

BOJ 1759

암호 만들기

소프트웨어학과 201921017 이지우

1. 접근

문제

바로 어제 최백준 조교가 방 열쇠를 주머니에 넣은 채 깜빡하고 서울로 가 버리는 황당한 상황에 직면한 조교들은, 702호에 새로운 보안 시스템을 설치하기로 하였다. 이 보안 시스템은 열쇠가 아닌 암호로 동작하게 되어 있는 시스템이다.

암호는 서로 다른 L 개의 알파벳 소문자들로 구성되며 최소 한 개의 모음(a, e, i, o, u)과 최소 두 개의 자음으로 구성되어 있다고 알려져 있다. 또한 정렬된 문자열을 선호하는 조교들의 성향으로 미루어 보아 암호를 이루는 알파벳이 암호에서 증가하는 순서로 배열되었을 것이라고 추측된다. 즉, abc는 가능성이 있는 암호이지만 bac는 그렇지 않다.

새 보안 시스템에서 조교들이 암호로 사용했을 법한 문자의 종류는 C 가지가 있다고 한다. 이 알파벳을 입수한 민식, 영식 형제는 조교들의 방에 침투하기 위해 암호를 추측해 보려고 한다. C 개의 문자들이 모두 주어졌을 때, 가능성 있는 암호들을 모두 구하는 프로그램을 작성하시오.

입력

첫째 줄에 두 정수 L, C 가 주어진다. ($3 \leq L \leq C \leq 15$) 다음 줄에는 C 개의 문자들이 공백으로 구분되어 주어진다. 주어지는 문자들은 알파벳 소문자이며, 중복되는 것은 없다.

출력

각 줄에 하나씩, 사전식으로 가능성 있는 암호를 모두 출력한다.

1. 접근

예제 입력 1 복사

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

예제 출력 1 복사

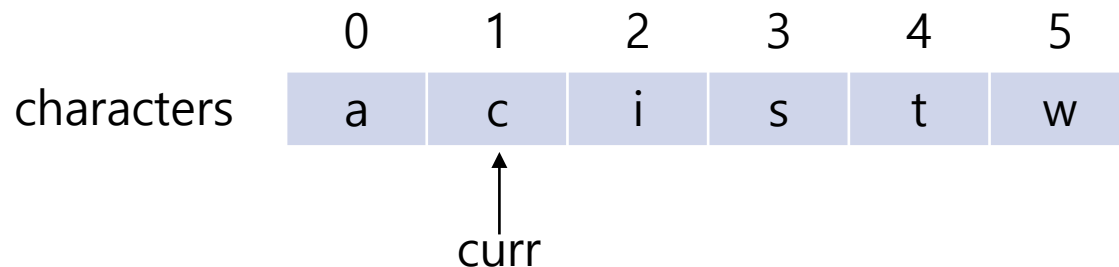
```
acis  
acit  
aciw  
acst  
acsw  
actw  
aist  
aisw  
aitw  
astw  
cist  
cisw  
citw  
istw
```

- 최대 입력의 크기가 15
- 가능한 모든 경우 출력

→ 백트래킹, 브루트포스

1. 접근

1. 암호는 알파벳이 사전순으로 배치되어 있음
→ 입력으로 들어오는 문자들을 정렬



2. 암호의 길이는 L
 - 길이가 L이 될 때까지 문자를 하나씩 추가하기
 - 이때, 글자가 중복되지 않도록 하기

→ 재귀를 사용하여 문자를 추가하되, 현재 인덱스 위치(curr)를 인자로 넘겨주어서

→ 그 다음 인덱스의 문자들이 사용될 수 있도록 해주면 됨!
3. 최소 1개의 모음과 2개의 자음으로 이루어짐
→ 만든 암호의 길이가 L이 되었을 때
→ 위의 조건을 만족하는지 for문을 사용하여 확인

2. 구현

```
public class Main {  
  
    private static int l = 0;  
    private static int c = 0;  
    private static char chs[];  
  
    public static void main(String[] args) {  
        Scanner sc = new Scanner(System.in);  
        l = sc.nextInt();  
        c = sc.nextInt();  
  
        chs = new char[c];  
        for(int i=0;i<c;i++){  
            chs[i] = sc.next().charAt(0);  
        }  
  
        Arrays.sort(chs);  
  
        solve("", -1, 0);  
    }  
}
```

```

private static void solve(String pwd, int currPos, int currLen){
    if(currLen == 1) {
        int vowel_nums = 0;
        int consonant_nums = 0;
        for (int i=0; i<1;i++) {
            if (pwd.charAt(i) == 'a' || pwd.charAt(i) == 'e' || pwd.charAt(i) == 'i' || pwd.charAt(i) == 'o' ||
pwd.charAt(i) == 'u')
                vowel_nums++;
            else
                consonant_nums++;

            if (vowel_nums >= 1 && consonant_nums >= 2) {
                System.out.println(pwd);
                break;
            }
        }
    } else if(currLen > 1)
        return;
    else{
        for(int i=currPos+1; i<c;i++)
            solve(pwd+chs[i], i, currLen+1);
    }
}
}

```

3. 결과

제출 번호	아이디	문제	문제 제목	결과	메모리	시간	언어	코드 길이
40673368	january	1759	암호 만들기	맞았습니다!!	20884 KB	284 ms	Java 11	1251 B

나 요즘 자바합니다🤖

$(\circ \cdot \forall \cdot) / ^{\circ}$

소프트웨어학과 201921017 이지우