

문제 1. 주어진 Prob1.java 파일을 이용하여 문자열내의 특정 문자를 지정한 문자로 수정하는 myReplace 메서드를 완성하십시오. myReplace 메서드의 첫번째 매개변수는 수정대상이 되는 문자열이며 두번째 매개변수는 문자열내에 존재하는 문자이며 세번째 매개변수는 변경하고자 하는 문자입니다. myReplace 메서드는 변경한 문자열을 리턴하며 만약 문자열내에 변경대상이 되는 문자가 없을 경우

변경 없이 그대로 리턴해야 합니다.

<<수행결과>>

```
문자열에 특정문자 변경하는 테스트
-----Sample 1 -----
he**o wor*d
-----Sample 2 -----
hello-world
-----Sample 3 -----
hello world
```

<<참고 및 주의사항>>

- main 메서드는 수정없이 그대로 사용합니다.
- 주어진 메서드의 signature 는 변경하지 않습니다.
- String 클래스의 replace, replaceAll 메서드와 StringTokenizer 클래스는 사용하지 말아야 합니다.

문제 2. Prob2 내에 leftPad 메소드를 작성하십시오. leftPad 메소드의 속성은 다음과 같습니다.

1. 메소드 시그니처 : public String leftPad(String str, int size, char padChar)
2. 입력 문자열(str)의 길이가 입력받은 수(size)가 될 때까지 문자열의 좌측을 지정된 문자(padChar)로 채운 후 리턴한다.
3. 단, 입력 문자열(str)의 길이가 입력받은 수(size) 보다 큰 경우에는 원본 문자열(str)을 그대로 리턴한다.

실행 예)

```
leftPad("ACORN", 8, '#')→ ### ACORN
leftPad("ACOR",7,'$')→$$ACORN
leftPad("ACOR", 5, '&') → ACORN
```

문제 3. 다음은 주어진 숫자(number)에서 지정된 숫자(i)를 찾는 FindNumbers 클래스 입니다. 입력값에 따라 다음과 같은 출력이 나오도록, 해당 int 의 개수를 리턴하는 countNumber 메소드를 작성하시오.

```
int countNumber(long number, int i)
```

- 숫자의 각 값들을 비교하여 입력받는 int i의 값과 동일한 숫자의 개수를 리턴한다.
- 반드시 1 값은 한자리 수이어야 한다.

#### [ 실행 예시 ]

```
입력 숫자 = 228754462
2 횟수 => 3
4 횟수 => 2
```

문제 4. 다음은 주어진 문자열에서 문자를 찾는 FindCharacters 클래스 입니다. 입력값에 따라 다음과 같은 출력이 나오도록, 해당 char 의 개수를 리턴하는 countChar 메소드를 작성하시오.

```
int countChar(String str, char c)
```

- 문자열의 각 내용들을 비교하여 입력받는 char c의 값과 동일한 char 의 개수를 리턴한다.
- 대소문자는 각각 다른 것으로 간주한다.

#### <<처리 결과>>

입력 스트링 : String s = " Boys, be ambitious" ;

a. c = 'i' 인 경우 => 2 리턴

b. c = 'b' 인 경우 => 2 리턴 (B와 b는 서로 다르므로 다른 것으로 간주)

참고 및 주의사항 :

- String 클래스의 toCharArray() 메소드를 활용할 수 있습니다. (필수 아님)
- 테스트 목적으로 스스로 main을 구현할 수 있습니다.

- main 메소드의 내용은 채점하지 않습니다.
- 주어진 메소드의 signature 는 변경하지 않습니다.

문제 5. 임의의 문자열(String str)과 구분자(char separator)를 입력 받아 str 에 포함된 구분자를 기준으로 str 을 문자열 배열로 만들어 반환하는 기능의 split() 메소드를 작성하세요.

```
public String[] split(String str, char separator){  
    // TODO 이부분을 작성  
  
}
```

<메인메소드 실행결과>

```
배열 크기 : 4  
서울시 강남구 테헤란로 에이콘아카데미
```

<<참고 및 주의사항>>

- String 클래스의 split() 메소드를 사용하지 말고 직접 구현해야 합니다.
- main 메소드의 내용은 채점하지 않으므로 테스트 목적으로 수정할 수 있습니다.
- 주어진 메소드의 signature 는 변경하지 않습니다.