스크래치

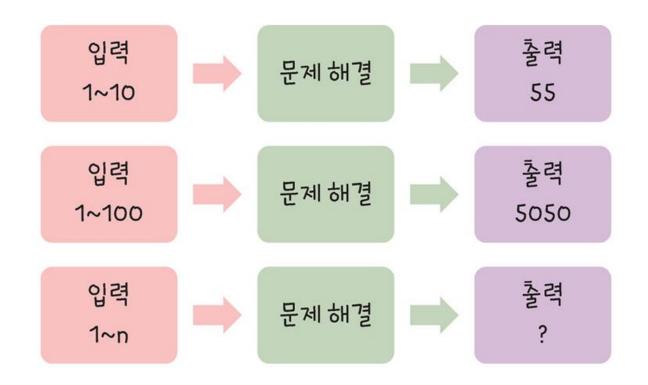
DAY 06 - 1부터 n까지 더하기

```
#include <iostream>
using namespace std;

int main() {
   cout<<"Hola Facebook\n";
   return 0;
}</pre>
```

1. 1부터 n까지 합 이해하기

- - 1부터 10까지 합은 간단히 계산 가능
 - 1부터 100까지 합은 어렵지만 어떻게든 구할 수 있음
- 그렇다면 1부터 n까지 합을 구하려면 어떻게 계산해야 할까?



1. 1부터 n까지 합 이해하기

- 알고리즘에서 가장 핵심: 효율성
 - 컴퓨터가 계산하더라도 수행 횟수를 줄여 좀 더 빠르게!
- 두 가지 방법
 - 1) 순서대로 더하기
 - 2) 첫째 항 1과 마지막 항 n을 더한 후 n/2을 곱하기



알고리즘 1. 1부터 순서대로 더하기



■ 1부터 10까지 합

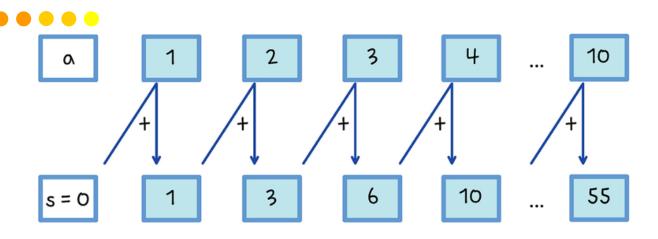
■ 1부터 100까지 합

■ 1부터 n까지 합



규칙 찾아보기





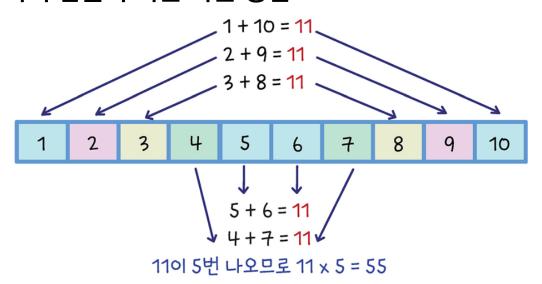
- a: 숫자를 1씩 증가시키는 데 필요한 변수
- s: a를 누적시키는 데 필요한 변수
- n: 어디까지 더할지를 알려 주는 데 필요한 변수

$$\alpha = 1$$
, $s = 0$, $n = 10$

반복 실행

알고리즘 2. 첫째 항과 마지막 항 n을 더한 후 n/2을 곱하기 ::

■ 1부터 10까지 합을 구하는 다른 방법

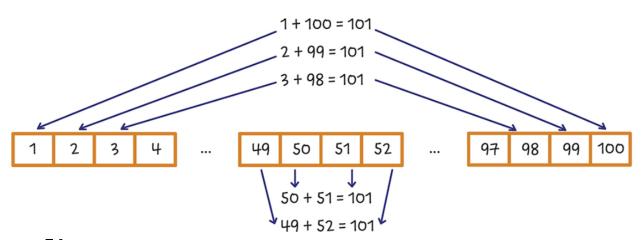


■ 식으로 표현

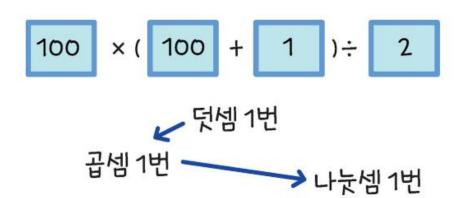
알고리즘 2



■ 1에서 100까지 합 구하기



■ 식으로 표현

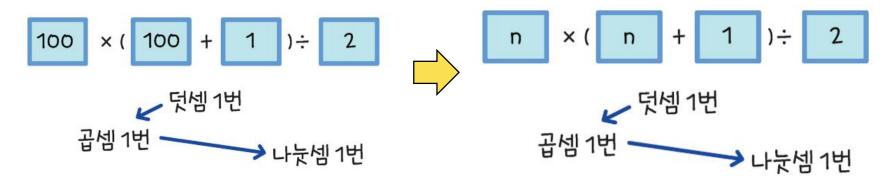




알고리즘 2



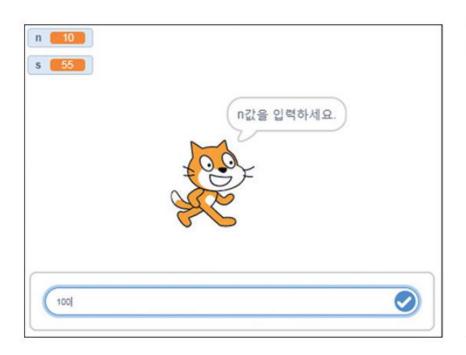
■ 1부터 n까지 합



- n 변수: 몇까지 합을 구할지 결정하는 변수
- s 변수: n값을 사용하여 합계를 저장할 변수, 합계를 출력합니다.

3. 스크래치에서 1부터 n까지 더하기

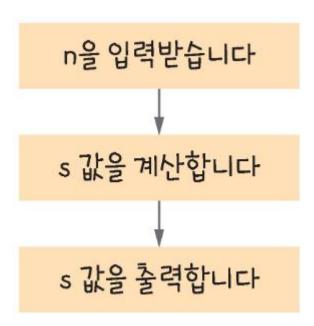
스크래치에서 n값을 입력받아 1부터 n까지 합을 구해 보자. n=10 이면 1부터 10 까지 합으로 55를 표시하고, n=100 이라면 1부터 100까지 합으로 5050을 표시한다. 사용자가 n 값을 자유롭게 입력하고, 그 결과로 나온 합을 표시하도록 한다.





