

비교 연산자

```
In [1]: True
```

```
Out[1]: True
```

```
In [2]: False
```

```
Out[2]: False
```

```
In [3]: 3 > 1
```

```
Out[3]: True
```

```
In [4]: 3 < 1
```

```
Out[4]: False
```

```
In [5]: 10 == 10
```

```
Out[5]: True
```

```
In [6]: 10 != 5
```

```
Out[6]: True
```

```
In [7]: 'Python' == 'Python'
```

```
Out[7]: True
```

```
In [8]: 'Python' == 'python'
```

```
Out[8]: False
```

```
In [9]: 10 <= 10
```

```
Out[9]: True
```

```
In [10]: 1 == 1.0
```

```
Out[10]: True
```

```
In [11]: 1 is 1.0
```

```
<>:1: SyntaxWarning: "is" with a literal. Did you mean "=="?
```

```
<>:1: SyntaxWarning: "is" with a literal. Did you mean "=="?
```

```
<ipython-input-11-2ebf4c947f31>:1: SyntaxWarning: "is" with a literal. Did you mean "="
```

```
="?  
1 is 1.0
```

Out[11]: False

```
In [12]: 1 is not 1.0
```

```
<>:1: SyntaxWarning: "is not" with a literal. Did you mean "!="?  
<>:1: SyntaxWarning: "is not" with a literal. Did you mean "!="?  
<ipython-input-12-7a2cea8bf2ec>:1: SyntaxWarning: "is not" with a literal. Did you mea  
n "!="?  
1 is not 1.0
```

Out[12]: True

논리 연산자

```
In [13]: True and True
```

Out[13]: True

```
In [14]: True and False
```

Out[14]: False

```
In [15]: False and False
```

Out[15]: False

```
In [16]: True or True
```

Out[16]: True

```
In [17]: True or False
```

Out[17]: True

```
In [18]: False or False
```

Out[18]: False

```
In [19]: not True
```

Out[19]: False

```
In [20]: not False
```

Out[20]: True

```
In [21]: # * not, and, or 순서로 판단! *  
not True and False or not False
```

```
# False and False or True
# False or True
# True
```

Out[21]: True

```
In [22]: # 헛갈리면 괄호 사용!
          ((not True) and False) or (not False)
```

Out[22]: True

```
In [23]: 10 == 10 and 10 != 5
```

Out[23]: True

```
In [24]: 10 > 5 or 10 < 3
```

Out[24]: True

```
In [25]: not 10 > 5
```

Out[25]: False

```
In [26]: not 1 is 1.0
```

```
<>:1: SyntaxWarning: "is not" with a literal. Did you mean "!="?
<>:1: SyntaxWarning: "is not" with a literal. Did you mean "!="?
<ipython-input-26-b4bf5bb1d071>:1: SyntaxWarning: "is not" with a literal. Did you mea
n "!="?
    not 1 is 1.0
```

Out[26]: True

```
In [27]: # 한 과목이라도 50점 미만이면 불합격.
```

```
In [28]: korean = 92
          english = 47
          mathematics = 86
          science = 81

          korean >= 50 and english >= 50 and mathematics >= 50 and science >= 50
```

Out[28]: False

input 을 이용해 국어 90 이상, 영어 80초과, 수학 85점 초과, 과학 80 이상 합격

```
In [29]: korean, english, math, science = map(int, input().split())

          90 81 86 80
```

```
In [30]: korean >= 90 and english > 80 and math >85 and science >= 80
```

Out[30]: True

```
In [31]: korean, english, math, science = map(int, input().split())  
  
90 80 85 80
```

```
In [32]: korean >= 90 and english > 80 and math >85 and science >= 80
```

Out[32]: False

문자열 사용

```
In [33]: hello = "Hello, world!"  
hello
```

Out[33]: 'Hello, world!'

```
In [34]: hello = '''Hello, Python!'''  
hello
```

Out[34]: 'Hello, Python!'

```
In [35]: python = """Python Programming"""  
python
```

Out[35]: 'Python Programming'

```
In [36]: hello = '''Hello, world!  
안녕하세요.  
Python입니다.'''
```

```
In [37]: hello
```

