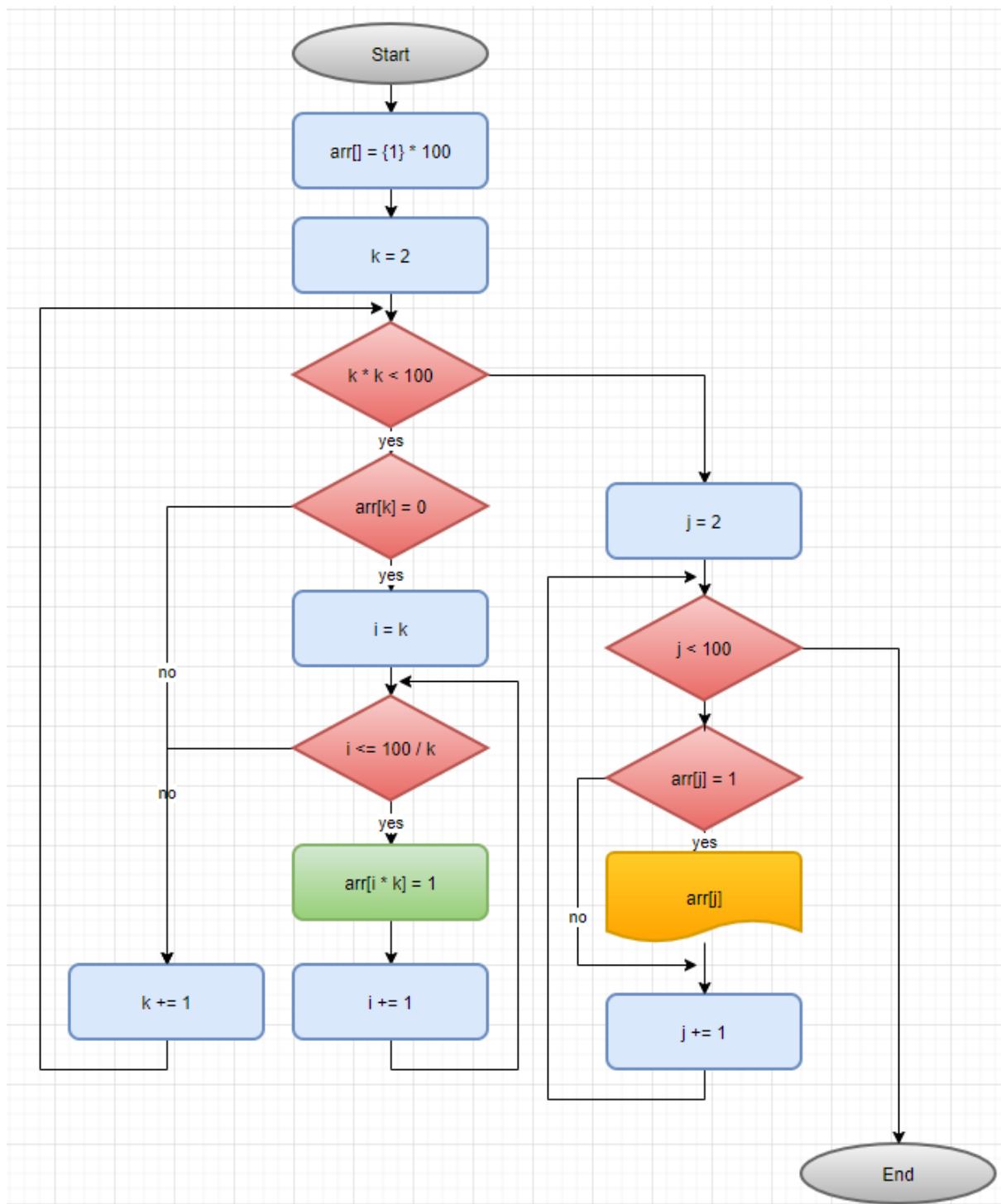


Day26; 20220804

🕒 작성일시	@2022년 8월 4일 오전 9:27
▼ 강의 번호	Lesson 26
📅 유형	@2022년 8월 4일
🔗 자료	
☑ 복습	<input type="checkbox"/>
☰ 속성	

Sieve of Eratosthenes 에라토스테네스의 체



```

package day26_1;

import java.util.Arrays;

public class Eratosthenes {

    public static void main(String[] args) {