3. 스크래치에서 1부터 n까지 더하기

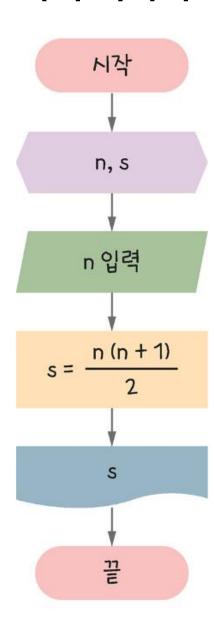
- 먼저 어떤 변수가 필요한지 고려
 - n 변수: 몇까지 합을 구할지 결정하는 변수
 - s 변수: n값을 사용하여 합계를 저장할 변수
- 실행 과정





3. 스크래치에서 1부터 n까지 더하기

■ 알고리즘을 순서도로 표현



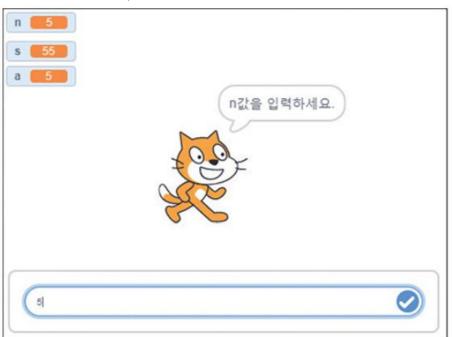
스크래치로 구현하기

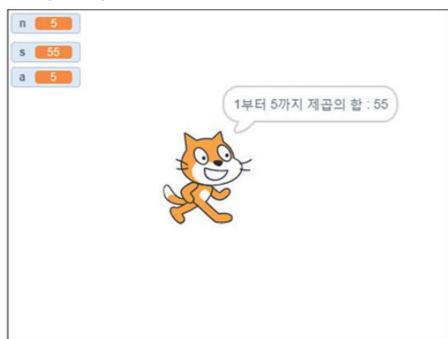




4. 연습 문제

■ 1부터 n까지 제곱의 합(12+22+32+ ··· + n2)을 두 가지 방식의 알고리즘으로 풀어 보고, 그중 효율적인 알고리즘으로 구현해 보자.





- 힌트
 - 1부터 10까지 제곱의 합은 385이다.
 - 수식은 $\frac{n(n+1)(2n+1)}{6}$ 이다.

스크래치

DAY 07 - 최대값 찾기

```
#include <iostream>
using namespace std;

int main() {
   cout<<"Hola Facebook n";
   return 0;
}</pre>
```

1. 최댓값이란?



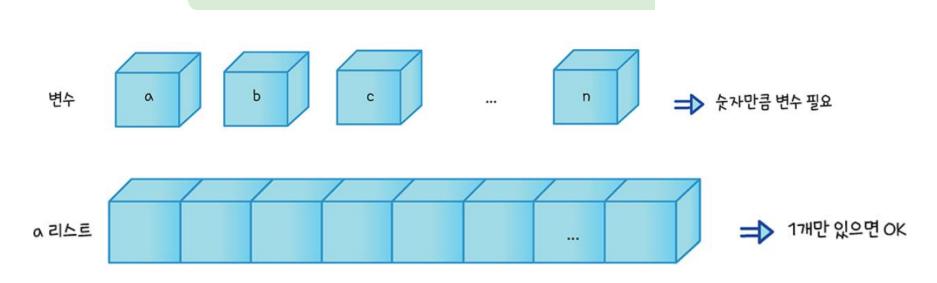
■ 최댓값이란 주어진 숫자 중에서 가장 큰 값을 의미

35 20 10 40 15

- 주어진 데이터 5개를 저장할 공간이 필요
- 이때 변수 5개를 사용하기보다는 좀 더 간단한 리스트를 사용해보자.

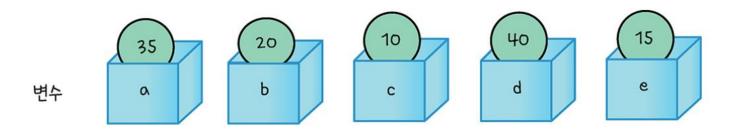
2. 리스트 이해하기

- 데이터 여러 개를 저장하는 공간으로 리스트가 유용함
 - 변수: 데이터 1개를 저장하는 공간
 - 리스트: 데이터 여러 개를 저장하는 공간

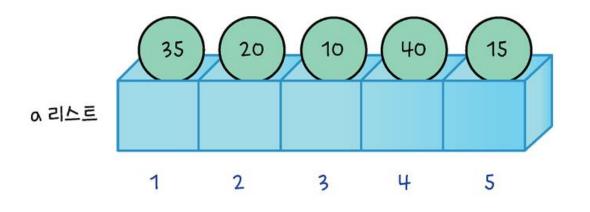


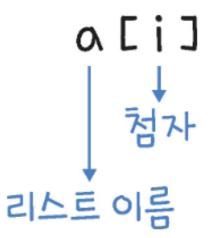
2. 리스트 이해하기

변수를 이용하여 5개 숫자 저장 → 5개 변수 필요



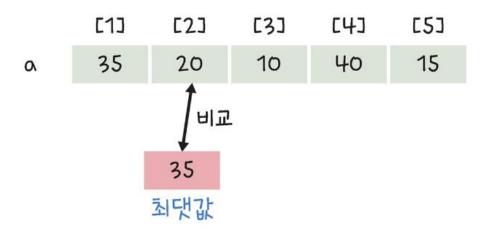
■ 리스트를 사용하여 5개 숫자 저장 → 1개 리스트 필요

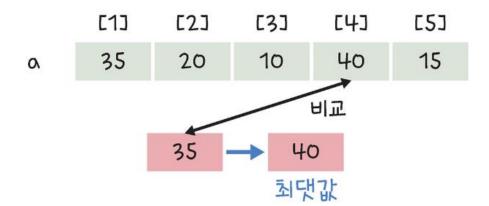


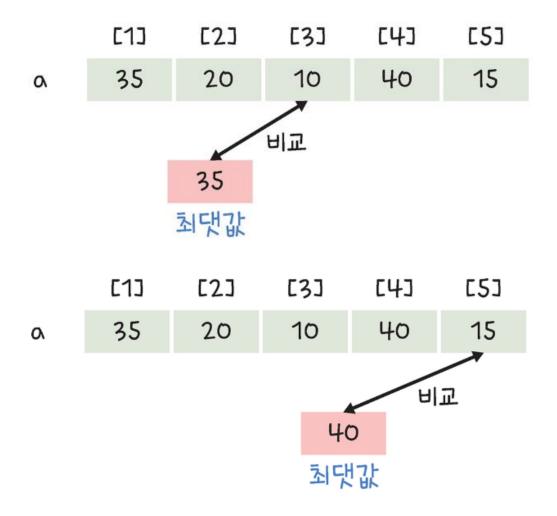




	[1]	[2]	[3]	[4]	[5]
Ο	35	20	10	40	15
	[47]	F27	[27]	CII 7	F E 7
	[1]	[2]	[3]	[4]	[5]
Ο	35	20	10	40	15
35					
	2	비댓값			



