

Industrial strength Java/JEE Career Companion to open more doors

[Home](#)
[Java FAQs](#)
[600+ Java Q&As](#)
[Career](#)
[Tutorials](#)
[Member](#)
[Why?](#)
[Can u Debug?](#)
[Java 8 ready?](#)
[Top X](#)
[Productivity Tools](#)
[Judging Experience?](#)

[Home](#) > [Interview](#) > [Core Java Interview Q&A](#) > [Algorithms](#) > ♦ Splitting input text & string processing coding Q&A

♦ Splitting input text & string processing coding Q&A

Posted on [April 2, 2015](#) by [Arulkumaran Kumaraswamipillai](#)

Q1. Can you write code to extract numbers from a given input text and return the sum of those numbers?

Example: "1 some text 2 more text 4 more text 8" should return 15 as the sum.

A The key steps are:

1. Split the given input into tokens. Split on 1 or more spaces.
2. Loop through the tokens of text and sum if it is a number
3. Return the sum.

Solution

```
1 package algorithms;
2
```

600+ Full Stack Java/JEE Interview Q&As ♥Free ♦FAQs

[open all](#) | [close all](#)

✚ [Ice Breaker Interview](#)

✚ [Core Java Interview C](#)

✚ [Java Overview \(4\)](#)

✚ [Data types \(6\)](#)

✚ [constructors-methc](#)

✚ [Reserved Key Wor](#)

✚ [Classes \(3\)](#)

✚ [Objects \(8\)](#)

✚ [OOP \(10\)](#)

✚ [GC \(2\)](#)

✚ [Generics \(5\)](#)

✚ [FP \(8\)](#)

✚ [IO \(7\)](#)

✚ [Multithreading \(12\)](#)

✚ [Algorithms \(5\)](#)

✚ ♦ [Splitting input t](#)

✚ ♦ [Tree traversal ;](#)

✚ ♥ ♦ [Java coding](#)

✚ [Searching algori](#)

✚ [Swapping, partiti](#)

✚ [Annotations \(2\)](#)

```

3 public class ExtractAndSumNumbersFromString {
4
5     public static void main(String[] args) {
6
7         int sumOfNumbers = getSumOfNumbers("1 so
8         System.out.println(sumOfNumbers);
9     }
10
11     public static int getSumOfNumbers(String s)
12     {
13         int total = 0;
14         String[] input = s.split("\\s+"); //spli
15
16         //loop through each split input
17         for (String in : input) {
18             int nextNumber = 0;
19
20             //determine if number or not
21             try {
22                 nextNumber = Integer.parseInt(in
23             } catch (NumberFormatException nfe)
24             {
25                 //not a number if exception is t
26             } finally {
27                 total += nextNumber;
28             }
29         }
30         return total;
31     }
32 }
33

```

- [Collection and Data](#)
- [Differences Between](#)
- [Event Driven Progr](#)
- [Exceptions \(2\)](#)
- [Java 7 \(2\)](#)
- [Java 8 \(24\)](#)
- [JVM \(6\)](#)
- [Reactive Programn](#)
- [Swing & AWT \(2\)](#)
- [JEE Interview Q&A \(3](#)
- [Pressed for time? Jav](#)
- [SQL, XML, UML, JSC](#)
- [Hadoop & BigData Int](#)
- [Java Architecture Inte](#)
- [Scala Interview Q&As](#)
- [Spring, Hibernate, & I](#)
- [Testing & Profiling/Sa](#)
- [Other Interview Q&A 1](#)
- [Free Java Interview](#)

Output:

```

1 15

```

Q2. Given a string, return a version without the first 2 chars. Except keep the first char if it is 'a' and keep the second char if it is 'b'.

For example:

```

1 removeCharacters("Absolute") --> absolute
2 removeCharacters("base") --> se
3 removeCharacters("Ant") --> at
4 removeCharacters("ubiquitous") --> biquitous
5

```

A2.

