

13718

접근방법

1. 워드의 길이 K개와 워드의 갯수 M개가 처음에 주어지는데 이는 만들 수 있는 조합의 수가 K^M 임을 알 수 있다.
2. K^M 개의 경우의 수 중 오름차순으로 하였을 때 X번째의 값만 알면 된다. 즉, 한 자리당 K개의 숫자가 가능한 M번째 digit까지의 숫자를 고르면 된다!
3. K 진수 값을 구하는 공식을 사용하여 숫자로 변환한다. 그 숫자는 가능한 K개의 알파벳을 오름차순으로 하였을 때 특정 위치의에 해당하는 알파벳으로 변환 될 수 있다.
4. 모든 값에 대하여 3번을 적용하고 기존에 string에 특정 부분을 바꿔준다

예시

문제에서 주어진 첫번째 예시를 사용해보자.

```
9 2 3 7
sol
znu
```

sol과 znu를 오름 차순으로 정리하면 los, nuz가 된다.

K = 3, M = 2 즉, 3진수로 2 digit을 가진 숫자로 표현 가능하다

- 00 = l n
- 01 = l u
- 02 = l z
- 10 = o n
- 11 = o u
- 12 = o z
- 20 = s n (7번째)
- 21 = s u
- 22 = s z

In [7]:

```
numbers = input()
original = input()
_, M, K, X = numbers.split()
M, K, X = int(M), int(K), int(X)-1

remainder=[]

# K+1 진수로 바꿔주는 부분
while X >= K:
    remainder.append(X%(K)) ## 이때 index가 1 더해져 있음
    X = X// (K)

remainder.append(X) ## 다 나누고 남은 나머지도 remainder에 넣어줌

if len(remainder) != M:
    for _ in range(M-len(remainder)):
        remainder.append(0)

letter = []

# 저장한 remainder 리스트에서 값을 하나씩 빼와서 해당 단어를 입력 받고 바꾸기
# 이때 remainder가 먼저 없어지는 경우도 생각해야 함 씨부렐
while remainder:
    #입력 받음
    letters = input()
```

```
#입력 받은 스트링을 list로 바꿈
for i in range(len(letters)):
    letter.append(letters[i])

letter.sort()

original = original.replace('#',letter[remainder.pop()],1)
letter=[]

print(original)
```

```
9 2 3 7
po#olje#i
sol
znu
posoljeni
```