String

문자열

❖ Palindrome

✓ 조건

- 주어진 문자열이 팰린드롬인지 확인
- 대소문자를 구분하지 않으며 영문자와 숫자 만을 대상으로 함

  입력: "A man, a plan, a canal : Panama"

  출력: true

  입력: "race a car"

  출력: false

문자열

- palindrome
  - ✓ Python
    - list를 이용한 해결

```
def isPalindrome(s):
    strs = []
    for char in s:
        if char.isalnum():
            strs.append(char.lower())

    while len(strs) > 1:
        if strs.pop(0) != strs.pop():
            return False
    return True


str1 = "A man, a plan, a canal:Panama"
print(isPalindrome(str1))

str2 = "race a car"
print(isPalindrome(str2))
```

문자열

- palindrome
  - ✓ Java
    - List를 이용한 해결

```
public class Palindrome {
    public static Boolean isPalindrome(String str) {
        List <Character> list = new ArrayList<>();

        for(int i=0; i<str.length(); i++) {
                Character ch = str.charAt(i);
                if(Character.isLetterOrDigit(ch))
                        list.add(Character.toLowerCase(ch));
        }

        while(list.size() > 1) {
                if(list.remove(0) != list.remove(list.size()-1))
                        return false;
        }
        return true;
    }
```

문자열

- palindrome
  - ✓ Java
    - List를 이용한 해결

```
        public static void main(String[] args) {

            String str1 = "A man, a plan, a canal:Panama";
            System.out.println(isPalindrome(str1));

            String str2 = "race a car";
            System.out.println(isPalindrome(str2));

        }
    }
```

문자열

- palindrome
  - ✓ Python
    - Deque를 이용한 해결

```
import collections

def isPalindrome(s):
    strs = collections.deque()
    for char in s:
        if char.isalnum():
            strs.append(char.lower())
    while len(strs) > 1:
        if strs.popleft() != strs.pop():
            return False
    return True

str1 = "A man, a plan, a canal:Panama"
print(isPalindrome(str1))

str2 = "race a car"
print(isPalindrome(str2))
```

문자열

- palindrome
  - ✓ Java
    - Deque를 이용한 해결

```
public class Palindrome {
    public static Boolean isPalindrome(String str) {
        Deque <Character> deque = new ArrayDeque<>();

        for(int i=0; i<str.length(); i++) {
                Character ch = str.charAt(i);
                if(Character.isLetterOrDigit(ch))
                        deque.add(Character.toLowerCase(ch));
        }

        while(deque.size() > 1) {
                if(deque.removeFirst() != deque.removeLast())
                        return false;
        }
        return true;
    }
```

문자열

- palindrome
  - ✓ Java
    - Deque를 이용한 해결

```
        public static void main(String[] args) {
            Long start = System.nanoTime();
            String str1 = "A man, a plan, a canal:Panama";
            System.out.println(isPalindrome(str1));

            String str2 = "race a car";
            System.out.println(isPalindrome(str2));
            Long end = System.nanoTime();

            System.out.println(end - start);
        }
    }
```

문자열

- palindrome
  - ✓ Python
    - 정규식을 이용한 해결

```
import re

def isPalindrome(s):
    s = s.lower()
    s = re.sub('[^a-z0-9]', '', s)
    return s == s[::-1]

str1 = "A man, a plan, a canal:Panama"
print(isPalindrome(str1))

str2 = "race a car"
print(isPalindrome(str2))
```

문자열

- palindrome
  - ✓ Java
    - 정규식을 이용한 해결

```
public class Palindrome {
    public static Boolean isPalindrome(String str) {
        str = str.toLowerCase();
        str = str.replaceAll("[^a-z0-9]", "");
        StringBuilder sb = new StringBuilder(str);
        return str.equals(sb.reverse().toString());
    }

    public static void main(String[] args) {
        Long start = System.nanoTime();
        String str1 = "A man, a plan, a canal:Panama";
        System.out.println(isPalindrome(str1));

        String str2 = "race a car";
        System.out.println(isPalindrome(str2));
        Long end = System.nanoTime();

        System.out.println(end - start);
    }
}
```

