

문제1.

어떤 수(number)의 각 자리 숫자(digit)의 합을 계산하는 `sumOfDigits()` 라는 함수를 작성하자. 입력한 수를 읽어 `sumOfDigits()` 함수를 호출하며, 이 함수는 합산할 숫자가 남지 않을 때까지 자신을 호출해, 최종적인 합을 사용자에게 표시한다.

힌트: 함수의 입력을 문자열로 바꿔서 `for` 문과 `int` 함수를 사용

출력 예시:

```
>>> sumOfDigits(47253)
21
>>> sumOfDigits(53472)
21
>>> sumOfDigits(643)
13
```

문제2.

알파벳 소문자로만 이루어진 단어 `s`가 주어진다. 각각의 알파벳에 대해서, 단어에 포함되어 있는 경우에는 처음 등장하는 위치를, 포함되어 있지 않은 경우에는 -1을 출력하는 함수 `alphabet_identifier` 작성하시오

입력: 단어 `s`가 주어진다. 단어의 길이는 100을 넘지 않으며, 알파벳 소문자로만 이루어져 있다. (그렇지 않은 경우, 에러 메시지 'invalid word'를 출력)

출력: 각각의 알파벳에 대해서, `a`가 처음 등장하는 위치, `b`가 처음 등장하는 위치, ... `z`가 처음 등장하는 위치를 공백으로 구분해서 출력한다. 만약, 어떤 알파벳이 단어에 포함되어 있지 않다면 -1을 출력한다. 단어의 첫 번째 글자는 0번째 위치이고, 두 번째 글자는 1번째 위치이다.

예시

```
>>> s = 'python'
>>> alphabet_identifier(s)
a -1
b -1
c -1
d -1
e -1
f -1
g -1
h 3
i -1
j -1
k -1
l -1
m -1
n 5
o 4
p 0
q -1
r -1
```

```
s -1
t 2
u -1
v -1
w -1
x -1
y 1
z -1
```

문제3.

상근이의 동생 상수는 수학을 정말 못한다. 상수는 숫자를 읽는데 문제가 있다. 이렇게 수학을 못하는 상수를 위해서 상근이는 수의 크기를 비교하는 문제를 내주었다. 상근이는 세 자리 수 두 개를 칠판에 써주었다. 그 다음에 크기가 큰 수를 말해보라고 했다.

상수는 수를 다른 사람과 다르게 거꾸로 읽는다. 예를 들어, 734와 893을 칠판에 적었다면, 상수는 이 수를 437과 398로 읽는다. 따라서, 상수는 두 수중 큰 수인 437을 큰 수라고 말할 것이다.

두 수가 주어졌을 때, 상수의 대답을 출력하는 함수 `sangsu`를 작성하시오.

예시

```
>>> print(sangsu(734, 893))
437
>>> print(sangsu(123, 941))
321
```