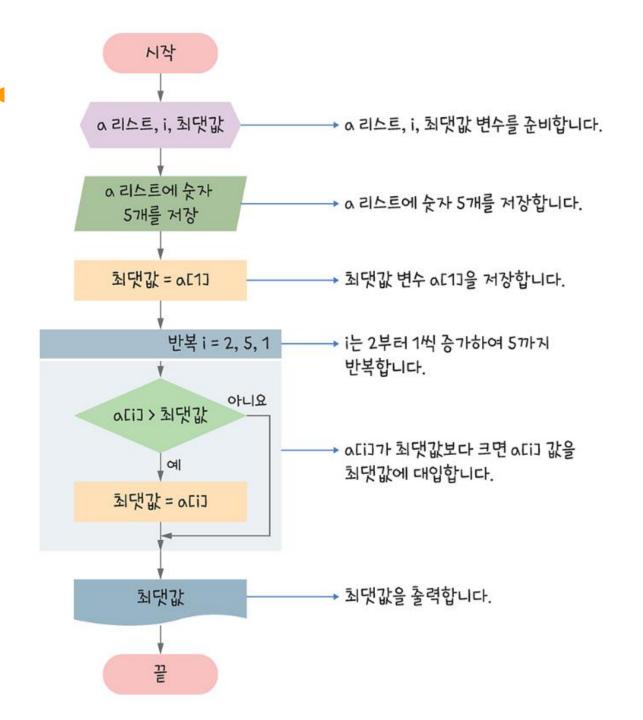






- $\bullet \bullet \bullet \bullet \bullet$
 - 알고리즘 정리
 - 11 a 리스트를 만듭니다.
 - 2 a 리스트에 데이터 5개를 저장합니다.
 - 3 a[1]을 최댓값에 저장합니다.
 - 4] 다음을 반복합니다.
 - a[2]와 최댓값을 비교하여 큰 값을 최댓값에 저장합니다.
 - a[3]과 최댓값을 비교하여 큰 값을 최댓값에 저장합니다.
 - a[4]와 최댓값을 비교하여 큰 값을 최댓값에 저장합니다.
 - a[5]와 최댓값을 비교하여 큰 값을 최댓값에 저장합니다.
 - 5 최종적으로 최댓값을 출력합니다.





4. 스크래치에서 리스트 만들기

[변수] 〉[리스트 만들기]를 클릭하면 새로운 리스트 입력 창이 나타난다.





리스트의 값을 수정할 때 사용하는 블록들

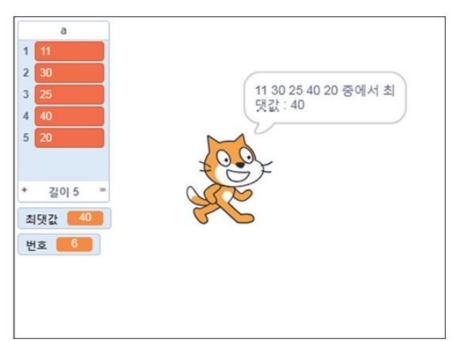




5. 스크래치에서 최댓값 찾기

 리스트에 숫자 11, 30, 25, 40, 20을 저장하고 최댓값을 찾는 스크래치를 구현해 봅시다.







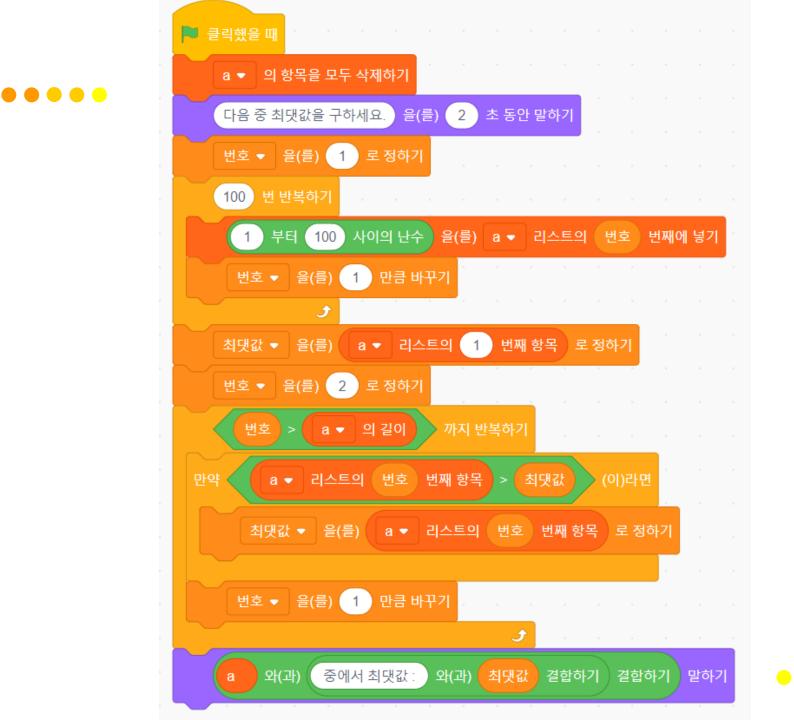
6. 응용하기

- 임의의 난수를 100개 마련하여 그 중에서 최댓값을 구해보자.
 - 난수는 예측할 수 없는 무작위 수를 의미
 - 1부터 100사이 난수는 1부터 100사이 숫자를 불규칙하게 생성



- 임의의 수 100개 준비하기
 - 반복문을 사용하여 간단히 구현





7. 연습 문제

 1~100까지 난수 데이터 5개를 리스트에 저장한 후 가장 작은 숫자를 찾는 알고리 즘을 구현해 보세요.