Out[37]: 'Hello, world!\n안녕하세요.\nPython입니다.'

```
In [38]: print(hello)
```

```
Hello, world!  
안녕하세요.  
Python입니다.
```

```
In [40]: s = "Python isn't difficult"  
s
```

Out[40]: "Python isn't difficult"

```
In [42]: s = 'He said "Python is easy"'  
s
```

Out[42]: 'He said "Python is easy"'

```
In [43]: # 작은 따옴표 안에 작은 따옴표, 큰따옴표 안에 큰 따옴표는 못 넣는다.
```

```
In [44]: s = 'Python isn't difficult'
```

```
File "<ipython-input-44-ee68b4c434be>", line 1
    s = 'Python isn't difficult'
          ^
SyntaxError: invalid syntax
```

```
In [45]: s = "He said "Python is easy""
```

```
File "<ipython-input-45-bcbcd830d9be>", line 1
    s = "He said "Python is easy""
              ^
SyntaxError: invalid syntax
```

```
In [46]: single_quote = '''안녕하세요."
'파이썬'입니다.'''
```

```
In [47]: double_quote1 = """Hello"
'Python'"""
```

```
In [48]: double_quote2 = """Hello, 'Python'"""
```

```
In [49]: print(single_quote)
print(double_quote1)
print(double_quote2)
```

```
"안녕하세요."
'파이썬'입니다.
"Hello"
'Python'
Hello, 'Python'
```

```
In [52]: ment = '''Python is a programming language that lets you work quickly
and
integrate systems more effectively.'''
print(ment)
```

```
Python is a programming language that lets you work quickly
and
integrate systems more effectively.
```

'Python' is a "programming language"

that lets you work quickly

and

integrate systems more effectively.

로 출력하기

```
In [55]: s = """'Python' is a "programming language"
that lets you work quickly
and
```

```
integrate systems more effectively."""  
print(s)
```

```
'Python' is a "programming language"  
that lets you work quickly  
and  
integrate systems more effectively.
```

list 만들기

```
In [57]: a = [38, 21, 53, 62, 19]  
a  
# list에 저장된 각 값을 '요소'라고 함.
```

```
Out[57]: [38, 21, 53, 62, 19]
```

```
In [59]: person = ['sungwon', 24, 177.2, True]  
person
```

```
Out[59]: ['sungwon', 24, 177.2, True]
```

```
In [60]: # 빈 list 만들기
```

```
In [62]: a = []  
a
```

```
Out[62]: []
```

```
In [63]: b = list()  
b
```

```
Out[63]: []
```

```
In [64]: # use the range for making list
```

```
In [65]: range(10)
```

```
Out[65]: range(0, 10)
```

```
In [66]: # range(0,10) == 0, 1, 2, 3, ..., 9
```

```
In [68]: # 리스트 = list(range(횟수))
```

```
In [67]: a = list(range(10))  
a
```

```
Out[67]: [0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
```

```
In [69]: # 리스트 = list(range(시작, 끝))
```

```
In [70]: b = list(range(5, 12))  
b
```

```
Out[70]: [5, 6, 7, 8, 9, 10, 11]
```

```
In [71]: # 리스트 = list(range(시작, 끝, 증가폭))
```

```
In [73]: c = list(range(-4, 10, 2))  
c
```

```
Out[73]: [-4, -2, 0, 2, 4, 6, 8]
```

```
In [75]: d = list(range(10, 0, -1))  
d
```

```
Out[75]: [10, 9, 8, 7, 6, 5, 4, 3, 2, 1]
```

```
In [76]: e = list(range(10, 0))  
e
```

```
Out[76]: []
```

튜플 = 저장된 요소를 변경, 추가, 삭제 못 함. '읽기 전용 리스트' 라고 이해.

```
In [78]: a = (38, 21, 53, 62, 19)  
a
```

```
Out[78]: (38, 21, 53, 62, 19)
```

```
In [80]: # 괄호 없이도 가능.
```

```
In [79]: a = 38, 21, 53, 62, 19  
a
```

```
Out[79]: (38, 21, 53, 62, 19)
```

```
In [81]: # list와 마찬가지로 자료형대를 섞어서 저장 가능
```

```
In [82]: person = ('sungwon', 24, 177.2, True)  
person
```

```
Out[82]: ('sungwon', 24, 177.2, True)
```

```
In [83]: # * 요소가 한 개 일 때, 튜플은 어떻게 만드는가? *
```

```
In [85]: (38) # 그냥 값 출력
```

