14 ways to reverse a string in java

<https://www.techiedelight.com/10-ways-reverse-a-string-java/>

1. Using charAt() and String Concatenation operator

2.Using StringBuilder reverse()

3. Using StringBuffer reverse()

4. Using toCharArray

5. Using TempArray

6. Using charArray toCharArray() and swap

7. Using byteArray getBytes() and swap

8. Using List and Collections reverse() and StringBuilder

9. Using List and Collections reverse() and replaceAll()

10. Using Stack and Array

11. Using Stack and StringBuilder

12. Using Unicode Right to Left Override (RLO)

13. Using Recursion

14. Using Substring

**Using charAt() and String Concatenation operator**

**public** String reverseStringUsingCharAt(String str, **int** n) {

**if** (str == **null** || str.isEmpty() || str.equals(""))

**return** str;

String result = "";

**for** (**int** i = n - 1; i >= 0; i--) {

result = result + str.charAt(i);

}

**return** result;

}

**Using StringBuilder reverse()**

**public** String reverseStringUsingStringBuilder(String str, **int** n) {

**if** (str == **null** || str.isEmpty() || str.equals(""))

**return** str;

**return** **new** StringBuilder(str).reverse().toString();

}

**Using StringBuffer reverse()**

**public** String reverseStringUsingStringBuffer(String str, **int** n) {

**if** (str == **null** || str.isEmpty() || str.equals(""))

**return** str;

**return** **new** StringBuffer(str).reverse().toString();

}

**Using toCharArray**

**public** String reverseStringUsingtoCharArray(String str, **int** n) {

**if** (str == **null** || str.isEmpty() || str.equals(""))

**return** str;

**char**[] arr = str.toCharArray();

String result = "";

n = arr.length;

**for** (**int** i = n - 1; i >= 0; i--) {

result = result + arr[i];

}

**return** result;

}

**Using TempArray**

**public** String reverseStringUsingTempArray(String str, **int** n) {

**if** (str == **null** || str.isEmpty() || str.equals(""))

**return** str;

**char**[] tempArr = **new** **char**[n];

**for** (**int** i = 0; i < n; i++) {

tempArr[n - i - 1] = str.charAt(i);

}

String result = **new** String(tempArr);

**return** result;

}

**Using charArray toCharArray() and swap**

**public** String reverseStringUsingCharArrayAndSwap(String str, **int** n) {

**if** (str == **null** || str.isEmpty() || str.equals(""))

**return** str;

**char**[] arr = str.toCharArray();

**for** (**int** l = 0, h = n - 1; l < h; l++, h--) {

**char** temp = arr[l];

arr[l] = arr[h];

arr[h] = temp;

}

String result = **new** String(arr);

**return** result;

}

**Using byteArray getBytes() and swap**

**public** String reverseStringUsingByteArrayAndSwap(String str, **int** n) {

**if** (str == **null** || str.isEmpty() || str.equals(""))

**return** str;

**byte**[] arr = str.getBytes();

**for** (**int** l = 0, h = n - 1; l < h; l++, h--) {

**byte** temp = arr[l];

arr[l] = arr[h];

arr[h] = temp;

}

String result = **new** String(arr);

**return** result;

}

**Using List and Collections reverse() and StringBuilder**

**public** String reverseStringUsingCollectionsReverseMethod(String str, **int** n) {

**if** (str == **null** || str.isEmpty() || str.equals(""))

**return** str;

**char**[] arr = str.toCharArray();

List<Character> list = **new** ArrayList<Character>();

**for** (**char** c : arr) {

list.add(c);

}

Collections.*reverse*(list);

StringBuilder sb = **new** StringBuilder(n);

**for** (**char** c : list) {

sb.append(c);

}

**return** sb.toString();

}

**Using List and Collections reverse() and replaceAll()**

**public** String reverseStringUsingCollectionsReverseMethodAndReplaceAll(String str, **int** n) {

**if** (str == **null** || str.isEmpty() || str.equals(""))

**return** str;

**char**[] arr = str.toCharArray();

List<Character> list = **new** ArrayList<Character>();

**for** (**char** c : arr) {

list.add(c);

}

Collections.*reverse*(list);

**return** list.toString().replaceAll("[,\\[\\]]", "").replaceAll(" ", "@").replaceAll(" ", "").replaceAll("@",

" ");

}

**Using Stack and Array**

**public** String reverseStringUsingStackAndArray(String str, **int** n) {

**if** (str == **null** || str.isEmpty() || str.equals(""))

**return** str;

**char**[] arr = str.toCharArray();

Stack<Character> stack = **new** Stack<Character>();

**for** (**int** i = 0; i < n; i++) {

stack.push(arr[i]);

}

**int** index = 0;

**while** (!stack.isEmpty()) {

arr[index++] = stack.pop();

}

String result = **new** String(arr);

**return** result;

}

**Using Stack and StringBuilder**

**public** String reverseStringUsingStackAndStringBuilder(String str, **int** n) {

**if** (str == **null** || str.isEmpty() || str.equals(""))

**return** str;

Stack<Character> stack = **new** Stack<Character>();

**for** (**int** i = 0; i < n; i++) {

stack.push(str.charAt(i));

}

StringBuilder sb = **new** StringBuilder();

**while** (!stack.isEmpty()) {

sb.append(stack.pop());

}

**return** sb.toString();

}

**Using Unicode Right to Left Override (RLO)**

**public** String reverseStringUsingUnicodeRightToLeftOverride(String str, **int** n) {

**if** (str == **null** || str.isEmpty() || str.equals(""))

**return** str;

**return** "\u202E" + str;

}

**Using Recursion**

**public** String reverseStringUsingRecursion(String str, **int** n) {

**if** (str == **null** || str.isEmpty() || str.equals(""))

**return** str;

**char**[] arr = str.toCharArray();

reverse(arr, 0, n - 1);

String result = **new** String(arr);

**return** result;

}

**public** **void** reverse(**char**[] arr, **int** l, **int** h) {

**if** (l < h) {

**char** temp = arr[l];

arr[l] = arr[h];

arr[h] = temp;

reverse(arr, l + 1, h - 1);

}

}

**Using Substring**

Solution 1:

**public** String reverse(String str, **int** n) {

**if** (str == **null** || str.isEmpty() || str.equals(""))

**return** str;

**return** str.charAt(str.length() - 1) + reverse(str.substring(0, str.length() - 1), str.length());

}

Solution 2:

**public** String reverse(String str) {

**if** (str == **null** || str.isEmpty() || str.equals(""))

**return** str;

**return** str.charAt(str.length() - 1) + reverse(str.substring(0, str.length() - 1));

}

Solution 3:

**public** String reverse(String str) {

**if** (str == **null** || str.isEmpty() || str.equals(""))

**return** str;

**return** reverse(str.substring(1)) + str.charAt(0);

}