문자열

- ❖ 문자열 뒤집기
  - ✓ 조건
    - ● 문자의 List 이고 리턴없이 List를 직접 조작해서 수행

문자열

❖ 문자열 뒤집기

✓ Python

● 2개의 포인터 이용

```
import time

def reverseStrtng(s):
    left, right = 0, len(s) - 1
    while left < right:
        s[left], s[right] = s[right], s[left]
        left += 1
        right -= 1

start = time.time()
str1 = ["H", "e", "l", "l", "o"]
reverseStrtng(str1)
print(str1)

str2 = ["A", "D", "A", "M"]
reverseStrtng(str2)
print(str2)

end = time.time()
print(end - start)
```

문자열

❖ 문자열 뒤집기

✓ Java

● 2개의 포인터 이용

```
public class Reverse {
    public static void reverse(List<Character> list) {
        int left = 0;
        int right = list.size() - 1;
        while(left < right) {
                    Character temp = list.get(left);
                    list.set(left, list.get(right));
                    list.set(right, temp);
                    left = left + 1;
                    right = right - 1;
        }
    }

    public static void main(String[] args) {
        Character [] ar = {'H', 'E', 'L', 'L', 'O'};
        List <Character> list1 = Arrays.asList(ar);
        reverse(list1);
        System.out.println(list1);
    }
}
```

문자열

- 문자열 뒤집기
  - ✓ Python
    - 제공되는 메서드 이용

```
import time

def reverseStrtng(s):
    s.reverse()

start = time.time()
str1 = ["H", "e", "l", "l", "o"]
reverseStrtng(str1)
print(str1)

str2 = ["A", "D", "A", "M"]
reverseStrtng(str2)
print(str2)

end = time.time()
print(end - start)
```

문자열

❖ 문자열 뒤집기

✓ Java

● 제공되는 메서드 이용

```
public class Reverse {
    public static void reverse(List<Character> list) {
       Collections.reverse(list);
    }

    public static void main(String[] args) {
       Character [] ar = {'H', 'E', 'L', 'L', 'O'};
       List <Character> list1 = Arrays.asList(ar);
       reverse(list1);
       System.out.println(list1);
    }
}
```

문자열

❖ 문자열 뒤집기

✓ Python

● 슬라이싱 이용

```
import time

def reverseStrtng(s):
    s[:] = s[::-1]

start = time.time()
str1 = ["H", "e", "l", "l", "o"]
reverseStrtng(str1)
print(str1)

str2 = ["A", "D", "A", "M"]
reverseStrtng(str2)
print(str2)

end = time.time()
print(end - start)
```

문자열

- ❖ 자연수를 뒤집어서 배열로 만들기
  - ✓ 문제
    - 자연수 n을 뒤집어 각 자리 숫자를 원소로 가지는 배열 형태로 리턴
    - 예를 들어 n 이 12345 이면 [5, 4, 3, 2, 1]을 리턴
    - N은 10,000,000,000 이하의 자연수

문자열

❖ 자연수를 뒤집어서 배열로 만들기

✓ Python

```
def solution(num):
    strs = str(num)
    ar = []
    for ch in strs:
        ar.insert(0, ch)
    return ar

print(solution(12345))
```

문자열

❖ 자연수를 뒤집어서 배열로 만들기

✓ Java

```
public class Reverse {
    public static int[] solution(long n) {
     //문자열로 변환
        String str = Long.toString(n);
        //문자열 뒤집기
        String reversed = new StringBuilder(str).reverse().toString();
        //문자 배열로 변환
        char[] arr = reversed.toCharArray();
        //문자 배열을 정수 배열로 변환
        int[] result = new int[arr.length];
        for (int i = 0; i < result.length; i++) {
            result[i] = arr[i] - '0';
        }
        return result;
    }
```