Out[85]: 38

In [87]: `(38,) # 튜플로 만들기`

Out[87]: (38,)

실무에서 가끔 튜플 형태를 이용해야하는 경우가 있기에 한 개짜리 튜플 필요

In [89]: `a = tuple(range(10))`
a

Out[89]: (0, 1, 2, 3, 4, 5, 6, 7, 8, 9)

In [90]: `b = tuple(range(5, 12))`
b

Out[90]: (5, 6, 7, 8, 9, 10, 11)

In [92]: `c = tuple(range(-4, 10, 2))`
c

Out[92]: (-4, -2, 0, 2, 4, 6, 8)

튜플과 리스트는 기능과 형태는 같음.

그러므로, 튜플을 리스트로 or 리스트를 튜플로 변환 가능

In [94]: `a = [1, 2, 3]`
`tuple(a)`

Out[94]: (1, 2, 3)

In [95]: `b = (4, 5, 6)`
`list(b)`

Out[95]: [4, 5, 6]

range를 이용하여 리스트 [5, 3, 1, -1, -3, -5, -7, -9]가 출력되게 만들기.

In [98]: `a = list(range(5, -10, -2))`
`print(a)`

[5, 3, 1, -1, -3, -5, -7, -9]

input, range 를 이용하여 튜플 출력 프로그램 작성.

In [100]: `tuple(range(-10, 10, int(input())))`
2

Out[100]: (-10, -8, -6, -4, -2, 0, 2, 4, 6, 8)

Sequence 자료형 = list, tuple, range, str

In [107]: `# 리스트`


```
a = [0, 10, 20, 30, 40, 50, 60, 70, 80, 90]
```

```
In [108]: 30 in a
```

```
Out[108]: True
```

```
In [109]: 100 in a
```

```
Out[109]: False
```

```
In [110]: 100 not in a
```

```
Out[110]: True
```

```
In [111]: 30 not in a
```

```
Out[111]: False
```

```
In [112]: # 튜플  
43 in (38, 76, 43, 62, 19)
```

```
Out[112]: True
```

```
In [114]: # range  
1 in range(10)
```

```
Out[114]: True
```

```
In [115]: # 문자열 str  
'P' in 'Hello, Python'
```

```
Out[115]: True
```

```
In [118]: # list 연결  
a = [0, 10, 20, 30]  
b = [9, 8, 7, 6]  
a + b
```

```
Out[118]: [0, 10, 20, 30, 9, 8, 7, 6]
```

```
In [119]: # tuple 연결  
a = 0, 10, 20, 30  
b = 9, 8, 7, 6  
a + b
```

```
Out[119]: (0, 10, 20, 30, 9, 8, 7, 6)
```

```
In [122]: # range 는 연결이 안 되므로, tuple 이나 list로 변환 후 연결  
range(0, 10) + range(10, 20)
```

```

-----
TypeError                                Traceback (most recent call last)
<ipython-input-122-ef3ad1266fb0> in <module>
      1 # range 는 연결이 안 되므로, tuple 이나 list로 변환 후 연결
----> 2 range(0, 10) + range(10, 20)

TypeError: unsupported operand type(s) for +: 'range' and 'range'

```

```
In [123]: list(range(0, 10)) + list(range(10, 20))
```

```
Out[123]: [0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19]
```

```
In [124]: tuple(range(0, 10)) + tuple(range(10, 20))
```

```
Out[124]: (0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19)
```

```
In [126]: # str 문자열은 바로 된다.
          'Hello, ' + 'world!'
```

```
Out[126]: 'Hello, world!'
```

```
In [128]: # 문자열과 숫자는 어떻게?
          'Hello, ' + str(10+54)
```

```
Out[128]: 'Hello, 64'
```

```
In [129]: [0, 10, 20, 30] * 3
```

```
Out[129]: [0, 10, 20, 30, 0, 10, 20, 30, 0, 10, 20, 30]
```

```
In [130]: (0, 10, 20, 30) * 3
```

```
Out[130]: (0, 10, 20, 30, 0, 10, 20, 30, 0, 10, 20, 30)
```

```
In [131]: # range 는 역시 리스트나 튜플로 변환해서 반복
          list(range(0, 5, 2)) * 3
```

```
Out[131]: [0, 2, 4, 0, 2, 4, 0, 2, 4]
```

```
In [132]: tuple(range(0, 5, 2)) * 3
```

```
Out[132]: (0, 2, 4, 0, 2, 4, 0, 2, 4)
```

```
In [133]: 'Hello, ' * 3
```

```
Out[133]: 'Hello, Hello, Hello, '
```

