

SWEA 5356

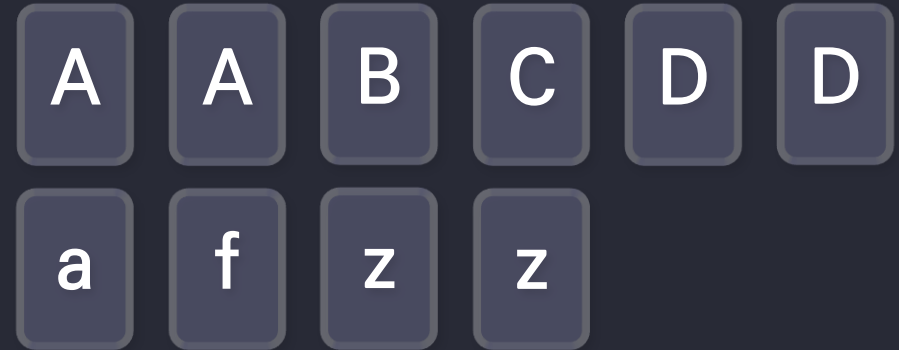
의석이의 세로로 말해요



$\left[\begin{array}{l} [A \ A \ B \ C \ D \ D], \\ [a \ f \ z \ z], \\ [0 \ 9 \ 1 \ 2 \ 1], \\ [a \ 8 \ E \ W \ g \ 6], \\ [p \ 5 \ h \ 3 \ k \ x] \end{array} \right]$

[A	A	B	C	D	D]	,
[a	f	z	z]	,
[0	9	1	2	1]	,
[a	8	E	W	g	6]	,
[p	5	h	3	k	x]	

```
def read_vertical(s):  
    answer = ''  
    cnt = 0  
    while cnt <= 15:  
        for string in s:  
            if string:  
                answer += string.pop(0)  
        cnt+=1  
    return answer  
  
for test in range(1, int(input())+1):  
    strings = list(list(input()) for _ in range(5))  
    print(f'#{test} {read_vertical(strings)}')
```



```
from collections import deque

T = int(input())

for tc in range(1, T+1):
    words = []
    res = []

    for _ in range(5):
        words.append(deque(input())) # words 내부엔 덱을 넣어 popleft() 사용

    while True:
        for word in words:
            if word:
                res.append(word.popleft())

        for word in words: # 탈출 검증 파트
            if word: # 덱이 안비었다면 탈출 X
                break
        else: # 덱이 비었다면 else에 도달. while 탈출
            break

    print(f"#{tc} ", *res, sep=' ')

# popleft()가 생각나 일부러 덱을 썼습니다. 리스트 안에 덱을 넣는 재밌는 경험이었습니.
```

```
for t in range(1, int(input())+1):  
    strings = [input() for _ in range(5)]  
    vertical = ''  
  
    for c in range(15):  
        for r in range(5):  
            if len(strings[r]) > c:  
                vertical += strings[r][c]  
  
    print(f'#{t} {vertical}')
```

Seungkyu

index # enumerate

```
T = int(input())

for test_case in range(1, T+1):
    words = []
    for _ in range(5):
        words += input().split()

    words_t = [[] for _ in range(15)] # 제일 긴 길이가 15라고 해서 배열을 만들었습니다.

    for word in words:
        for idx, chr in enumerate(word): # ex) 4글자이면 words_t[4]에 저장되도록
            words_t[idx] += chr

    print(f'#{test_case}', end=' ')
    for words in words_t:
        print(*words, sep=' ', end=' ')
    print()
```



```
T = int(input())

for tc in range(1, T + 1):
    ans = ['']*15
    for _ in range(5):
        word = input()
        for i in range(len(word)):
            # ans i 번째에 각 줄의 i번째 문자 배치
            ans[i] += word[i]

    print('#%s %s' % (tc, ''.join(ans)))
```

```
t = int(input())

for tc in range(t):
    arr = [input() for _ in range(5)] # 다섯개의 단어들 입력 받기

    m = 0 # 다섯개 단어 중 가장 긴 단어의 길이 m
    for i in range(5):
        if len(arr[i]) > m:
            m = len(arr[i])

    for i in range(5):
        difference = m - len(arr[i]) # 가장 긴 단어와 차이가 나는 것 만큼 공백을 '-'로 채우기

        arr[i] += '-' * difference # 그러면 모든 단어의 길이 통일 가능

    ans = '' # 정답을 담을 ans 생성
    for i in range(m): # arr 에 담아놔던 단어 배열을 세로로 읽기
        string = ''

        for j in range(5):
            if arr[j][i] != '-': # '-' 가 들어간 부분은 스킵하며 더해 # 가기
                string += arr[j][i]
        ans += string

    print(f'#{tc+1} {ans}')
```

```
from itertools import zip_longest

if __name__ == "__main__":
    T = int(input())

    for tc in range(1, T+1):
        arr = [input() for _ in range(5)]

        print(f"#{tc}", end = " ")

        for i in zip_longest(*arr, fillvalue=""):
            print(*i, sep="", end=" ")

        print()
```