

주어진 입력에서 괄호 {}, ()가 제대로 짝을 이뤘는지 검사하는 프로그램을 만드시오.
예를 들어, {}()는 제대로 된 짝이지만, {()}는 제대로 된 짝이 아니다. 입력은 한 줄의 파이썬 코드일 수도 있고, 괄호만 주어질 수도 있다.
정상적으로 짝을 이룬 경우 1, 그렇지 않으면 0을 출력한다.
print('{}')같은 경우는 입력으로 주어지지 않으므로 고려하지 않아도 된다.

ex) `print('#{ } {}'.format(tc, find()))`

ex) { () }

반복문

조건 i) (를 만나면 따로 저장! &) 를 만나면 제거

조건 ii) { 를 만나면 따로 저장! & } 를 만나면 제거

조건 iii) 추가 및 제거가 끝나면 리스트에 괄호가 남아있으면 안된다

```
for word in 문자열:
    if word == ( —————> 리스트에 추가!
    if word == ) —————> 리스트의 가장 마지막 원소가 '(' 여야 한다.
    if word == { —————> 리스트에 추가!
    if word == } —————> 리스트의 가장 마지막 원소가 '{' 여야 한다.

반복이 끝난 후, 리스트에 괄호가 남았는 지 확인
```

[SWEA] 4873

문자열 s(길이 1000이내)에서 반복된 문자를 지우려고 한다. 지워진 부분은 다시 앞 뒤를 연결하는데, 만약 연결에 의해 또 반복문자가 생기면 이 부분을 다시 지운다.

반복문자를 지운 후 남은 문자열의 길이를 출력 하시오. 남은 문자열이 없으면 0을 출력한다.

ex) S = C A A A B B A

제거

C A B B A

제거

C A A

제거

C

문자열 반복 —————> 새로운 리스트에 차례로 추가

조건 i) 리스트 [-1] == 현재 추가할 원소 => 제거!

list = []

for word in S:

조건) list [-1] == word —> list [-1] 제거

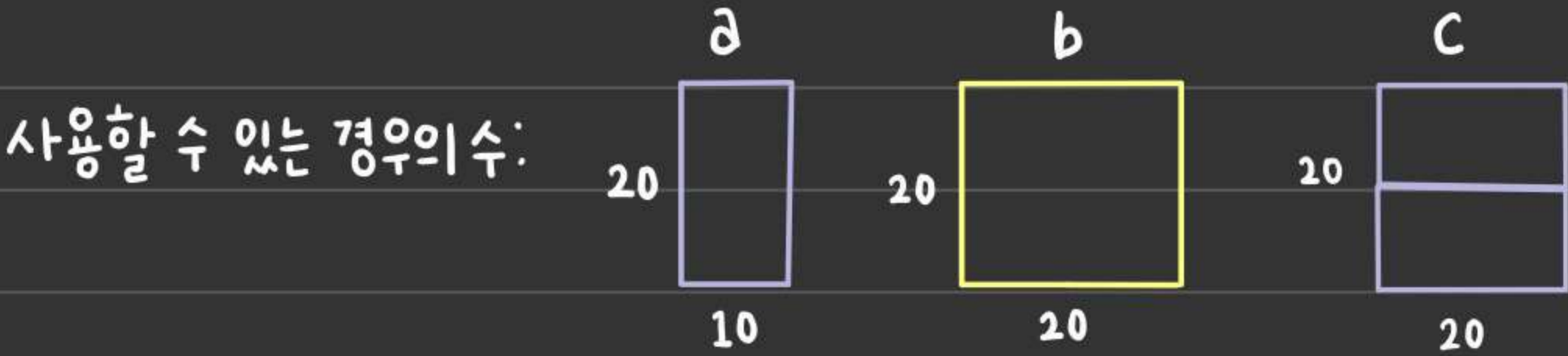
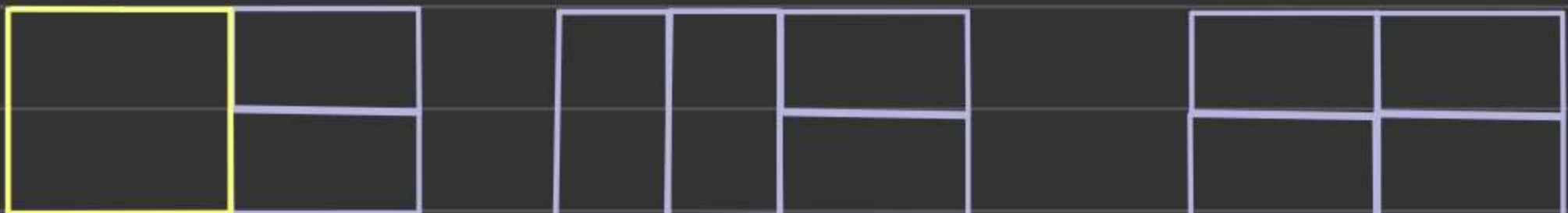
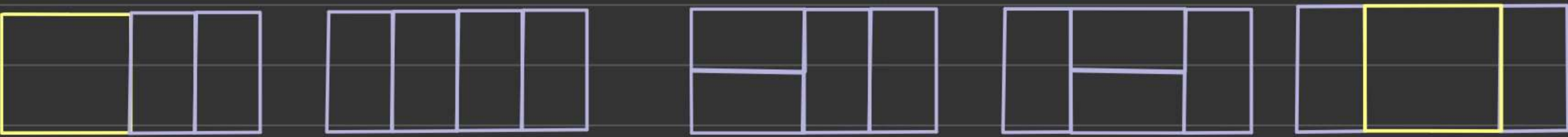
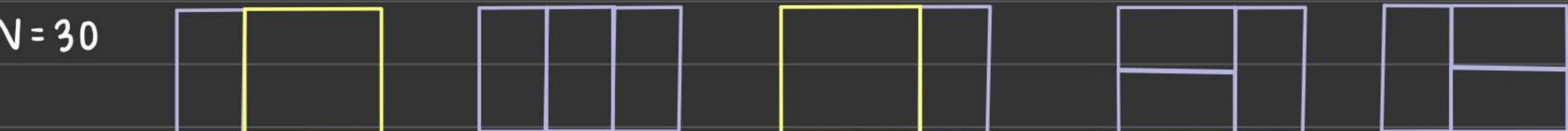
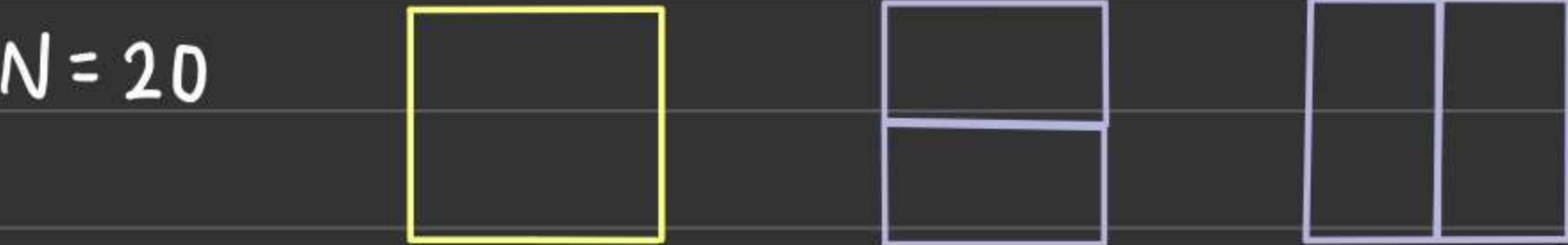
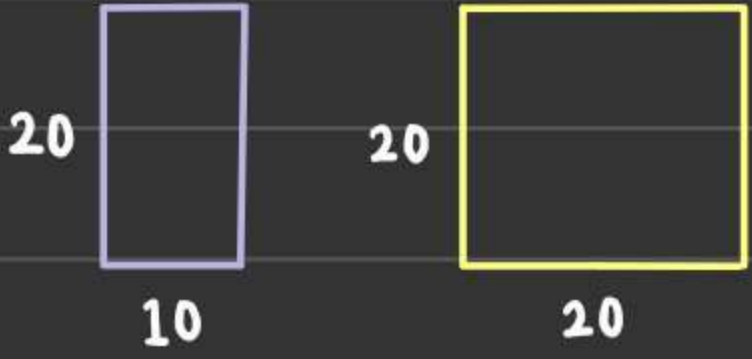
그외) list에 word 추가!