문자열

❖ 자연수를 뒤집어서 배열로 만들기

✓ Java

```
        public static void main(String [] args) {
         System.out.println(Arrays.toString(solution(12345)));
        }
    }
```

문자열

❖ 시저 암호

✓ 문제

- 알파벳을 일정한 거리만큼 밀어서 다른 알파벳으로 변경하는 암호화 방식이 시저 암호
- AB를 1밀면 BC가 되고 3밀면 DE
- z를 1밀면 a
- 문자열 s 와 거리 n을 입력받아 암호문을 만드는 함수를 작성
- 문자의 구성은 대문자, 소문자, 공백으로 구성
- s 의 길이는 8000 이하
- n 은 1이상 25 이하
- 공백은 그대로 공백

문자열

❖ 시저 암호

✓ Python

```
def solution(s, n):
    s = list(s)
    for i in range(len(s)):
        if s[i] == ' ': continue
        corr = ord('A') if s[i].isupper() else ord('a')
        s[i] = chr((ord(s[i]) - corr + n) % 26 + corr)

    return ''.join(s)

print(solution("AB", 1))
print(solution("AB", 3))
print(solution("z", 1))
```

문자열

❖ 시저 암호

✓ Java

```
public class Caesar {
    private static char push(char c, int n) {
        if (!Character.isAlphabetic(c)) return c;

        int offset = Character.isUpperCase(c) ? 'A' : 'a';
        int position = c - offset;
        position = (position + n) % ('Z' - 'A' + 1);
        return (char) (offset + position);
    }

    public static String solution(String s, int n) {
        StringBuilder builder = new StringBuilder();
        for (char c : s.toCharArray()) {
            builder.append(push(c, n));
        }
        return builder.toString();
    }
```

문자열

❖ 시저 암호

✓ Java

```
        public static void main(String [] args) {
         System.out.println(solution("AB", 1));
         System.out.println(solution("AB", 3));
         System.out.println(solution("z", 1));
        }
    }
```

문자열

- ❖ 문자열 생성
  - ✓ 문제
    - 문자열 s는 한 개 이상의 단어로 구성
    - 각 단어는 하나 이상의 공백 문자로 구분
    - 각 단어의 짝수 번째 알파벳은 대문자로 홀수 번째 알파벳은 소문자로 바꾼 문자열을 리턴하는 함수를 작성
    - 문자열의 짝수/홀수 인덱스가 아니고 단어 내에서의 짝수/홀수 번째 인덱스로 판단
    - 첫번째 글자는 0번째 인덱스로 판단해서 짝수 번째로 판단
    - 입력: try hello world
    - 출력: TrY HeLlO WoRlD

문자열

❖ 문자열 생성

✓ Python

```
def solution(s):
    s = list(s)
    cnt = 0

    for i in range(len(s)):
        if s[i] == ' ':
            cnt = 0
            continue

        s[i] = s[i].upper() if cnt % 2 == 0 else s[i].lower()
        cnt += 1

    return ''.join(s)


print(solution("try hello world"))
```

문자열

❖ 문자열 생성

✓ Java

```
public class StringCreate {
    public static String solution(String s) {
        StringBuilder builder = new StringBuilder();
        boolean toUpper = true;

        for (char c : s.toCharArray()) {
            if (!Character.isAlphabetic(c)) {
                builder.append(c);
                toUpper = true;
            } else {
                if (toUpper) {
                    builder.append(Character.toUpperCase(c));
                } else {
                    builder.append(Character.toLowerCase(c));
                }
                toUpper = !toUpper;
            }
        }

        return builder.toString();
    }
```

문자열

❖ 문자열 생성

✓ Java

```
        public static void main(String [] args) {
         System.out.println(solution("try hello world"));
        }
    }
```

