Recursive Programming: Palindromes

**Case 0: Length of String ≤ 1 (Base Case)**

*Handling this case:* Return **true** since single characters and the empty string are trivially palindromes.

The diagram below will be helpful for understanding cases 1 – 4.

```
x   ...   y
```

$x$ and $y$ are being used as symbols to denote the *first* and *last* characters, respectively.

**Case 1: $x$ is not a letter**

*Handling this case:*

```
x   ...   y
```

**Case 2: $y$ is not a letter**

*Handling this case:*

```
x   ...   y
```

**Case 3: $x = y$**

*Handling this case:*

```
x   ...   y
```

**Case 4: $x \neq y$**

*Handling this case:* Return **false** since $x$ and $y$ are **letters that do not match**, breaking any possibility of a palindrome. Note the strategic placement of this case.

**Side Note:** Cases 1 – 3 can be placed in any order – do you know why?

**Methods useful for this lab**

**String class:**
- `int length()`
- `char charAt(int)`
- `String substring(int, int)`

**Character class:**
- `static boolean isLetter(char)`