```

```
// TODO Auto-generated method stub
// 소수만 찾아내는 알고리즘
// 101의 배열을 만들어 1로 초기화
int[] arr = new int[101];

int k = 2;

while(k * k < 100) {
    if(arr[k] == 0) {
        int i = 2;
        while( i <= 100 / k) {
            arr[k * i] = 1;
            i++;
        }
    }
    k++;
}

int sum = 0;
for(int j = 2; j < 101; j++) {
    if(arr[j] == 0) {
        System.out.print(j + " ");
    }
}
}
```

유클리드 알고리즘 Euclidean Algorithms

- 최대 공약수를 구하는 알고리즘
- 최대 공약수는 약수들 중에서 가장 큰 수를 말한다.

반복 구조를 이용하는 중요한 알고리즘

- 최대 공약수

약수

3의 약수 - 1, 3

4의 약수 - 1, 2, 4

5의 약수 - 1, 5

6의 약수 - 1, 2, 3, 6

8의 약수 - 1, 2, 4, 8

12의 약수 - 1, 2, 3, 4, 6, 12

공약수

8과 12의 공통된 약수 - 1, 2, 4

최대 공약수

두 수의 공약수 중 최대값, 8과 12의 최대 공약수는 4이다.

최대 공약수를 구하는 절차

먼저 어떤 복수의 수를 **소수의 곱셈** 형태로 분해하자. - 소인수 분해

$$8 = 1 * 2 * 2 * 2$$

$$12 = 1 * 2 * 2 * 3$$

이들 중 공통되는 소수를 서로 곱한 수가 바로 두 수의 최대 공약수이다.

$$1 * 2 * 2 = 4$$

그러나 이런 절차를 반복하는 것은 상당히 복잡하다.

어떤 수를 소인수 분해하려면 먼저 그 수 이하의 소수들을 모두 구해야 한다.

그리고 그 소수 중 작은 숫자부터 순서대로 원래의 수를 나누고, 나누어지지 않으면 그 다음 소수의 순서로 계속 계산을 반복해야한다.

따라서 단순해 보이지만 절차는 상당히 복잡해진다.

이러한 복잡성에 비해 매우 간단한 방법으로 최대 공약수를 구하는 것이 바로 '유클리드 알고리즘'이다.

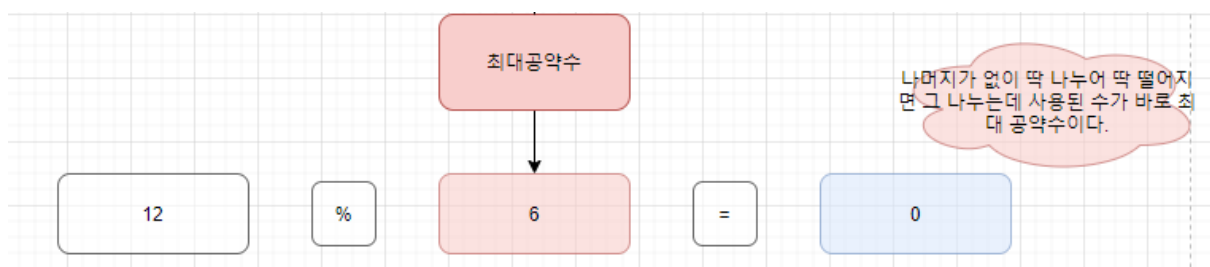
유클리드 알고리즘이란?????

약 2300년 전의 고대 그리스의 수학자로 수많은 수학적 이론을 생각해냈다. 그 중 하나가 바로 유클리드 알고리즘이다. 간단히 말하면 두 수의 나눗셈을 반복하여 최대 공약수를 구하는 것이다.

