

— 문제 : 길이 n 과 n 개의 정수가 주어진다.

이 배열은 두 정수의 약수들이 섞여있는 것이다.

이 배열에서 두 정수 a, b 를 출력하는 문제

— 접근 : 더 큰 수는 찾기 쉽다. 배열에서 가장 큰 값.

그럼 이 값이 a 가 되고 arr에서 a 의 약수들을 제외한 최대 값이 b 가 될것.

↳ 참고 : n 의 약수 구하기.

```
for (int i=1; i<=n; i++) {  
    if (n%i==0) {  
        int j = n/i;  
        v.push_back(i);  
        if (i!=j) v.push_back(j);  
    }  
}
```

↳ 제곱수 n 의 경우 $i==j$ 인 경우 존재.

but, 대회 당시에 순간 약수 구하는 법이

떠오르지 않아서 편법을 썼다.

arr에 있는 값들의 \sqrt{a} 이하 배열을 만들고 $a \% i == 0$ 이면

처음 방문 시 -1로 바꾸고 방문 check를 했다.

(그럼 중복 요소는 하나만 남을 테니까).

↳ 어차피 한 n 의 약수는 중복되지 않기 때문에
 a 의 약수인 i 들에 대해 값을 바꿔도 무방.

— 구현 : 정석대로라면 cnt 배열을 만들고 (arr의 $cnt[i]++$)

$1 \sim a$ 까지 반복문 돌며 a 의 약수에 대해 $cnt[i]--$.

~~$1 \leq i \leq 10000$ 에 대해 $cnt[i]$ 가 양수면~~

~~그 중에 max를 찾는다 → 그게 b 가 됨.~~

그냥 arr 중에서 $\text{cnt}[\text{arr}[i]]$ 로 체크하면 10000까지
안가도 됨. (25n ≤ 128니까)

☆☆
- 또다른 풀이.

- ① a 를 찾는다.
- ② $\text{int mn} = 0$ 으로 두고 arr에서 a 를 나누어 떨어지게
하지 않는 수 중에 최대값을 b 라고 한다.
- ③ b 가 0이라면 (아직 찾지 못했나?), cnt 가 2인 값 중에 찾는다.
 max 를.

증명)

a 약수: / 2 3 5 6 10 15 30.

b 약수: / 3 7 2 / \Rightarrow 이 경우에는 2를 찾을 수 있다.

b' 약수: / 2 5 10. \Rightarrow 이 경우에는 탐색 후 b 는 0인데
모든 약수가 a 의 약수와 중복된다고
가정해도 된다. 10이 이미 30의
약수이기 때문에 다른 수들도 이미
약수. (중복)

* 그래서 두 경우로 나눌 수 있게 된다.

b 의 최대값이 이미 a 의 약수인 경우: 모든 원소가 중복.
 b 의 최대값이 a 의 약수가 아닐 때: a 를 나누지 못하는
것중 최대값.

\Rightarrow ① a 를 나누지 못하는 $\text{arr}[i]$ 중 최대값 찾기

② if, b 가 아직 0이면 2개씩 있는 값들 중 최대값 찾기.

간단한 $cnt == 2$ 인 값 찾기.

```
for (int i = 1; i ≤ n; ++i) {  
    cnt[arr[i]]++;  
    if (cnt[arr[i]] == 2) b = max(b, arr[i]);  
}
```

뭐 그냥 입력 받을 때 $cnt[input]++$ 해도 되고..

* 두 개 함께 합쳐서

if (a_2 으로 만들지 X || 중복) max 찾기.

간단!

why b의 경우 $\left\langle \begin{array}{l} a의 약수 : 무조건 중복 발생. \\ or \\ a의 약수 X \end{array} \right\}$ 이 두 경우
모두 중에
최대값 찾기.