요소 개수

```
In [134]: a = [0, 10, 20, 30, 40, 50, 60, 70, 80, 90]
          len(a)
```

Out[134]: 10

```
In [135]: b = (38, 76, 43, 62, 19)
          len(b)
```

Out[135]: 5

```
In [137]: len(range(0, 10, 2))
```

Out[137]: 5

```
In [138]: hello = 'Hello, world!'
          len(hello)
```

Out[138]: 13

```
In [139]: hello = '안녕하세요'
          len(hello)
```

Out[139]: 5

인덱스

```
In [140]: a = [38, 21, 53, 62, 19]
```

```
In [141]: a[0]
```

Out[141]: 38

```
In [142]: a[2]
```

Out[142]: 53

```
In [143]: a[4]
```

Out[143]: 19

```
In [144]: b = (38, 21, 53, 62, 19)
```

```
In [145]: b[0]
```

Out[145]: 38

```
In [146]: r = range(0, 10, 2)
```

```
In [147]: r[2]
```

Out[147]: 4

```
In [148]: hello = 'Hello, world!'
```

```
In [149]: hello[7]
```

```
Out[149]: 'w'
```

```
In [150]: a = [38, 21, 53, 62, 19]
          a[-1]
```

```
Out[150]: 19
```

```
In [151]: a[-5]
```

```
Out[151]: 38
```

```
In [152]: b = (38, 21, 53, 62, 19)
          b[-1]
```

```
Out[152]: 19
```

```
In [153]: r = range(0, 10, 2)
          r[-3]
```

```
Out[153]: 4
```

```
In [154]: hello = 'Hello, world!'
          hello[-4]
```

```
Out[154]: 'r'
```

```
In [156]: a = [38, 21, 53, 62, 19]
          a[len(a)]
```

```
-----
IndexError                                Traceback (most recent call last)
<ipython-input-156-82080ca663b9> in <module>
      1 a = [38, 21, 53, 62, 19]
----> 2 a[len(a)]
```

```
IndexError: list index out of range
```

```
In [157]: a[len(a) - 1]
```

```
Out[157]: 19
```

```
In [164]: # tuple 은 변경, 추가, 삭제 안 된다고 함.
```

```
In [159]: a = [0, 0, 0, 0, 0]
          a
```

Out[159]: [0, 0, 0, 0, 0]

```
In [162]:
a[0] = 38
a[1] = 21
a[2] = 53
a[3] = 62
a[4] = 19
a
```

Out[162]: [38, 21, 53, 62, 19]

```
In [163]:
a = [38, 21, 53, 62, 19]
del a[2]
a
```

Out[163]: [38, 21, 62, 19]

시퀀스객체 [시작인덱스 : 끝인덱스]

끝 인덱스는 가져오려는 범위에 포함되지 않음. 따라서, 끝 인덱스는 실제로 가져오려는 인덱스보다 1을 더 크게 지정해야 함.

```
In [166]:
a = [0, 10, 20, 30, 40, 50, 60, 70, 80, 90]
a[0:4]      # 인덱스 0부터 3까지 잘라서 새 리스트를 만들기
```

Out[166]: [0, 10, 20, 30]

```
In [167]:
a = [0, 10, 20, 30, 40, 50, 60, 70, 80, 90]
a[4:7]      # 인덱스 4부터 6까지 요소 3개를 가져옴
```

Out[167]: [40, 50, 60]

```
In [168]:
a[4:-1]     # 인덱스 4부터 -2까지 요소 5개를 가져옴
```

Out[168]: [40, 50, 60, 70, 80]

```
In [169]:
# 시퀀스객체 [시작인덱스 : 끝인덱스 : 인덱스증가폭]
```

```
In [173]:
a = [0, 10, 20, 30, 40, 50, 60, 70, 80, 90]
a[2:8:3]    # 인덱스 2부터 3씩 증가시키면서 인덱스 7까지 가져옴
```