문자열

- 3진법
  - ✓ 문제
    - 자연수 n을 매개변수로 받아서 3진법으로 치환한 후 앞뒤로 뒤집은 후 다시 10진법으로 변환해서 리턴하는 함수를 작성
    - 자연수 n 은 100,000,000 이하의 자연수
    - 입력 45 -> 출력 7
    - 입력 125 -> 229

문자열

❖ 3진법

✓ Python

```
def solution (num):
    if num == 0:
        return '0'
    nums = []
    while num:
        num, digit = divmod(num, 3)
        nums.append(str(digit))
    return int(''.join(reversed(nums))[::-1], 3)

print(solution(45))
print(solution(125))
```

문자열

❖ 문자열 생성

✓ Java

```
public class TernarySystem {
    public static Integer solution(int n) {
        String str = Integer.toString(n, 3);
        String reversed = new StringBuilder(str).reverse().toString();
        return Integer.valueOf(reversed, 3);
    }

    public static void main(String [] args) {
     System.out.println(solution(45));
     System.out.println(solution(125));
    }
}
```

문자열

- 문자 개수 세기
  - ✓ 문제
    - 대문자 와 소문자가 섞여 있는 문자열 s에서 p 와 y의 개수가 일치하면 true 그렇지 않으면 false를 리턴하는 함수를 작성
    - p 와 y 모두 없는 경우에는 true를 리턴
    - 대소문자를 구분하지 않음
    - 입력: pPoooyY 출력: true
    - 입력: pyy 출력: false

문자열

❖ 문자 개수 세기

✓ Python

```
def solution (s):
    ps = 0
    ys = 0
    for ch in s:
        if ch.lower() == 'p':
            ps += 1
        elif ch.lower() == 'y':
            ys += 1
    if ps == ys:
        return True
    return False

print(solution("pPoooyY"))
print(solution("pyy"))
```

문자열

❖ 문자 개수 세기

✓ Java

```
public class CharacterCount {
    boolean solution(String s) {
            int ps = 0;
            int ys = 0;
            for (char c : s.toCharArray()) {
                    switch (c) {
                    case 'p', 'P' -> ps++;
                    case 'y', 'Y' -> ys++;
                    }
            }
            return ps == ys;
    }

    public static void main(String [] args) {
            System.out.println("pPoooyY");
            System.out.println("pyy");
    }
}
```

문자열

- ❖ 숫자 문자열 과 영단어
  - ✓ 문제
    - 네오 와 프로도가 숫자 놀이를 하고 있는데 네오가 프로도에게 숫자를 건넬 때 일부 자릿수를 영단어로 바꾼 카드를 건네 주면 프로도는 원래 숫자를 찾는 게임
    - 예
      - 1478 one4seveneight
      - 234567 23four5six7
      - 10203 1zerotwozero3
    - 제한 사항
      - 1 <= s의 길이 <= 50
      - s가 zero 또는 0 으로 시작하는 경우는 주어지지 않음
      - return 값이 1 이상 2,000,000,000 이하의 정수가 되는 올바른 입력만 s로 주어짐
      -

문자열

- ❖ 숫자 문자열 과 영단어
  - ✓ 문제
    - ● 숫자에 해당하는 영단어
      - ◆ 0 zero
      - ◆ 1 one
      - ◆ 2 two
      - ◆ 3 three
      - ◆ 4 four
      - ◆ 5 five
      - ◆ 6 six
      - ◆ 7 seven
      - ◆ 8 eight
      - ◆ 9 nine

문자열

❖ 숫자 문자열 과 영단어

✓ Python

```
def solution(s):
    words = {
            "zero":"0", "one":"1", "two":"2", "three":"3", "four":"4",
        "five":"5", "six":"6", "seven":"7", "eight":"8", "nine":"9",
    };
    for key, val in words.items():
        s = s.replace(key, val)
    return s

print(solution("one4seveneight"))
print(solution("23four5six7"))
print(solution("1zerotwozero3"))
```