스크래치

DAY 08 - 소수 구하기

```
#include <iostream>
using namespace std;

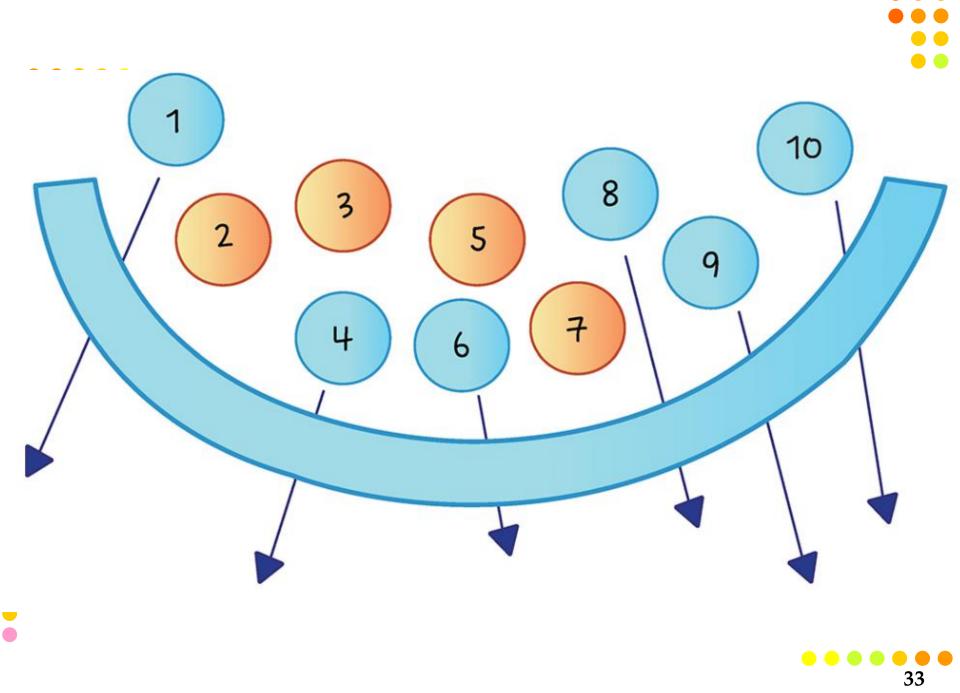
int main() {
   cout<<"Hola Facebook\n";
   return 0;
   return 0;</pre>
```

1. 소수란?



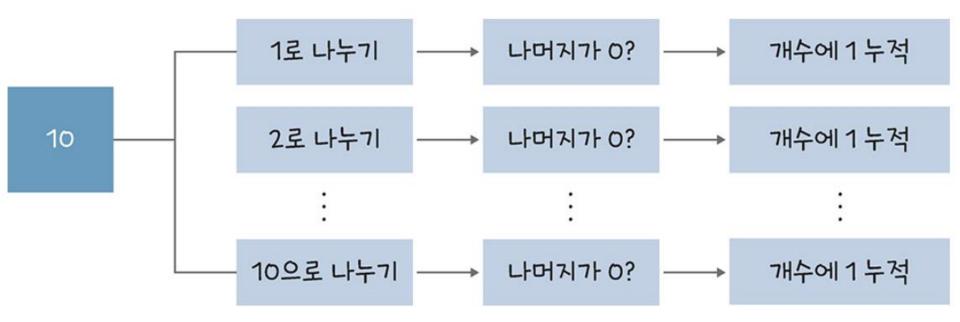


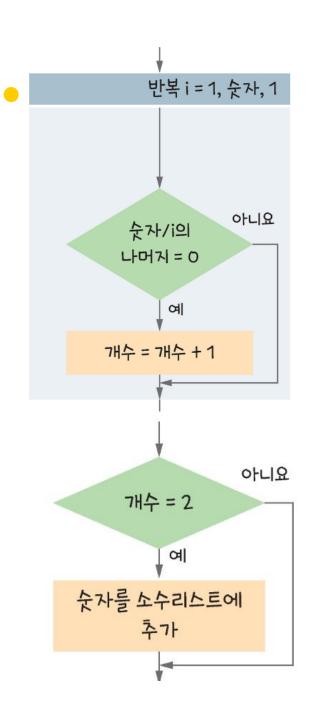
1 (2) (3) 4 (5) 16 (7) 18 97 10

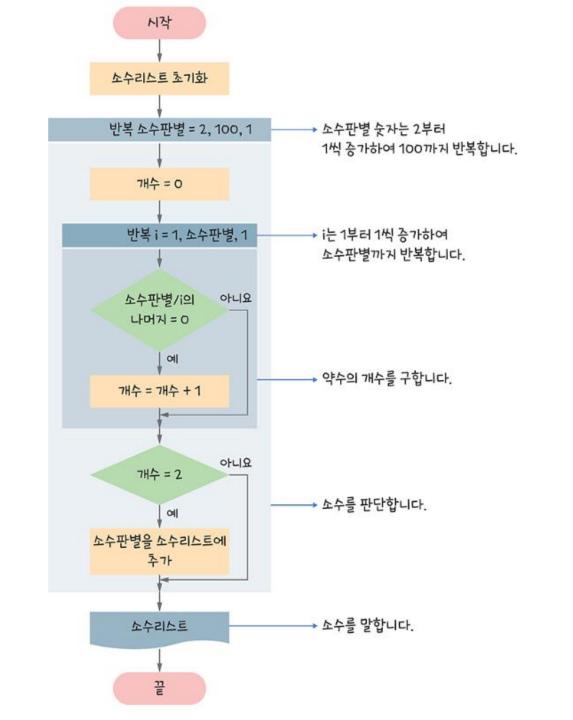


2. 소수 구하기 알고리즘



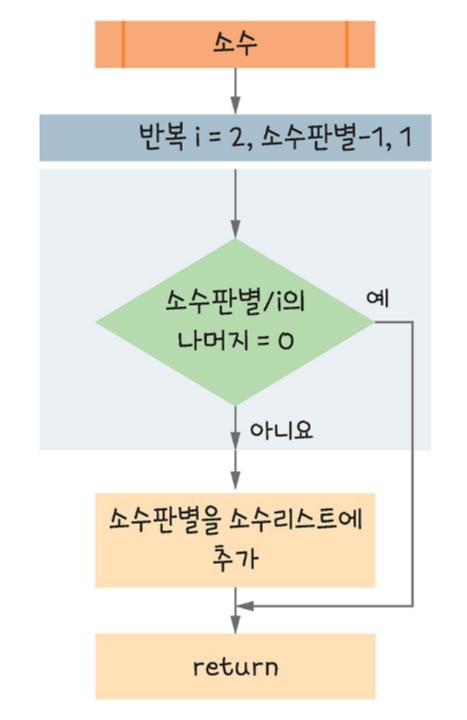






	6 나누기 2	나머지=
6은 소수일까요?	6 LF 7 1 4	
	6 LF 71 5	

 $\bullet \bullet \bullet \bullet \bullet$



• • • •

3. 알고리즘 선택하기

■ 알고리즘 2를 이용하여 스크래치로 구현해 보겠습니다.

