```
import java.io.*;
import java.math.*;
import java.security.*;
import java.text.*;
import java.util.*;
import java.util.concurrent.*;
import java.util.function.*;
import java.util.regex.*;
import java.util.stream.*;
import static java.util.stream.Collectors.joining;
import static java.util.stream.Collectors.toList;

class Result {

   /*
    * Complete the 'passort' function below.
    *
    * The function is expected to return an INTEGER.
    * The function accepts STRING line as parameter.
    */

   public static int passort(String line) {
          String input = line;

          String cleanInput = "";
          for (char c : input.toCharArray())
          {
             if ((c >= 'a' && c <= 'z') || (c >= 'A' && c <= 'Z') || (c >= '0' && c <= '9'))
                cleanInput += c;
```

```
}

char[] inputArray = cleanInput.toCharArray();
Arrays.sort(inputArray);
String goal = String.valueOf(inputArray);

int sortNumber = 0;
while (!cleanInput.equals(goal))
{
   for (int i = 0; i < cleanInput.length() - 1; i++)
   {
      if (goal.charAt(i) != cleanInput.charAt(i))
      {
         int smallest = cleanInput.indexOf(goal.charAt(i), i);

         char[] tmpArray = cleanInput.toCharArray();
         char swap = tmpArray[i];

         if (swap != tmpArray[smallest])
         {
            tmpArray[i] = tmpArray[smallest];
            tmpArray[smallest] = swap;
            cleanInput = String.valueOf(tmpArray);

            sortNumber++;
         }

         break;
      }
```

```
            }

            for (int i = cleanInput.length() - 1; i > 1; i--)
            {
               if (goal.charAt(i) != cleanInput.charAt(i))
               {
                  int largest = cleanInput.lastIndexOf(goal.charAt(i), i);

                  char[] tmpArray = cleanInput.toCharArray();
                  char swap = tmpArray[i];

                  if (swap != tmpArray[largest])
                  {
                     tmpArray[i] = tmpArray[largest];
                     tmpArray[largest] = swap;
                     cleanInput = String.valueOf(tmpArray);

                     sortNumber++;
                  }

                  break;
               }
            }
         }
      return sortNumber;
   }

}
```

```
public class Solution {

    public static void main(String[] args) throws IOException {

        BufferedReader bufferedReader = new BufferedReader(new InputStreamReader(System.in));

        BufferedWriter bufferedWriter = new BufferedWriter(new
FileWriter(System.getenv("OUTPUT_PATH")));


        String line = bufferedReader.readLine();


        int result = Result.passort(line);


        bufferedWriter.write(String.valueOf(result));

        bufferedWriter.newLine();


        bufferedReader.close();

        bufferedWriter.close();

    }

}
```