문자열

❖ 숫자 문자열 과 영단어

✓ Java

```
public class DigitAndEngWord {
    private static final String[] words = {
            "zero", "one", "two", "three", "four", "five", "six", "seven", "eight", "nine",
    };

    public static int solution(String s) {
        for (int i = 0; i < words.length; i++) {
            s = s.replace(words[i], Integer.toString(i));
        }

        return Integer.parseInt(s);
    }

    public static void main(String[] args) {
            System.out.println(solution("one4seveneight"));
            System.out.println(solution("23four5six7"));
            System.out.println(solution("1zerotwozero3"));
    }
}
```

문자열

- ❖ 문자열 다루기
  - ✓ 문제
    - 문자열 s의 길이가 4 혹은 6이고 숫자로만 구성돼 있는지 확인해주는 함수를 생성
    - 예
      - ◆ a234 -> false
      - ◆ 1234 -> true

문자열

❖ 문자열 다루기

✓ Python

```
import re

def solution(s):
    return len(s) in {4,6} and bool(re.match('^[0-9]*$', s))

print(solution("a234"))
print(solution("1234"))
```

문자열

❖ 문자열 다루기

✓ Java

```
public class StringEdit {

    public static boolean solution(String s) {
        if (s.length() != 4 && s.length() != 6) return false;

        for (char c : s.toCharArray()) {
            if (!Character.isDigit(c)) return false;
        }

        return true;
    }

    public static void main(String[] args) {
            System.out.println(solution("a234"));
            System.out.println(solution("1234"));
    }
}
```

문자열

❖ 문자열 다루기

✓ Java

```
public class StringEdit {
    public static boolean solution(String s) {
        return s.matches("[0-9]{4}|[0-9]{6}");
    }

    public static void main(String[] args) {
            System.out.println(solution("a234"));
            System.out.println(solution("1234"));
    }
}
```

문자열

❖ 신규 아이디 추천

✓ 문제

- 카카오에 입사한 신입 개발자 네오는 카카오계정개발팀에 배치되어 카카오 서비스에 가입하는 유저들의 아이디를 생성하는 업무를 담당하게 되었습니다.
- 네오에게 주어진 첫 업무는 새로 가입하는 유저들이 카카오 아이디 규칙에 맞지 않는 아이디를 입력했을 때 입력된 아이디와 유사하면서 규칙에 맞는 아이디를 추천해주는 프로그램을 개발하는 것
- 카카오 아이디의 규칙
  - 아이디의 길이는 3자 이상 15자 이하
  - 아이디는 알파벳 소문자, 숫자, 빼기(-), 밑줄(_), 마침표(.) 문자만 사용할 수 있음
  - 마침표(.)는 처음과 끝에 사용할 수 없으며 또한 연속으로 사용할 수 없음

문자열

❖ 신규 아이디 추천

✓ 문제

- 네오는 다음과 같이 7단계의 순차적인 처리 과정을 통해 신규 유저가 입력한 아이디가 카카오 아이디 규칙에 맞는 지 검사하고 규칙에 맞지 않은 경우 규칙에 맞는 새로운 아이디를 추천
- 신규 유저가 입력한 아이디가 new_id 라고 하는 경우
  - new_id의 모든 대문자를 대응되는 소문자로 치환
  - new_id에서 알파벳 소문자, 숫자, 빼기(-), 밑줄(_), 마침표(.)를 제외한 모든 문자를 제거
  - new_id에서 마침표(.)가 2번 이상 연속된 부분을 하나의 마침표(.)로 치환
  - new_id에서 마침표(.)가 처음이나 끝에 위치한다면 제거
  - new_id가 빈 문자열이라면, new_id에 "a"를 대입
  - new_id의 길이가 16자 이상이면, new_id의 첫 15개의 문자를 제외한 나머지 문자들을 모두 제거하는데 제거 후 마침표(.)가 new_id의 끝에 위치한다면 끝에 위치한 마침표(.) 문자를 제거
  - new_id의 길이가 2자 이하라면, new_id의 마지막 문자를 new_id의 길이가 3이 될 때까지 반복해서 끝에 추가