Solution:

16 Technical Key Areas

[open all](#) | [close all](#)

- [Best Practice \(6\)](#)
- [Coding \(26\)](#)
- [Concurrency \(6\)](#)
- [Design Concepts \(7\)](#)
- [Design Patterns \(11\)](#)
- [Exception Handling \(3](#)
- [Java Debugging \(21\)](#)
- [Judging Experience In](#)
- [Low Latency \(7\)](#)
- [Memory Management](#)
- [Performance \(13\)](#)
- [QoS \(8\)](#)
- [Scalability \(4\)](#)
- [SDLC \(6\)](#)
- [Security \(13\)](#)
- [Transaction Managen](#)

```

1  package algorithms;
2
3  public class RemoveCharactersFromStringInput {
4
5      public static void main(String[] args) {
6          System.out.println(removeCharacters("Abs"));
7          System.out.println(removeCharacters("bas"));
8          System.out.println(removeCharacters("Ant"));
9          System.out.println(removeCharacters("ubi"));
10     }
11
12     public static String removeCharacters(String
13
14         input = input.toLowerCase();
15
16         String firstChar = (input.length() > 0 &
17         String secondChar = (input.length() > 1 &
18
19         String remaing = input.length() > 2 ? in
20
21         return firstChar + secondChar + remaing;
22     }
23 }
24 }
25
26

```

Q3. Given a string and allowed repetition number, remove characters consecutively repeated more than the repetition number?

For example:

```

1  removeConsecutiveCharacters("aaab", 2) --> "aab"
2  removeConsecutiveCharacters("aaaabbbb", 3) --> "a"
3  removeConsecutiveCharacters("aabbccccc", 1) --> "a"
4

```

A3.

1. Convert the input string to a char array
2. Loop through each character, and if the current character == previous character, and within the repetition count, do nothing. If outside the repetition count, replace the character with a space character (i.e. ' '). You could use some other characters unlikely to be used in the input text instead of a space character.
3. After looping, find and replace all the space (i.e. ' ') characters with no-space (i.e. "").

Solution

80+ step by step Java Tutorials

[open all](#) | [close all](#)

- [Setting up Tutorial \(6\)](#)
- [Tutorial - Diagnosis \(2\)](#)
- [Akka Tutorial \(9\)](#)
- [Core Java Tutorials \(2\)](#)
- [Hadoop & Spark Tuto](#)
- [JEE Tutorials \(19\)](#)
- [Scala Tutorials \(1\)](#)
- [Spring & Hibernate Ti](#)
- [Tools Tutorials \(19\)](#)
- [Other Tutorials \(45\)](#)

100+ Java pre-interview coding tests

[open all](#) | [close all](#)

- [Can you write code? \(](#)
- [♦ Complete the given](#)
- [Converting from A to I](#)
- [Designing your classe](#)
- [Java Data Structures](#)
- [Passing the unit tests](#)
- [What is wrong with th](#)
- [Writing Code Home A](#)
- [Written Test Core Jav](#)
- [Written Test JEE \(1\)](#)

How good are your?

[open all](#) | [close all](#)

- [Career Making Know-](#)

```
1 package algorithms;
2
3 public class RemoveCharactersFromStringInput {
4
5     private static final char MARKER_FOR_REMOVAL
6
7     public static void main(String[] args) {
8         System.out.println(removeConsecutiveChar
9         System.out.println(removeConsecutiveChar
10        System.out.println(removeConsecutiveChar
11        System.out.println(removeConsecutiveChar
12    }
13
14    public static String removeConsecutiveCharac
15
16        char[] charArray = input.toCharArray();
17        char lastChar = charArray[0]; //first ch
18        int count = 1; //start with 1 as first
19
20        //loop from 2nd character
21        for (int i = 1; i < charArray.length; i+
22            //Oops bad "MARKER_FOR_REMOVAL" sele
23            if(charArray[i] == MARKER_FOR_REMOVA
24                throw new IllegalArgumentExceptionExcepti
25            }
26
27            if(charArray[i] == lastChar){
28                ++count;
29                lastChar = charArray[i];
30                if(count > allowedRepetitionCoun
31                    charArray[i] = MARKER_FOR_REI
32            }
33        } else {
34            count = 1; //reset count 1 to in
35            lastChar = charArray[i];
36        }
37    }
38
39    String output= new String(charArray);
40    String result = output.replace(MARKER_FO
41
42
43
44    return result;
45
46    }
47
48 }
49
```

