

1. [3+3=6점] 거리두기?

문자열 xs를 인수로 받아서 인접한 문자가 같은 경우가 하나도 없으면 True를, 하나라도 있으면 False를 리턴하는 함수 distanced를 재귀 함수로 작성하자.

- (1) 아래 뼈대 코드 형식에 맞추어 if-else 제어구조를 활용하여 먼저 완성하고,
- (2) 이어서 논리식만으로 다시 작성하자.

실행 사례

```
distanced("") => True
distanced("0") => True
distanced("00") => False
distanced("0122345") => False
distanced("0123456") => True
```

2. [6점] 만 나이 계산 함수

생년월일을 인수로 년, 월, 일 각각 자연수로 받아 만으로 몇 살인지 알려주는 함수 how_old를 작성하자. 태어나는 날부터 만 0살이고, 1년 후 돌이 되는 날 만 1살이 된다. 예를 들어, 2002년 9월 15일 생은 2022년 9월 15일에 만 20살이 되고, 2021년 4월에는 만 18살이다. 파이썬 표준 라이브러리의 datetime 모듈의 date 객체를 활용하면 오늘 날짜는 다음과 같이 년, 월, 일 별도로 얻을 수 있다.

```
>>> from datetime import date
>>> today = date.today()
>>> today.year
2021
>>> today.month
4
>>> today.day
23
>>>
```

인수는 모두 실제로 존재하는 년, 월, 일로만 주어진다 고 가정한다.

실행 사례

```
how_old(2021,4,24) => -1
how_old(2021,4,23) => 0
how_old(2002,4,22) => 19
how_old(2002,4,23) => 19
how_old(2002,4,24) => 18
how_old(2002,3,22) => 19
how_old(2002,3,24) => 19
how_old(2002,5,22) => 18
how_old(2002,5,24) => 18
```

3. [3+2+2=7점] 자연수 짝수열 n개의 누적 합

자연수 짝수를 증가하는 순으로 누적하여 합하면 다음과 같다.

$$\begin{aligned}
 &2 = 2 \\
 &4 + 2 = 6 \\
 &6 + 4 + 2 = 12 \\
 &8 + 6 + 4 + 2 = 20 \\
 &10 + 8 + 6 + 4 + 2 = 30 \\
 &\dots
 \end{aligned}$$

$$2 \times n + \dots + 10 + 8 + 6 + 4 + 2 = \text{sigma_even}(n)$$

이와 같이 자연수 짝수열 n 개의 누적 합을 계산하는 (1) 재귀 함수 sigma_even을 구현하고, 이어서 (2) 꼬리재귀 함수, (3) while 루프 함수로 변환하자.

실행 사례

```
sigma_even(0) => 0
sigma_even(1) => 2
sigma_even(2) => 6
sigma_even(3) => 12
sigma_even(4) => 20
sigma_even(5) => 30
sigma_even(6) => 42
sigma_even(7) => 56
```

4. [3+2+2+2=9점] 차집합 함수

- (1) 리스트로 표현된 두 집합을 인수로 받아서, 차집합을 리턴해주는 재귀 함수 difference를 작성하자. 인수 리스트에는 중복된 원소가 없다고 가정한다. 그리고 리턴하는 리스트에도 중복된 원소가 없어야 한다.
- (2) 작성한 재귀 함수를 꼬리재귀 함수로 변환하자.
- (3) 작성한 꼬리재귀 함수를 while 루프로 변환하자.
- (4) while 루프 대신, for 루프를 사용하여 함수를 재작성 하자.

실행 사례

```
difference([],[]) => []
difference([1,2],[1]) => [1, 2]
difference([],[3,4]) => []
difference([1,2],[3,4]) => [1, 2]
difference([1,2],[2,3]) => [1]
difference([1,2],[2,1]) => []
difference([1,2,3],[3,2,1]) => []
difference([1,2,3],[3,2,4]) => [1]
difference([1,2,3],[4,5,6]) => [1, 2, 3]
```

5. [3+2+2=7점] 리스트의 선두에서 연속 동일 원소 모두 리스트로 모으기

(1) 리스트 xs를 인수로 받아서 선두 부분에서 같은 원소로 이어지는 부분만 리스트로 리턴하는 함수 head_equiv를 재귀로 작성하자.

재귀 알고리즘은 다음과 같다.

- 원소가 하나만 있는 경우에는 그대로 리턴한다.
- 원소가 둘 이상 있는 경우, 선두 원소와 바로 다음 원소를 비교하여
- 다르면 선두 원소만으로 구성된 리스트를 리턴한다.
- 같으면 선두원소를 후미리스트로 재귀 호출한 결과와 이어붙여서 리턴한다.

- (2) 작성한 재귀 함수를 꼬리재귀 함수로 재작성하자.
- (3) 작성한 꼬리재귀 함수를 while 루프를 이용한 함수로 재작성하자.

실행 사례

```
head_equiv([]) => []
head_equiv([2]) => [2]
head_equiv([2,3]) => [2]
head_equiv([2,2]) => [2,2]
head_equiv([2,2,2]) => [2,2,2]
head_equiv([2,2,2,3,3]) => [2,2,2]
```