

# 백준 1759 암호 만들기

202020988 조아영

# 001 문제 소개

# 002 아이디어

# 003 코드 설명

# # 문제 소개

## 예제 입력 1 복사

```
4 6  
a t c i s w
```

## 예제 출력 1 복사

```
acis  
acit  
aciw  
acst  
acsw  
actw  
aist  
aisw  
aitw  
astw  
cist  
cisw  
citw  
istw
```

< 암호의 조건 >

- 서로 다른 L개의 알파벳 소문자들로 구성
- 최소 한개의 모음 (a, e, i, o, u)
- 최소 두개의 자음
- 증가하는 순서로 배열 (abc(O), bac(X))

입력 : 두 정수 L, C ( $3 \leq L \leq C \leq 15$ )  
암호로 사용된 문자의 종류 C가지

출력 : 가능성있는 모든 암호 (사전순)

# # 예시

가능한 모든 암호의 조합 !

백트래킹 !

암호 하나씩 append, pop하며

조건 하나하나 비교

1. 원하는 암호의 길이가 되면
2. 자음, 모음의 최소 개수 충족하는 지 확인

+ 입력받은 암호가 될 수 있는 문자들은 사전순으로 정렬하여 증가하는 배열의 암호가 되도록 함 !

# # 코드 설명

```
1 import sys
2 def dfs(cnt, index):
3     if(cnt == L):
4         v = c = 0
5         for i in result:
6             if i in vowel:
7                 v += 1
8             else:
9                 c += 1
10        if v >= 1 and c >= 2:
11            for i in result:
12                print(i, end="")
13            print("")
14        return
```

3행 : 암호의 길이가 L이 되면  
4~9행 : 모음과 자음의 개수 세어  
10행 : 암호의 조건과 일치하면 출력  
14행 : return

```
16 for i in range(index, C):
17     if(check[i]):
18         continue
19
20     check[i] = True
21     result.append(arr[i])
22     dfs(cnt + 1, index + 1)
23     result.pop()
24     for i in range(i+1, C):
25         check[i] = False
```

17행 : 이미 사용한 문자인 경우  
20행 : 사용한 문자임을 표시  
21행 : 암호 append  
22행 : cnt, index 값 1씩 증가, 함수 호출  
( 다음 암호 글자 선택 )  
23행 : 문자 제거  
24~25행 : 사전 순 암호 위해  
check 배열의 값 저장

# # 코드 설명

```
27 L, C = map(int, sys.stdin.readline().split())
28 arr = sys.stdin.readline().split()
29
30 arr.sort()
31 check = [False] * C
32 result = []
33 vowel = ['a', 'e', 'i', 'o', 'u']
34
35 dfs(0, 0)
```

27행 : 암호의 길이, 암호 문자 개수 입력  
28행 : 암호 문자 입력

30행 : 암호 문자 정렬 ( 사전 순 암호 위해 )  
31행 : 암호 사용 여부 체크할 배열 초기화  
32행 : 암호 저장할 배열 초기화  
33행 : 모음 저장

35행 : 함수 호출

감사합니다