문자열 압축

알파벳 대문자로 이루어진 문자열을 입력받아 같은 문자가 연속으로 반복되는 경우 반복되는 문자 바로 오른쪽에 반복 횟수를 표기하는 방법으로 문자열을 압축하는 프로그램을 작성하세요. 단 반복횟수가 1인 경우 생략합니다.

입출력 예:

s	result
"KKHSSSSSSSE"	"K2HS7E"
"AAABCCCDD"	"A2BC3D2"

제한사항:

- 문자열 s의 길이는 100을 넘지 않습니다.

기본코드형태:

```
import java.util.*;
class Main {
        public String solution(String s){
                String answer="";
                return answer;
        }

        public static void main(String[] args){
                Main T = new Main();
                System.out.println(T.solution("KKHSSSSSSSE"));
        }
}
```

회문 문자열

앞에서 읽을 때나 뒤에서 읽을 때나 같은 문자열을 회문 문자열이라고 합니다.
문자열이 입력되면 해당 문자열이 회문 문자열이면 "YES", 회문 문자열이 아니면 “NO"를 출력하는 프로그램을 작성하세요.
단 회문을 검사할 때 대소문자를 구분하지 않습니다.

입출력 예:

s	result
"gooG"	"YES"
"Moon"	"NO"

제한사항:

- 문자열 s의 길이는 100을 넘지 않습니다.

특정 문자 뒤집기

영어 알파벳과 특수문자로 구성된 문자열이 주어지면 영어 알파벳만 뒤집고, 특수문자는 자기 자리에 그대로 있는 문자열을 만들어 반환하는 프로그램을 작성하세요.

입출력 예:

s	result
"a#b!GE*T@S"	"S#T!EG*b@a"
"###ab*@@Sty"	"###yt*@@Sba"

제한사항:

- 문자열 s의 길이는 100을 넘지 않습니다.

회문문자열2

문자열 s가 주어지면 s가 최대 문자 1개까지 지워서 회문문자열이 되면 “YES"를 출력하고, 그렇지 않으면 ”NO"를 출력하는 프로그램을 작성하세요.

입출력 예:

s	result
"abcbdcba"	"YES"
"abcabbakcba"	"YES"
"abcacbakcba"	"NO"

제한사항:

- 문자열 s의 길이는 100을 넘지 않습니다.

입력예제 1 설명 :

abcbdcba에서 4번째 문자인 b를 제거하면 abcdcba로 회문이된다. 또는 5번째 문자인 d를 제거하면 abcbcba로 회문이 된다.

학급 회장

학급 회장을 뽑는데 후보로 기호 A, B, C, D, E 후보가 등록을 했습니다.
투표용지에는 반 학생들이 자기가 선택한 후보의 기호(알파벳)가 쓰여져 있으며 선생님은 그 기호를 발표하고 있습니다.
매개변수 s에 투표용지에 쓰여져 있던 각 후보의 기호가 선생님이 발표한 순서대로 문자열로 주어지면 어떤 기호의 후보가 학급 회장이 되었는지 반환하는 프로그램을 작성하세요.
반드시 한 명의 학급회장이 선출되도록 투표결과가 나왔다고 가정합니다.

입출력 예:

s	result
"BACBACCACCBDEDE"	C
"AAAAABBCCDDDD"	A

제한사항:

- 문자열 s의 길이는 100을 넘지 않습니다.

입력예제 1 설명 :

A기호 3표, B기호 3표, C기호 5표, D기호 2표, E기호 2표 를 받아 C가 학급회장이 되었습니다.

한번 사용한 최초 문자

문자열에서 한번만 사용한 문자를 찾으려고 합니다.
한번만 사용한 문자 중 문자열에서 먼저 나타난 문자의 인덱스 번호를 반환하는 프로그램을 작성하세요. 인덱스는 1부터 시작합니다. 한번만 사용한 문자가 없을 경우 -1를 반환하세요.

입출력 예:

s	result
"statitsics"	3
"aabb"	-1
"stringshowtime"	3

제한사항:

- 문자열 s의 길이는 100을 넘지 않습니다.
- 문자열은 소문자로만 이루어져 있습니다.