Output:

```
1 aab
2 aaabbb
3 abc
4 aabbabcc
5
```

Popular Posts

♦ 11 Spring boot interview questions & answers

823 views

♦ Q11-Q23: Top 50+ Core on Java OOP Interview Questions & Answers

765 views

18 Java scenarios based interview Questions and Answers

399 views

001A: ♦ 7+ Java integration styles & patterns interview questions & answers

388 views

01b: ♦ 13 Spring basics Q8 – Q13 interview questions & answers

295 views

♦ 7 Java debugging interview questions & answers

293 views

01: ♦ 15 Ice breaker questions asked 90% of the time in Java job interviews with hints

285 views

♦ 10 ERD (Entity-Relationship Diagrams) Interview Questions and Answers

279 views

♦ Q24-Q36: Top 50+ Core on Java classes, interfaces and generics interview questions & answers

239 views

001B: ♦ Java architecture & design concepts interview questions & answers

201 views

Bio

Latest Posts



Arulkumaran Kumaraswamipillai

Mechanical Eng to freelance Java developer in 3 yrs. Contracting since 2003, and attended 150+ Java job interviews, and often got 4 - 7 job offers to choose from. It pays to prepare. So, published Java interview Q&A books via [Amazon.com](https://www.amazon.com) in 2005, and sold 35,000+ copies. Books are outdated and replaced with this subscription



based site.**945+** paid members. [join my LinkedIn Group](#). [Reviews](#)



About [Arulkumaran Kumaraswamipillai](#)

Mechanical Eng to freelance Java developer in 3 yrs. Contracting since 2003, and attended 150+ Java job interviews, and often got 4 - 7 job offers to choose from. It pays to prepare. So, published Java interview Q&A books via [Amazon.com](#) in 2005, and sold 35,000+ copies. Books are outdated and replaced with this subscription based site.**945+** paid members. [join my LinkedIn Group](#). [Reviews](#)

◀ ♦ Finding the perfect number

♥ 5 tips that will put your Java contracting career on auto-pilot mode ▶

Posted in Algorithms, Can you write code?, member-paid

Empowers you to open more doors, and fast-track

Technical Know Hows

☀ [Java generics in no time](#) ☀ [Top 6 tips to transforming your thinking from OOP to FP](#) ☀ [How does a HashMap internally work? What is a hashing function?](#)
 ☀ [10+ Java String class interview Q&As](#) ☀ [Java auto un/boxing benefits & caveats](#) ☀ [Top 11 slacknesses that can come back and bite you as an experienced Java developer or architect](#)

Non-Technical Know Hows

☀ [6 Aspects that can motivate you to fast-track your career & go places](#) ☀ [Are you reinventing yourself as a Java developer?](#) ☀ [8 tips to safeguard your Java career against offshoring](#) ☀ [My top 5 career mistakes](#)

Prepare to succeed

☀ [Turn readers of your Java CV go from “Blah blah” to “Wow”?](#) ☀ [How to prepare for Java job interviews?](#) ☀ [16 Technical Key Areas](#) ☀ [How to choose from multiple Java job offers?](#)

© Disclaimer

The contents in this Java-Success are copy righted. The author has the right to correct or enhance the current content without any prior notice.

These are general advice only, and one needs to take his/her own circumstances into consideration. The author will not be held liable for any damages caused or alleged to be caused either directly or indirectly by these materials and resources. Any trademarked names or labels used in this blog remain the property of their respective trademark owners. No guarantees are made regarding the accuracy or usefulness of content, though I do make an effort to be accurate. Links to external sites do not imply endorsement of the linked-to sites.