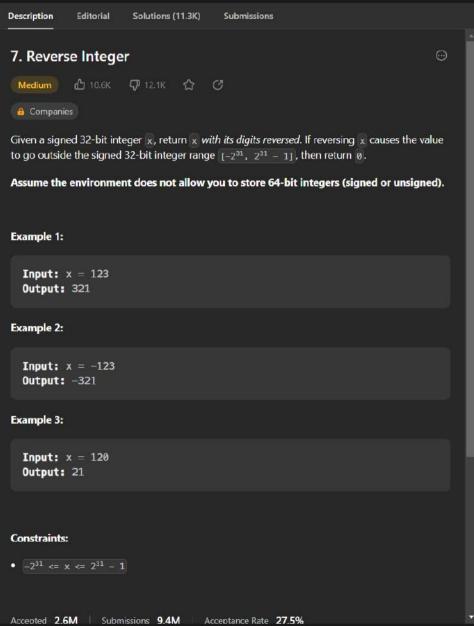


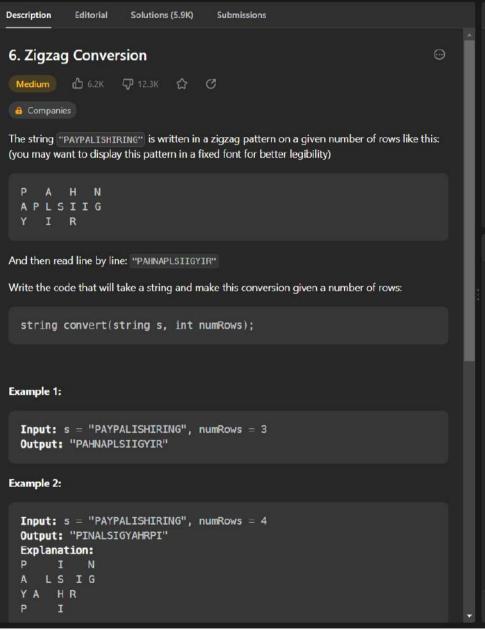
```
1 class Solution {
         public int myAtoi(String s) {
            int len = s.length();
             if (len == 0) {
                 return 0; // Handle empty string case
             double num = 0:
             int i = 0;
             while (i < len && s.charAt(i) == ' ') {
                 i++;
             if (i == len) {
                 return 0; // All characters are whitespace
Testcase
        Result
Accepted Runtime: 0 ms

    Case 3

 Case 1
             Case 2
Input
 "42"
Output
 42
Expected
 42
                                        O Contribute a testcase
                                                                                            Submit
Console Y
                                                                                   Run
```



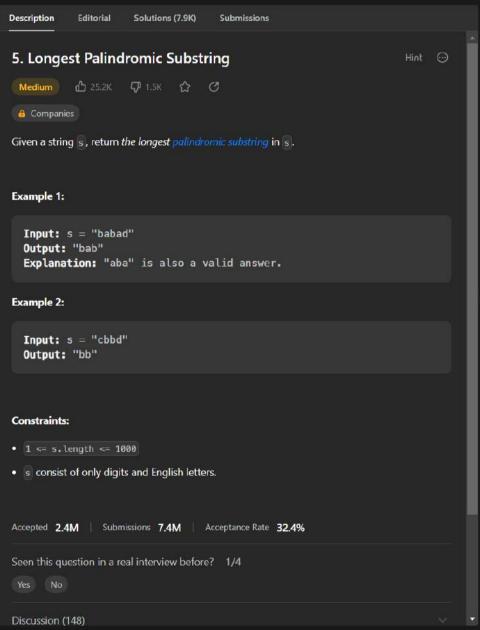
```
i Java ∨ Auto
         public int reverse(int x) {
              long reverse = 0;
             while (x != 0) {
                 int digit = x % 10;
                 reverse = reverse * 10 + digit;
             if (reverse > Integer.MAX VALUE | | reverse < Integer.MIN VALUE) return 0;
 10
             return (int) reverse;
Testcase Result
Accepted Runtime: 0 ms
 Case 1
             • Case 2 • Case 3
  123
Output
 321
Expected
 321
                                         O Contribute a testcase
Console Y
                                                                                    Run
                                                                                            Submit
```



```
i Java ∨ Auto
         public String convert(String s, int numRows) {
             if (numRows == 1) {
                  return s;
             StringBuilder result = new StringBuilder();
             int n = s.length();
             int cycleLen = 2 * numRows - 2;
             for (int i = 0; i < numRows; i++) {
                  for (int j = 0; j + i < n; j \leftarrow cycleLen) {
                     result.append(s.charAt(j + i));
                      if (i != 0 && i != numRows - 1 && j + cycleLen - i < n) {
                          result.append(s.charAt(j + cycleLen - i));
Testcase Result
Accepted Runtime: 0 ms

    Case 3

 Case 1
             • Case 2
Input
 "PAYPALISHIRING"
 3
Output
 "PAHNAPLSIIGYIR"
Expected
 "PAHNAPLSIIGYIR"
                                                                                               Submit
Console Y
                                                                                      Run
```



```
int maxLen = 0;
         int lo = 0;
         public String longestPalindrome(String s) {
              char[] input = s.toCharArray();
             if(s.length() < 2) {
                 return s;
             for(int i = 0; i<input.length; i++) {</pre>
                 expandPalindrome(input, i, i);
                 expandPalindrome(input, i, i+1);
             return s.substring(lo, lo+maxLen);
Testcase
Accepted Runtime: 0 ms
             • Case 2
 Case 1
Input
  "babad"
Output
  "bab"
Expected
  "bab"
                                         O Contribute a testcase
Console Y
                                                                                   Run
                                                                                            Submit
```