어린이 알고리즘 교실의 선생님은 경우의 수 놀이를 위해, 가로x세로 길이가 10x20, 20x20인 직사각형 종이를 잔뜩 준비했다.

그리고 교실 바닥에 20xN 크기의 직사각형을 테이프로 표시하고, 이 안에 준비한 종이를 빈틈없이 붙이는 방법을 찾아보려고 한다.

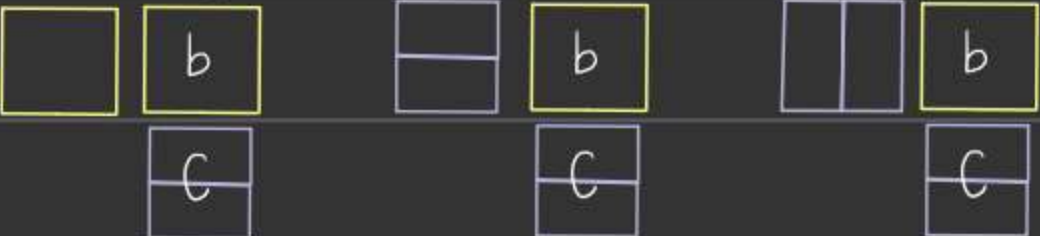
10의 배수인 N이 주어졌을 때, 종이를 붙이는 모든 경우를 찾으려면 테이프로 만든 표시한 영역을 몇 개나 만들어야 되는지 계산하는 프로그램을 만드시오. 직사각형 종이가 모자라는 경우는 없다.

예시)



N = 40은 N = 30에 비해 10 증가 → N = 30의 각 경우의 수에 a를 붙인다. ∴ 5 × 1
N = 20에 비해 20 증가 → N = 20의 각 경우의 수에 b, c를 붙인다. ∴ 3 × 2

이 둘의 합이 N = 40의 경우의 수



```
10 ≤ N ≤ 300 N은 10의 배수: 모든 경우의 수를 만든다면...
case = [0, 1, 3]
for i in Range(3, 31)
    case[-1] + case[-2] * 2 를 case에 추가
```

