1

Write a Program that will check whether a given String is Palindrome or not

```
String
```

2

Given two strings, append them together (known as "concatenation") and return the result. However, if the concatenation creates a double-char, then omit one of the chars. If the inputs are “Mark” and “Kate” then the ouput should be “markate”. (The output should be in lowercase)

```
String
```

3

Given a string, return a new string made of n copies of the first 2 chars of the original string where n is the length of the string. The string may be any length. If there are fewer than 2 chars, use whatever is there. If input is "Wipro" then output should be "WiWiWiWiWi".

```
String
```

4

Given a string of even length, return the first half. So the string "CatDog" yields "Cat". If the string length is odd number then return null.

```
String
```

5

Given a string, return a version without the first and last char, so "Wipro" yields "ipr". The string length will be at least 2.

```
String
```

6

Given 2 strings, a and b, return a string of the form short+long+short, with the shorter string on the outside and the longer string on the inside. The strings will not be the same length, but they may be empty (length 0). If input is "hi" and "hello", then output will be "hihellohi".

```
String
```

7

Given a string, if the first or last chars are 'x', return the string without those 'x' chars, and otherwise return the string unchanged. If the input is "xHix", then output is "Hi".

```
String
```

8

Given two strings, word and a separator, return a big string made of count occurrences of the word, separated by the separator string. if the inputs are "Wipro","X" and 3 then the output is "WiproXWiproXWipro".

```
String
```

9

Return a version of the given string, where for every star (*) in the string the star and the chars immediately to its left and right are gone. So "*ab*cd" yields "ad" and "ab**cd" also yields "ad".

```
String
```

10

Given two strings, a and b, create a bigger string made of the first char of a, the first char of b, the second char of a, the second char of b, and so on. Any leftover chars go at the end of the result. If the inputs are "Hello" and "World", then the output is "HWeolrllod".

```
String
```

11

Given two strings, a and b, create a bigger string made of the first char of a, the first char of b, the second char of a, the second char of b, and so on. Any leftover chars go at the end of the result. If input is “abc” and “xyz”, then output should be “axbycz”.

```
String
```

12

Given a string and an int n, return a string made of n repetitions of the last n characters of the string. You may assume that n is between 0 and the length of the string, inclusive. For example if the inputs are “Wipro” and 3, then the output should be “propropro”.

```
String
```

13

Given a string and a non-empty word string, return a string made of each char just before and just after every appearance of the word in the string. Ignore cases where there is no char before or after the word, and a char may be included twice if it is between two words.

If inputs are "abcXY123XYijk" and "XY", output should be "c13i". If inputs are "XY123XY" and "XY", output should be "13". If inputs are "XY1XY" and "XY", output should be "11".

`String`