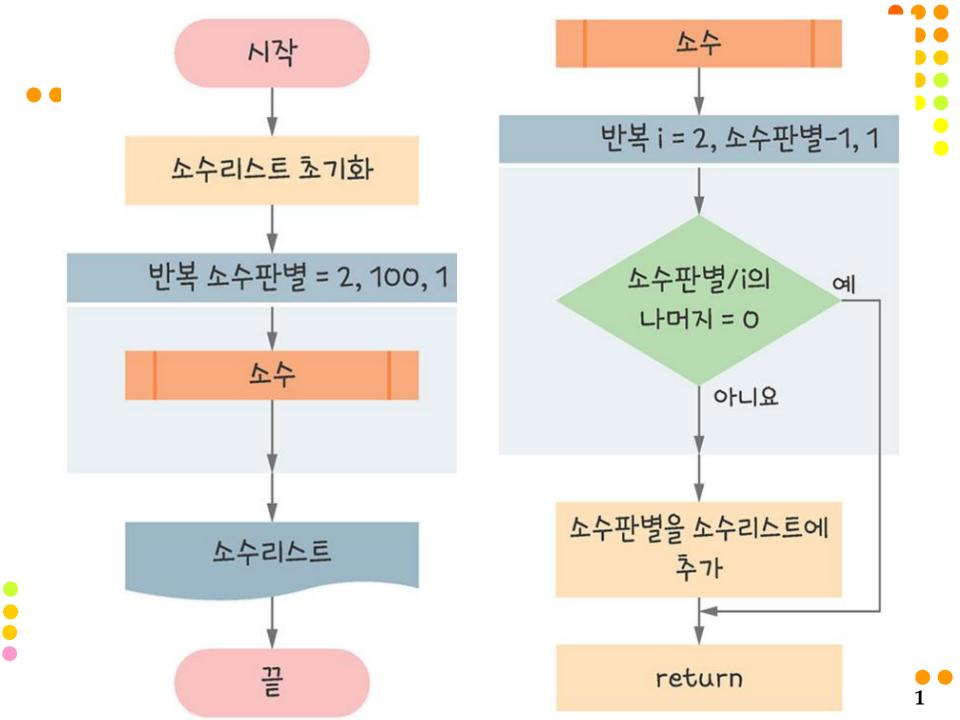


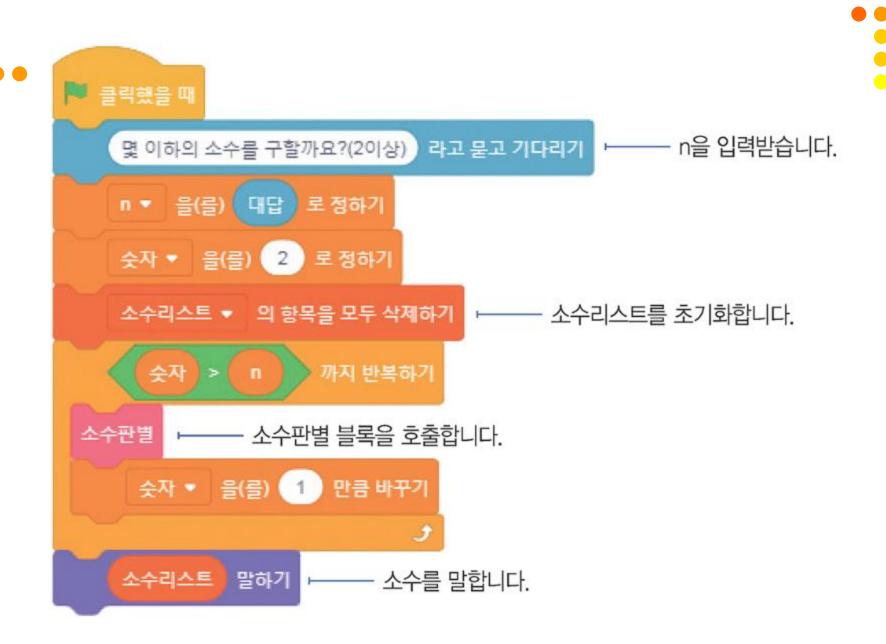
4. 스크래치에서 소수 구하기

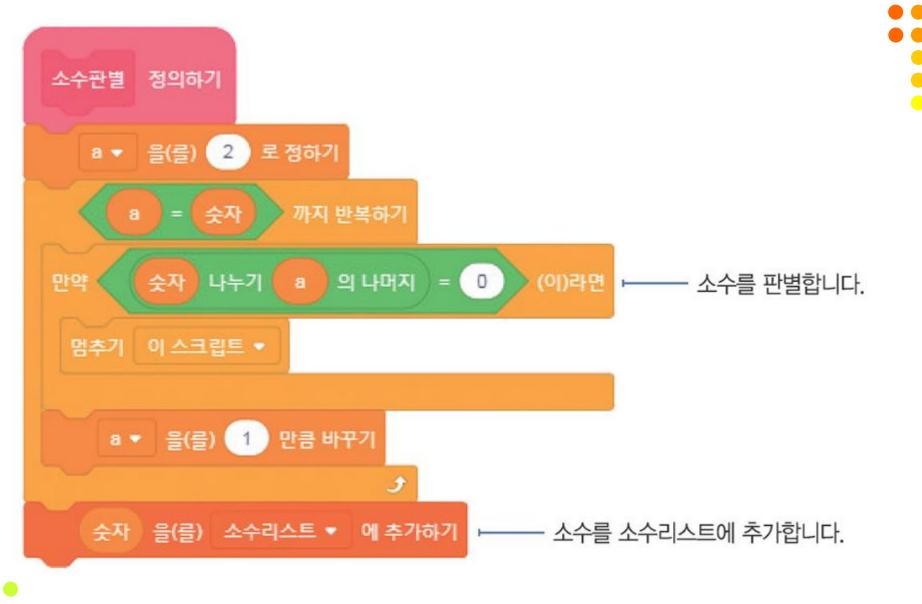






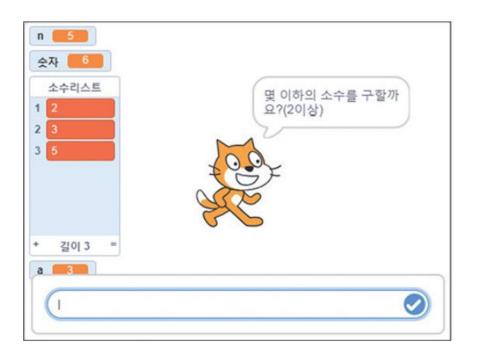


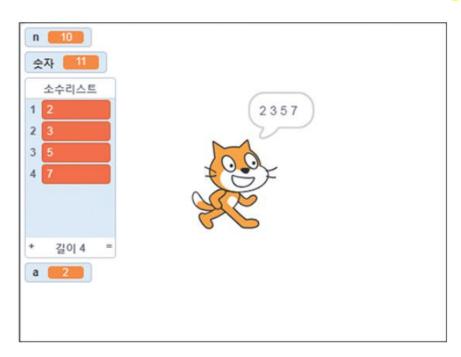




5. 연습 문제

■ 1부터 100까지 숫자 중 소수를 모두 나열하세요.







