

1. [3+3=6점] 홀로?

문자열 xs를 인수로 받아서 xs에 두 번이상 나타나는 문자가 하나라도 있으면 False를, 그렇지 않으면 True를 리턴하는 함수 solitary를 재귀 함수로 작성하자.

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

실행 사례

```
solitary("") => True
solitary("0") => True
solitary("0120") => False
solitary("012345") => True
```

2. [6점] 성인 인증 함수

생년월일을 인수로 년, 월, 일 각각 자연수로 받아서 성년인지 확인하는 함수 check_adult를 작성하자. 만 19살이 되는 생일부터 성년으로 인정한다. 즉, 만 19살 생일이거나 지났으면 True를 리턴하고, 그렇지 않으면 False를 리턴한다. 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
21
>>>
```

인수는 모두 실제로 존재하는 년, 월, 일로만 주어진다고 가정한다.

실행 사례

```
check_adult(2001,12,31) => True
check_adult(2002,3,1) => True
check_adult(2002,4,20) => True
check_adult(2002,4,21) => True
check_adult(2002,4,22) => False
check_adult(2002,5,1) => False
```

3. [3+2+2=7점] 2의 n 승의 누적 합

자연수 n 까지의 모든 자연수에 대하여 2의 n 승을 누적하여 합하면 다음과 같다.

$$\begin{aligned}
 &1 = 1 \\
 &2 + 1 = 3 \\
 &4 + 2 + 1 = 7 \\
 &8 + 4 + 2 + 1 = 15 \\
 &16 + 8 + 4 + 2 + 1 = 31 \\
 &\dots
 \end{aligned}$$

$$2^n + \dots + 16 + 8 + 4 + 2 + 1 = \text{sigma_power_of_2}(n)$$

이와 같이 자연수 2의 n 승의 누적 합을 계산하는 (1) 재귀 함수 sigma_power_of_2를 구현하고, 이어서 (2) 꼬리재귀 함수, (3) while 루프 함수로 변환하자.

실행 사례

```
sigma_power_of_2(0) => 1
sigma_power_of_2(1) => 3
sigma_power_of_2(2) => 7
sigma_power_of_2(3) => 15
sigma_power_of_2(4) => 31
sigma_power_of_2(5) => 63
sigma_power_of_2(6) => 127
sigma_power_of_2(7) => 255
```

4. [3+2+2+2=9점] 합집합 함수

- (1) 리스트로 표현된 두 집합을 인수로 받아서, 합집합을 리턴 해주는 재귀 함수 union을 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 재작성하자.

실행 사례

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

5. [3+2+2=7점] 리스트에서 중복 원소 모두 없애기

- (1) 리스트 xs를 인수로 받아서 중복된 원소를 모두 리스트에서 제거한 리스트를 리턴하는 함수 remove_duplicates를 재귀로 작성하자.

재귀 알고리즘은 다음과 같다.

- 선두원소가 후미리스트에 있으면 선두원소를 무시하고 후미리스트로 재귀 호출한 결과를 리턴한다.
- 선두원소가 후미리스트에 없으면 선두원소를 후미리스트로 재귀 호출한 결과와 이어붙여서 리턴한다.
- 빈 리스트이면 그대로 리턴한다.

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

실행 사례

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