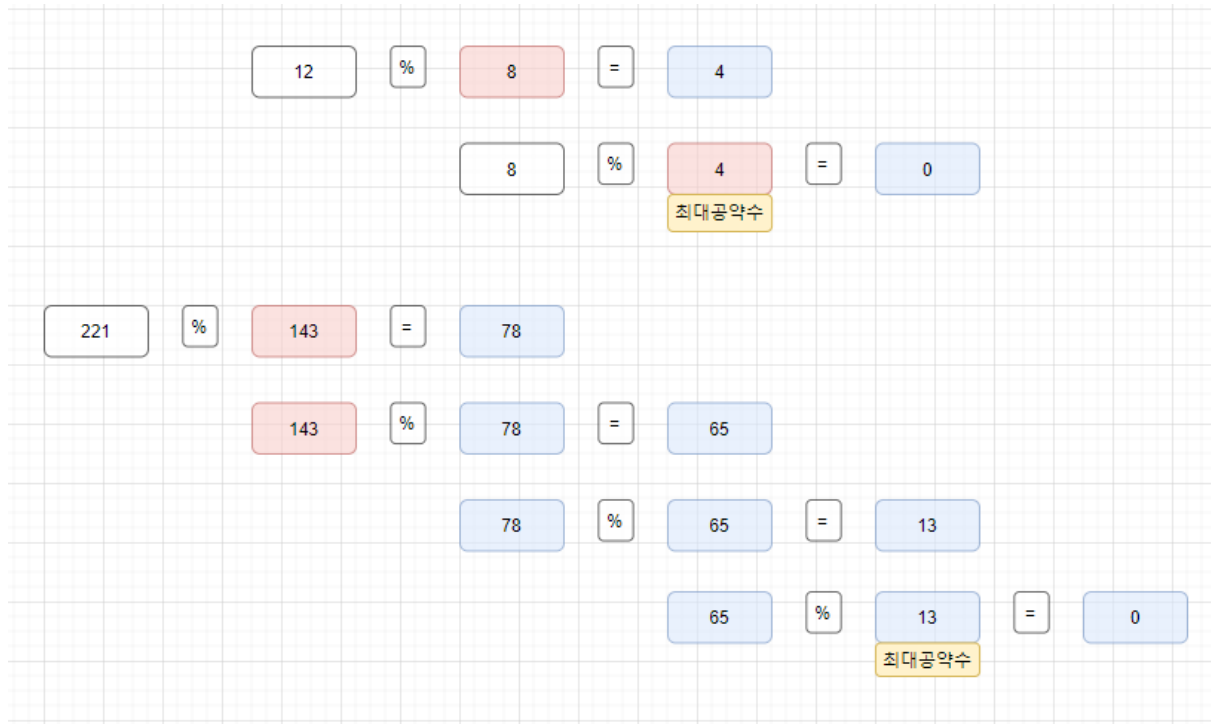
그러면 어떻게 나눗셈을 반복할까....