스크래치

DAY 09 - 피보나치 수열

```
#include <iostream>
using namespace std;

int main() {
   cout<<"Hola Facebook\n";
   return 0;
}</pre>
```

1. 피보나치 수열이란?



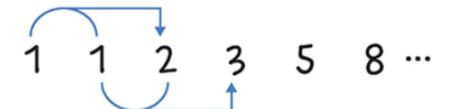
(n-2)번째 피보나치수



(n-1)번째 피보나치 수

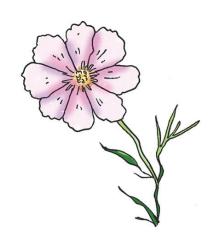


n번째 피보나치 수







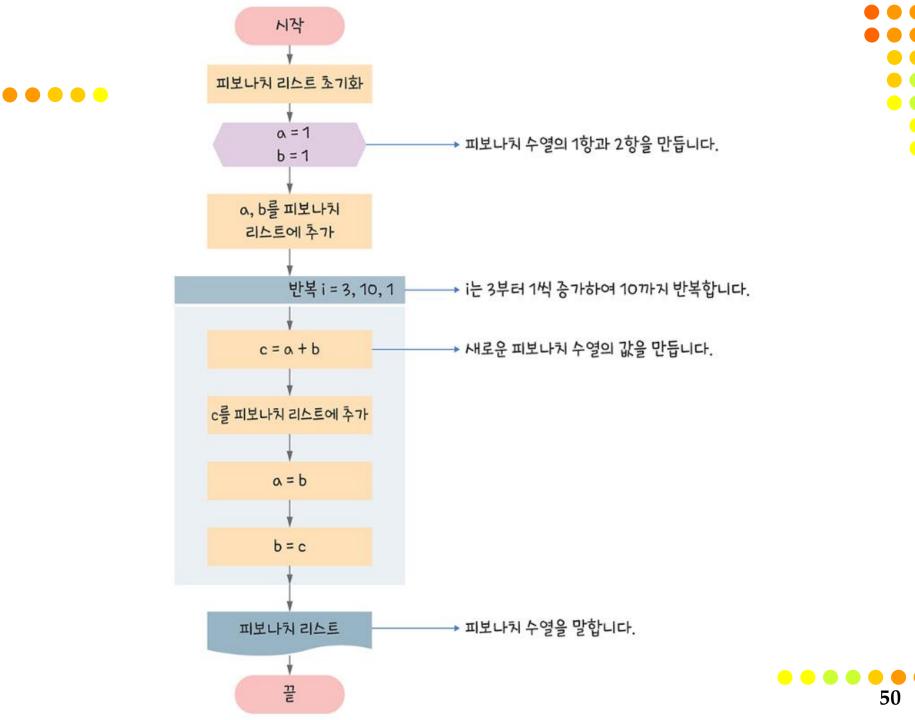


	개월 수	토끼	토끼(쌍)		
	처음		1쌍		
	1개월		1쌍		
	2개월	A B B C C C C C C C C C C C C C C C C C	2쌍		
	3개월	A C B OF O	3쌍		
	4개월	A D C B E	5 <i>1</i> 6		
	5개월	A F D C G B H E If A A A A A A A A A A A A A A A A A A A	8 <i>#</i> }		

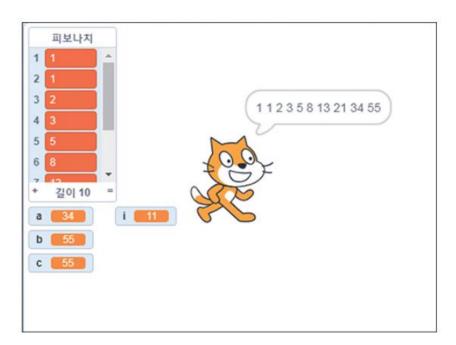
2. 리스트로 구현한 피보나치 수열



항	피보나치 값
1	1
2	1 1
3	1 1 2
4	1 1 2 3
5	1 1 2 3 5
6	1 1 2 3 5 8
7	1 1 2 3 5 8 13
8	1 1 2 3 5 8 13 21
9	1 1 2 3 5 8 13 21 34
10	1 1 2 3 5 8 13 21 34 55









3. 블록을 이용한 피보나치 수열

• 일고리즘

먼저 진행 순서를 정리하면 다음과 같습니다.

- 1 n 값을 입력받습니다.
- 2 | 피보나치 수열 블록을 호출합니다.
- 3 | 피보나치 리스트를 표시합니다.

피보나치 수열의 블록은 다음과 같습니다.

- 1 피보나치 수열을 저장할 리스트를 초기화합니다.
- 첫째 항(a)과 둘째 항(b)에 1을 저장합니다.
- 3 a 값과 b 값을 피보나치 리스트에 추가합니다.
- 4 i는 3부터 10까지 반복합니다.

c=a+b

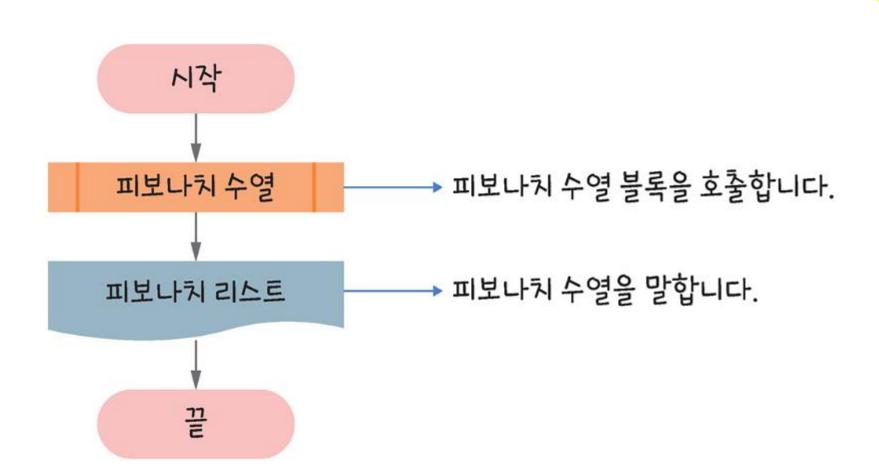
c를 피보나치 수열에 추가합니다.