Out[173]: [20, 50]

```
In [179]:
a[2:9:3]    # 인덱스 2부터 3씩 증가시키면서 인덱스 8까지 가져옴
```

Out[179]: [20, 50, 80]

```
In [180]:
a = [0, 10, 20, 30, 40, 50, 60, 70, 80, 90]
a[:7]      # 리스트 처음부터 인덱스 6까지 가져옴
```

Out[180]: [0, 10, 20, 30, 40, 50, 60]

```
In [181]: a[7:] # 인덱스 7부터 마지막 요소까지 가져옴
```

```
Out[181]: [70, 80, 90]
```

```
In [182]: a[:] # 리스트 전체를 가져옴
```

```
Out[182]: [0, 10, 20, 30, 40, 50, 60, 70, 80, 90]
```

```
In [183]: a = [0, 10, 20, 30, 40, 50, 60, 70, 80, 90]
a[7:2] # 리스트의 처음부터 인덱스를 2씩 증가시키면서 인덱스 6까지 가져옴
```

```
Out[183]: [0, 20, 40, 60]
```

```
In [184]: a[7::2]
```

```
Out[184]: [70, 90]
```

```
In [185]: a[::2] # 리스트 전체에서 인덱스 0부터 2씩 증가시키면서 요소를 가져옴
```

```
Out[185]: [0, 20, 40, 60, 80]
```

```
In [186]: a[::] # 리스트 전체를 가져옴
```

```
Out[186]: [0, 10, 20, 30, 40, 50, 60, 70, 80, 90]
```

```
In [187]: a = [0, 10, 20, 30, 40, 50, 60, 70, 80, 90]
a[0:len(a)] # 시작 인덱스에 0, 끝 인덱스에 len(a) 지정하여 리스트 전체를 가져옴
```

```
Out[187]: [0, 10, 20, 30, 40, 50, 60, 70, 80, 90]
```

```
In [188]: a[:len(a)]
```

```
Out[188]: [0, 10, 20, 30, 40, 50, 60, 70, 80, 90]
```

튜플 [시작인덱스: 끝인덱스]

튜플 [시작인덱스: 끝인덱스: 인덱스증가폭]

```
In [189]: b = (0, 10, 20, 30, 40, 50, 60, 70, 80, 90)
b[4:7] # 인덱스 4부터 6까지 요소 3개를 가져옴
```

```
Out[189]: (40, 50, 60)
```

```
In [190]: b[4:] # 인덱스 4부터 마지막 요소까지 가져옴
```

```
Out[190]: (40, 50, 60, 70, 80, 90)
```

```
In [191]: b[7:2] # 튜플의 처음부터 인덱스를 2씩 증가시키면서 인덱스 6까지 가져옴
```

```
Out[191]: (0, 20, 40, 60)
```

range객체 [시작인덱스 : 끝인덱스]

range객체 [시작인덱스 : 끝인덱스 : 인덱스증가폭]

```
In [192]: r = range(10)
          r
```

```
Out[192]: range(0, 10)
```

```
In [194]: r[4:7]      # 인덱스 4부터 6까지 숫자 3개를 생성하는 range 객체를 생성
```

```
Out[194]: range(4, 7)
```

```
In [195]: r[4:]       # 인덱스 4부터 9까지 숫자 6개를 생성하는 range 객체를 생성
```

```
Out[195]: range(4, 10)
```

```
In [196]: r[:7:2]
          # 인덱스 0부터 2씩 증가시키면서 인덱스 6까지 숫자 4개를 생성하는 range 객체를 생성
```

```
Out[196]: range(0, 7, 2)
```

```
In [197]: list(r[:7:2])
```

```
Out[197]: [0, 2, 4, 6]
```

문자열 [시작인덱스 : 끝인덱스]

문자열 [시작인덱스 : 끝인덱스 : 인덱스증가폭]

```
In [198]: hello = 'Hello, world!'
          hello[2:9]    # 인덱스 2부터 인덱스 8까지 잘라서 문자열을 생성
```

```
Out[198]: 'llo, wo'
```

```
In [199]: hello[2:]    # 인덱스 2부터 마지막 요소까지 잘라서 문자열을 생성
```

```
Out[199]: 'llo, world!'
```

```
In [200]: hello[:9:2]   # 문자열의 처음부터 인덱스를 2씩 증가시키면서 인덱스 8까지 잘라서 문자열
```

```
Out[200]: 'Hlo o'
```