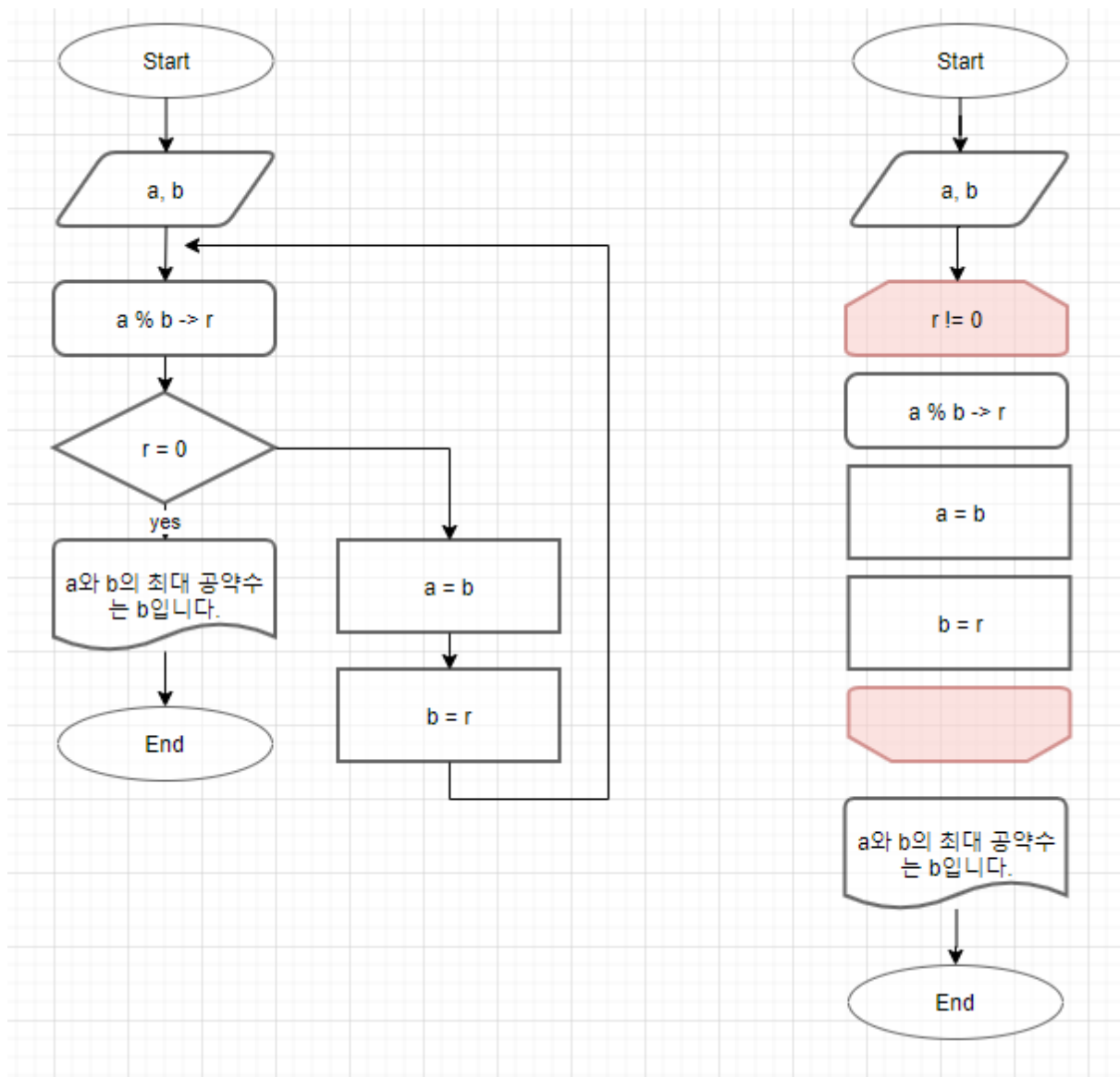
먼저 큰 수를 작은 수로 나눈다. 나머지가 나오지 않게 되면 (%) 그때의 나누는데 사용된 작은 수가 바로 최대공약수이다.

예를 들면 12와 6의 최대 공약수는 6이다.



나머지가 0이 될 때까지 작업을 반복해준다.





- 코드로 구현하기

```
1 package day26_2;
2
3 import java.util.Scanner;
4
5 public class GreatestCommonDivisor {
6
7     public static void main(String[] args) {
8         // TODO Auto-generated method stub
9         Scanner stdIn = new Scanner(System.in);
10        System.out.println("숫자를 입력하세요.");
11        int a = stdIn.nextInt();
12        int aa = a;
13        int b = stdIn.nextInt();
14        int bb = b;
15        int r = a % b;
16
17        if( r == 0) {
18            System.out.printf("%d와 %d의 최대 공약수는 %d입니다.", aa, bb, b);
19        }else {
20            while(r != 0) {
21                r = a % b;
22                a = b;
23                b = r;
24            }
25            System.out.printf("%d와 %d의 최대 공약수는 %d입니다.", aa, bb, a);
26        }
27    }
28 }
29
30
```

<terminated> GreatestCommonDivisor [Java Application] C:\Program Files\Java\jdk-11.0.15\bin\javaw.exe
숫자를 입력하세요.
221
143
221와 143의 최대 공약수는 13입니다.

```
package day26_2;

import java.util.Scanner;

public class GreatestCommonDivisor {

    public static void main(String[] args) {
        // TODO Auto-generated method stub
        Scanner stdIn = new Scanner(System.in);
        System.out.println("숫자를 입력하세요.");
        int a = stdIn.nextInt();
        int aa = a;
        int b = stdIn.nextInt();
```



```
int bb = b;
int r = a % b;

if( r == 0) {
    System.out.printf("%d와 %d의 최대 공약수는 %d입니다.", aa, bb, b);

}else {
    while(r != 0) {
        r = a % b;
        a = b;
        b = r;
    }
    System.out.printf("%d와 %d의 최대 공약수는 %d입니다.", aa, bb, a);
}
}
```