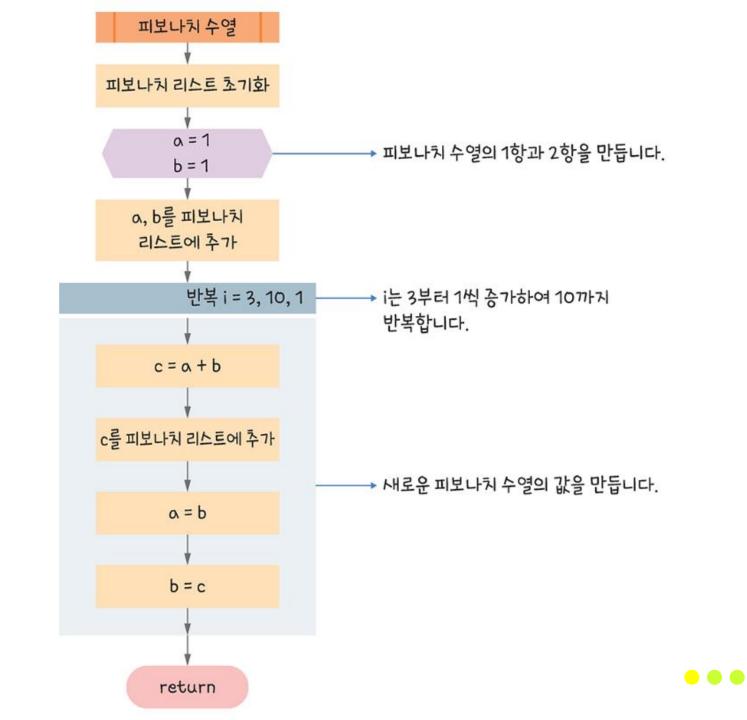
a=b

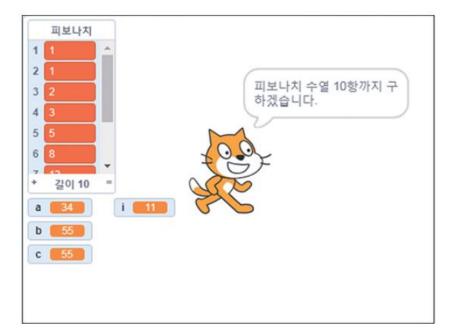
b=c

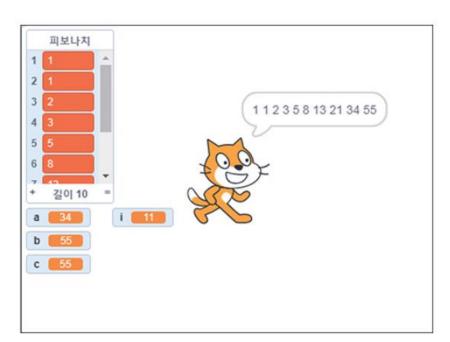
이것을 순서도로 표현하면 다음과 같습니다.









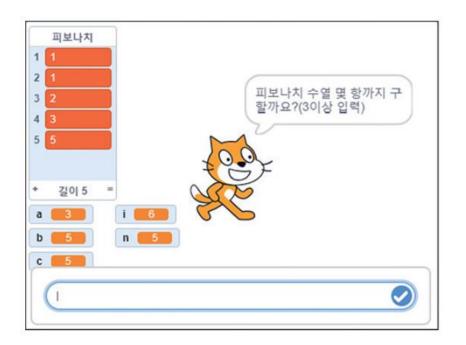


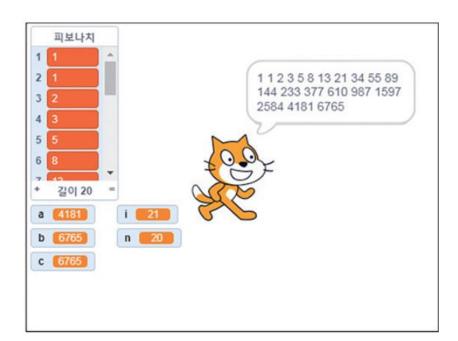




4. 연습 문제

■ n 값을 입력받아 1부터 n항까지 피보나치 수열을 구하는 내용을 구현해 보세요.





스크래치

DAY 10 - 최대공약수

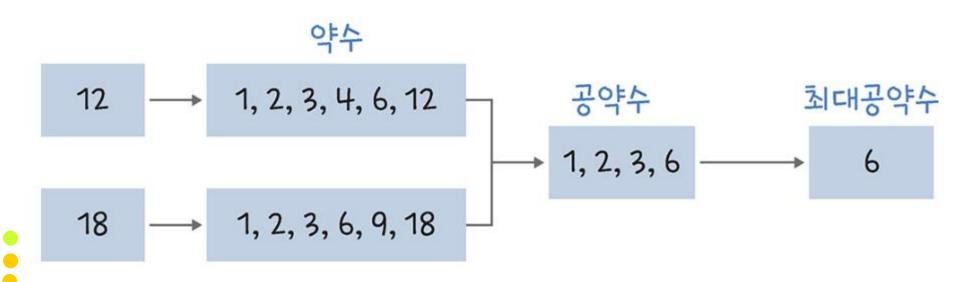
```
#include <iostream>
using namespace std;
using namespace std;
int main() {
    cout<<"Hola Facebook\n";
    return 0;
    return 0;</pre>
```

1. 최대공약수란?





공약수 중에서 가장 큰 수



2. 소수의 곱셈으로 나타내기



$$12 = 1 \times 12$$

$$2 \times 6$$

$$12 = 1 \times 2 \times 2 \times 3$$

$$2 \times 3$$



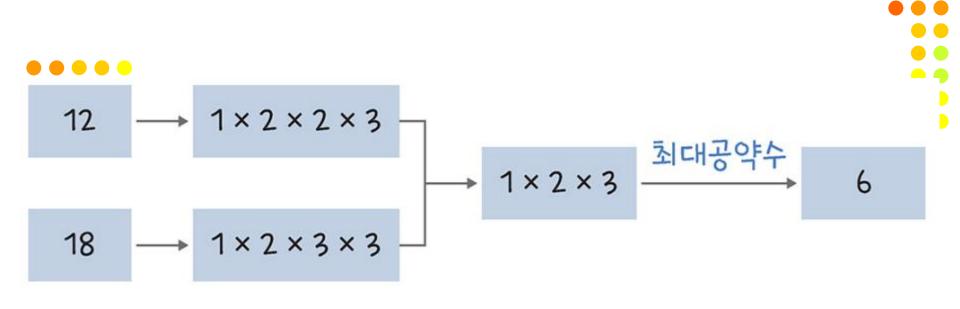
$$18 = 1 \times 18$$

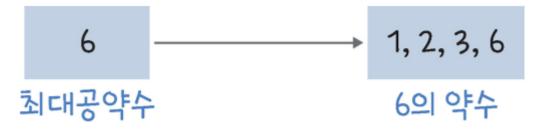
$$2 \times 9$$

$$18 = 1 \times 2 \times 3 \times 3$$

$$3 \times 3$$

• • • •





3. 최대공약수로 나누기



2)12 18 → 12와 18의 공약수 중 1 이외의 공약수는 2, 3, 6입니다.