문자열

- ❖ 신규 아이디 추천
  - ✓ 문제
    - new_id 값이 "...!@BaT#*..y.abcdefghijklm" 라면 위 7단계를 거치고 나면 new_id는 아래와 같이 변경
      - ◆ 1단계 대문자 'B'와 'T'가 소문자 'b'와 't ' 로 변경

        "...!@BaT#*..y.abcdefghijklm" → "...!@bat#*..y.abcdefghijklm"
      - ◆ 2단계 '!', '@', '#', '*' 문자가 제거

        "...!@bat#*..y.abcdefghijklm" → "...bat..y.abcdefghijklm"
      - ◆ 3단계 '...'와 '..' 가 '. ' 로 변경

        "...bat..y.abcdefghijklm" → ".bat.y.abcdefghijklm"
      - ◆ 4단계 아이디의 처음에 위치한 '.'가 제거

        ".bat.y.abcdefghijklm" → "bat.y.abcdefghijklm"
      - ◆ 5단계 아이디가 빈 문자열이 아니므로 변화가 없음

        "bat.y.abcdefghijklm" → "bat.y.abcdefghijklm"
      - ◆ 6단계 아이디의 길이가 16자 이상이므로, 처음 15자를 제외한 나머지 문자들 제거

        "bat.y.abcdefghijklm" → "bat.y.abcdefghi"

문자열

- ❖ 신규 아이디 추천
  - ✓ 문제
    - new_id 값이 "...!@BaT#*..y.abcdefghijklm" 라면 위 7단계를 거치고 나면 new_id는 아래와 같이 변경
      - ◆ 7단계 아이디의 길이가 2자 이하가 아니므로 변화가 없음

        "bat.y.abcdefghi" → "bat.y.abcdefghi"
      - ◆ 신규 유저가 입력한 new_id가 "...!@BaT#*..y.abcdefghijklm"일 때, 네오의 프로그램이 추천하는 새로운 아이디는 "bat.y.abcdefghi"
    - 신규 유저가 입력한 아이디를 나타내는 new_id가 매개변수로 주어질 때 네오가 설계한 7단계의 처리 과정을 거친 후의 추천 아이디를 return 하도록 solution 함수를 작성

문자열

❖ 신규 아이디 추천

✓ Python

```
import re

def solution(new_id):
    # 1
    answer = new_id.lower()
    # 2
    filtered = []
    for c in answer:
        if c.isalpha() or c.isdigit() or c in {'-', '_', '.'}:
            filtered.append(c)
    answer = ''.join(filtered)
    # 3
    while '..' in answer:
        answer = answer.replace('..', '.')
    # 4
    answer = answer.strip('.')
```

문자열

❖ 신규 아이디 추천

✓ Python

```
# 5
if answer == '': answer = 'a'
# 6
if len(answer) > 15: answer = answer[:15]
if answer[-1] == '.': answer = answer[:-1]
# 7
while len(answer) < 3:
    answer += answer[-1]
return answer


print(solution("...!@BaT#*..y.abcdefghijklm"))
```

문자열

❖ 신규 아이디 추천

✓ Java

```
public class IDRecommand {

    public static String solution(String newId) {
        newId = newId.toLowerCase();
        newId = newId.replaceAll("[^a-z0-9₩₩-_.]", "");
        newId = newId.replaceAll("₩₩.+", ".");
        newId = newId.replaceAll("^₩₩.+|₩₩.+$", "");
        if (newId.isEmpty()) newId = "a";
        if (newId.length() >= 16) {
            newId = newId.substring(0, 15);
            newId = newId.replaceAll("₩₩.+$", "");
        }
        while (newId.length() < 3) {
            newId += newId.charAt(newId.length() - 1);
        }

        return newId;
    }
```

문자열

❖ 신규 아이디 추천

✓ Java

```
        public static void main(String[] args) {
                System.out.println(solution("...!@BaT#*..y.abcdefghijklm"));
        }
}
```