입력예제 1 설명 :

한번만 사용한 문자는 a, c이고, 문자열에서 먼저 나타난 것은 a이고 인덱스는 3입니다.

같은 빈도수 만들기

소문자 a, b, c로 이루어진 문자열이 주어지면 해당 문자열에서 a, b, c의 최소의 개수를 추가하여 a, b, c의 빈도수가 동일하게 되도록 해야 합니다. 동일빈도수가 되는 최소 추가 개수를 알파벳 a, b, c순으로 배열에 저장하여 반환하는 프로그램을 작성하세요.
만약 주어진 문자열이 "aaabc" 라면 빈도수는 a:3 , b:1, c:1 이고 최소 개수를 추가하여 동일 빈도수가 되게 하려면 b를 2개 추가, c를 2개 추가하면 모두 빈도수가 3개로 동일해집니다.

입출력 예:

s	result
"aaabc"	[0, 2, 2]
"aabb"	[0, 0, 2]
"abc"	[0, 0, 0]

제한사항:

- 문자열 s의 길이는 100을 넘지 않습니다.

팰린드롬 길이

문자열이 주어지면 해당 문자열의 문자들을 가지고 만들 수 있는 최대길이 팰린드롬을 만들고 그 길이를 구하세요. 문자열은 소문자로만 이루어져 있습니다.
만약 “abcbbbccaa" 가 주어진다면 만들 수 있는 가장 긴 팰린드롬은 ”bbcaaacbb"이고 답은 9입니다.

입출력 예:

s	result
"abcbbbccaa"	9
"abcde"	1
"ccc"	3

제한사항:

- 문자열 s의 길이는 100을 넘지 않습니다.

음성인식

음성 인식 기술을 사용하면 사람이 말하는 음성 데이터를 문자 데이터로 변환할 수 있습니다. 당신은 오늘의 집에 전화로 들어온 주문을 자동으로 처리하기 위해, 음성 데이터를 문자 데이터로 변환하려 합니다. 당신은 이 문자 데이터를 쓰기 전에 먼저 반복적으로 사용된 말버릇 패턴을 삭제해야 합니다. 말버릇 패턴이란 문자 데이터에서 가장 많이 등장하는 길이 1 이상의 패턴이며, 문자 데이터에 등장하는 해당 패턴을 모두 삭제하면 됩니다. 단, 이러한 패턴은 대소문자를 구분하지 않으며, 가장 많이 등장한 패턴이 여러 개일 경우 그러한 패턴을 모두 삭제합니다.

다음은 문자 데이터에서 말버릇 패턴을 삭제하는 예시입니다.

삭제 전	삭제 후
“abcabcdefabc"	"def"
"abxdeydeabz"	"xyz"

• “abcabcdeabc"에서 ”abc"가 3번 등장하며, 가장 많이 등장한 패턴입니다. ”abc"를 삭제하면 "def"가 남게 됩니다.
• "abxdeydeabz"에서 “ab"와 ”de"가 2번 등장하며, 가장 많이 등장한 패턴입니다. “ab"와 ”de"를 삭제하면 "xyz"가 남게 됩니다.

음성 데이터를 문자 데이터로 변환한 문자열 cell이 매개변수로 주어집니다. 이때, 가장 많이 등장한 말버릇을 삭제한 결과를 문자열로 return하도록 solution 함수를 완성해주세요.

<제한사항>
• 3<= cell의 길이 <=1,000
• cell은 알파벳 대소문자로만 구성되어 있습니다.
• 가장 많이 등장하는 말버릇 패턴을 삭제했을 때, 남은 문자열의 길이가 1이상인 경우만 입력으로 주어집니다.

입출력 예

cell	result
“abcabcdefabc"	"def"
"abxdeydeabz"	"xyz"
"abcabca"	"bcbc"
"ABCabcA"	"BCbc"

입출력 예 #3
“abcabca"에서 가장 많이 등장한 패턴은 ”a“입니다. ”a“를 삭제하면 "bcbc"가 남게 됩니다.
입출력 예 #4
대소문자가 다른 점을 제외하면 3번 예시와 같습니다. "ABCabcA"에서 가장 많이 등장한 패턴은 대소문자를 구분하지 않으므로 “A"/"a"입니다. “A"/"a"를 삭제하면 "BCbc"가 남게 됩니다.