3) 6 9 → 6과 9의 공약수 중 1 이외의 공약수는 3입니다.

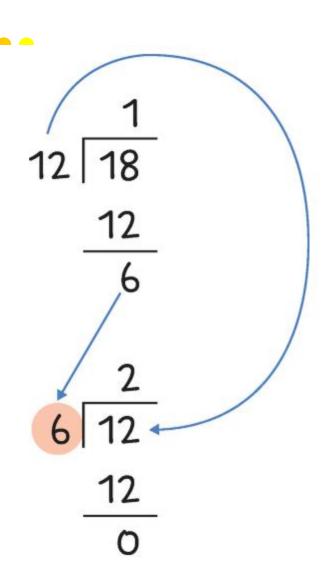
2 3 → 2와 3은 1이외의 공약수가 없습니다.

12와 18의 최대공약수



6)	18	6
	3	1

 $\bullet \bullet \bullet \bullet \bullet$

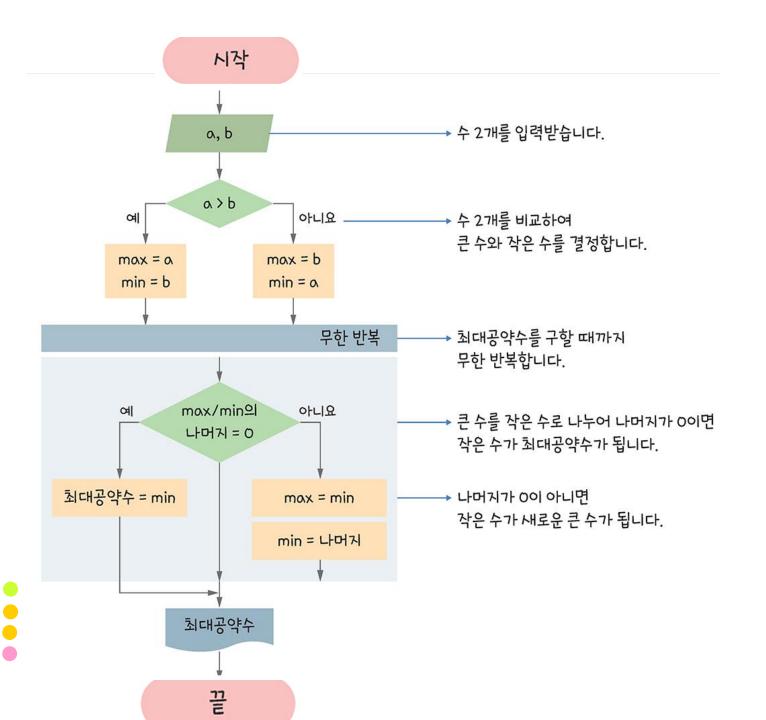


4. 알고리즘



앞서 나열한 내용을 정리하면 다음과 같습니다.

- 1 수 2개를 입력받습니다.
- 2 수 2개 중에서 큰 수와 작은 수를 결정합니다.
- 3] 큰 수 나누기 작은 수를 하여 나머지를 구합니다.
- 4 나머지가 0이면 작은 수가 최대공약수가 되고, 프로그램이 종료됩니다.
- 5 나머지가 0이 아니면 작은 수가 새로운 큰 수가 되고, 나머지는 새로운 작은 수가 되어 프로그램을 반복합니다.

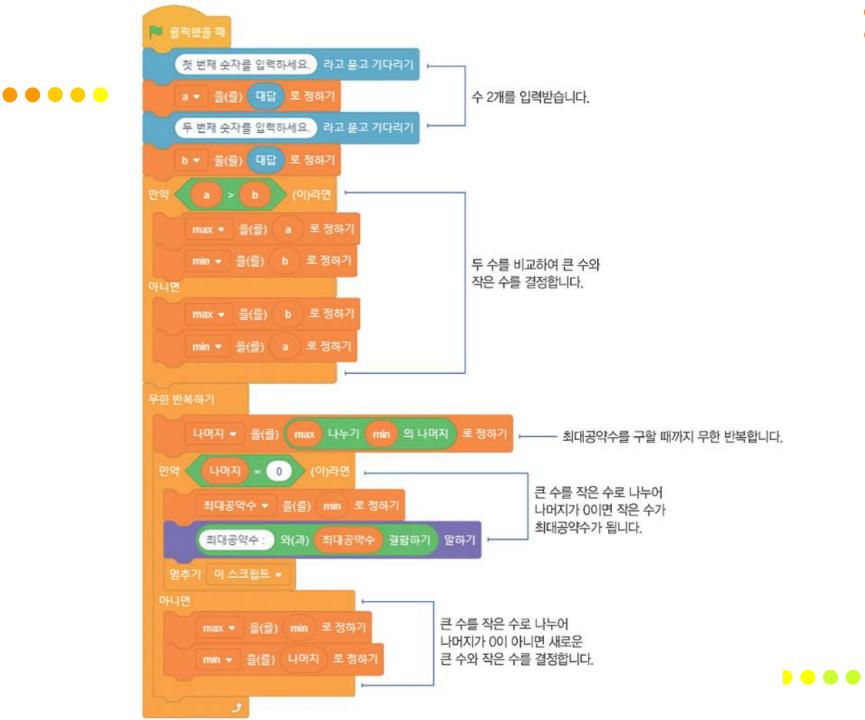


5. 스크래치에서 최대공약수 구하기

숫자 2개를 입력받아 최대공약수를 구하는 내용을 스크래치로 구현해 보겠습니다







6. 연습 문제

두 수의 최대공약수를 구한 결과 최대공약수가 1이면 특별히 '서로소 '라고 합니다. 수 2개를 입력받아 최대공약수를 구하고, 최대공약수가 1이라면 '서로소 '라고 표시하는 내용을 스크래치로 구현해 보세요.

〈조건〉 최대공약수를 구하는 부분을 나만의 블록으로 처리하세요.







 $\bullet \bullet \bullet \bullet \bullet$