시퀀스객체 [시작인덱스 : 끝인덱스] = 시퀀스객체

```
In [201]: a = [0, 10, 20, 30, 40, 50, 60, 70, 80, 90]
          a[2:5] = ['a', 'b', 'c']    # 인덱스 2부터 4까지 값 할당
          a
```

```
Out[201]: [0, 10, 'a', 'b', 'c', 50, 60, 70, 80, 90]
```

```
In [203]: a = [0, 10, 20, 30, 40, 50, 60, 70, 80, 90]
          a[2:5] = ['a']    # 인덱스 2부터 4까지에 값 1개를 할당하여 요소의 개수가 줄어듦
```

```
a
```

```
Out[203]: [0, 10, 'a', 50, 60, 70, 80, 90]
```

```
In [204]: a = [0, 10, 20, 30, 40, 50, 60, 70, 80, 90]
a[2:5] = ['a', 'b', 'c', 'd', 'e'] # 인덱스 2부터 4까지 값 5개를 할당하여 요소의 개수
a
```

```
Out[204]: [0, 10, 'a', 'b', 'c', 'd', 'e', 50, 60, 70, 80, 90]
```

시퀀스객체 [시작인덱스 : 끝인덱스 : 인덱스증가폭] = 시퀀스객체

```
In [205]: a = [0, 10, 20, 30, 40, 50, 60, 70, 80, 90]
a[2:8:2] = ['a', 'b', 'c'] # 인덱스 2부터 2씩 증가시키면서 인덱스 7까지 값 할당
a
```

```
Out[205]: [0, 10, 'a', 30, 'b', 50, 'c', 70, 80, 90]
```

```
In [206]: a = [0, 10, 20, 30, 40, 50, 60, 70, 80, 90]
del a[2:5] # 인덱스 2부터 4까지 요소를 삭제
a
```

```
Out[206]: [0, 10, 50, 60, 70, 80, 90]
```

```
In [207]: a = [0, 10, 20, 30, 40, 50, 60, 70, 80, 90]
del a[2:8:2] # 인덱스 2부터 2씩 증가시키면서 인덱스 6까지 삭제
a
```

```
Out[207]: [0, 10, 30, 50, 70, 80, 90]
```

최근 3년간 연도와 인구수를 리스트로 출력

```
In [209]: year = [2011, 2012, 2013, 2014, 2015, 2016, 2017, 2018]
population = [10249679, 10195318, 10143645,
              10103233, 10022181, 9930616, 9857426, 9838892]
```

```
In [211]: print(year[-3:])
print(population[-3:])
```

```
[2016, 2017, 2018]
[9930616, 9857426, 9838892]
```

튜플 n에서 인덱스가 홀수인 요소들이 출력

```
In [212]: n = -32, 75, 97, -10, 9, 32, 4, -15, 0, 76, 14, 2
```

```
In [214]: n[1::2]
```

```
Out[214]: (75, -10, 32, -15, 76, 2)
```

input 으로 입력 받은 list 에서 마지막 5개 요소 빼는 프로그램 작성

```
In [221]: x = input().split()
del x[-5:]
tuple(x)
```



```
1 2 3 4 5 6 7 8 9 10
```

```
Out[221]: ('1', '2', '3', '4', '5')
```

```
In [222]: x = input().split()
           del x[-5:]
           tuple(x)
```

```
oven bat pony total leak wreck curl crop space navy loss knee
```

```
Out[222]: ('oven', 'bat', 'pony', 'total', 'leak', 'wreck', 'curl')
```

첫 번째 문자열에서 인덱스가 홀수인 문자와 두 번째 문자열에서 인덱스가 짝수인 문자를 연결하여 출력하는 프로그램 작성

```
In [229]: a = input()
           b = input()
           a = a[1::2]
           b = b[0::2]
           print(a + b)
```

```
python
python
yhnpto
```

```
In [230]: a = input()
           b = input()
           a = a[1::2]
           b = b[0::2]
           print(a + b)
```

```
apple
strawberry
plsrwer
```

```
In [ ]:
```