공통문자찾기

N개의 문자열이 주어지면 모든 문자열에 공통으로 들어있는 문자를 찾아 출력하는 프로그램을 작성하세요. 만약 어떤 문자가 모든 문자열에서 2번 나타난다면 답에서도 2번 나타나게 해야 합니다.
문자열 배열 words가 주어지면 모든 문자열에 공통으로 들어있는 공통문자를 문자배열형태로 반환하는 프로그램을 작성하세요.
문자의 순서는 상관없습니다. 반드시 공통문자는 존재합니다.

입출력 예:

words	result
["steasue", "sasseysu", "kseseas"]	["s", "s", "e", "a"]
["ackky", "kabck", "yokkcs"]	["k", "k", "c"]
["longlong", "longtong", "longbig"]	["l", "o", "n", "g", "g"]

제한사항:

- 문자열 words의 길이는 30을 넘지 않습니다.
- words[i]의 길이는 100을 넘지 않습니다.
- 모든 문자열은 소문자로 이루어져 있습니다.

가장 가까운 시간

0시부터 24시까지의 시간이 “HH:MM" 의 표현으로 각 시간대가 주어집니다. 이 중 임의의 2개의 시간대의 차가 가장 작은 시간차이를 분단위로 구하여 반환하는 프로그램을 작성하세요.

입출력 예:

times	result
["00:12","00:00", "01:05", "00:57"]	8
["00:00","23:59","00:00"]	0
["23:59","00:00", "23:57"]	1

제한사항:

- 문자열 times의 길이는 10,000을 넘지 않습니다.

입출력 1번 설명
“00:57”과 “01:05”와의 시간차이가 8분으로 가장 차이가 적다.

공부시간

당신은 스마트폰 어플리케이션을 이용하여 공부한 시간을 기록하려합니다. 어플리케이션의 기능은 다음과 같습니다.

1. 시작 버튼 : 공부를 시작할 때의 시각을 기록합니다.
2. 중지 버튼 : 공부를 중지할 때의 시각을 기록합니다.

하지만, 어플리케이션에 기록된 시간에 항상 공부만 했다는 보장이 없기 때문에 다음과 같은 규칙을 적용해 실제로 공부한 시간을 구하려 합니다.

1. 공부를 시작하고 5분이 지나기 전에 중지했다면 실제로 공부한 시간에 포함시키지 않습니다.
2. 공부를 시작하고 1시간 45분이 넘어서 중지했다면 1시간 45분까지만 공부한 시간으로 인정합니다.

공부를 시작한 시각과 중지한 시각이 연속적으로 주어집니다. 처음 주어진 시각은 무조건 공부를 시작한 시각이며, 마지막 시각은 무조건 공부를 중한 시각입니다. 차례대로 번갈아 가면서 [시작 시각, 중지 시각, 시작 시각, 중지 시작, ... , 중지 시각] 형태로 주어집니다. 이때 실제로 공부한 시간을 구하려 합니다.
어플리케이션의 기록을 담은 문자열 log가 매개변수로 주어졌을 때, 실제로 공부한 시간을 HH:MM 형태로 return하도록 solution 함수를 완성해주세요.

• HH:MM 형태를 시:분을 뜻합니다. 이때 시 혹은 분이 한 자리 수라면 왼쪽에 0을 채워 항상 두자리게 되게 합니다.

■ 제안사항

• log의 길이는 짝수입니다.
• 2≤log의 길이≤1,440
  ◦ log의 원소는 시각을 나타내며 길이는 항상 5입니다.
  ◦ 시각은 항상 시:분을 뜻하는 HH:MM 형태로 주어집니다.
  ◦ 잘못된 시각은 주어지지 않습니다.
  ◦ 중복된 시각은 주어지지 않습니다.
  ◦ 시각은 오름차순으로 주어지며, 모두 같은 날 기록한 내용만 주어집니다.

■ 입출력 예

log	result
["08:30", "09:00", "14:00", "16:00", "16:01", "16:06", "16:07", "16:11"]	"02:20"
["01:00", "08:00", "15:00", "15:04", "23:00", "23:59"]	"02:44"