

Assignment No:-34

Name:-Suryawanshi Sangramsinh Sambhaji

Batch: - Delta - DCA (Java) 2024 Date:-25/6/2024

TOP STRING QUESTIONS FOR INTERVIEW AND LOGIC BUILDING.

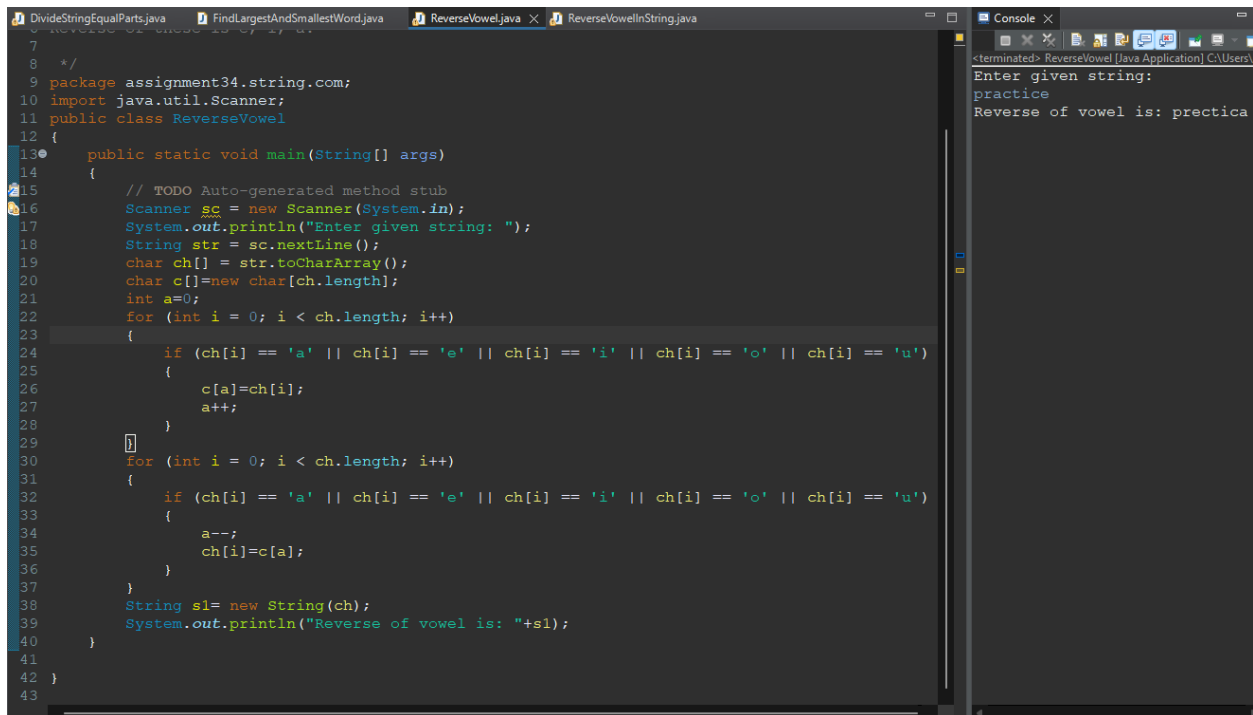
Q1. Reverse the vowels only (Flipkart)

S = "practice"

Output: prectica

Explanation: The vowels are a, i, e

Reverse of these is e, i, a.



```
7
8
9 package assignment34.string.com;
10 import java.util.Scanner;
11 public class ReverseVowel
12 {
13     public static void main(String[] args)
14     {
15         // TODO Auto-generated method stub
16         Scanner sc = new Scanner(System.in);
17         System.out.println("Enter given string: ");
18         String str = sc.nextLine();
19         char ch[] = str.toCharArray();
20         char c[]=new char[ch.length];
21         int a=0;
22         for (int i = 0; i < ch.length; i++)
23         {
24             if (ch[i] == 'a' || ch[i] == 'e' || ch[i] == 'i' || ch[i] == 'o' || ch[i] == 'u')
25             {
26                 c[a]=ch[i];
27                 a++;
28             }
29         }
30         for (int i = 0; i < ch.length; i++)
31         {
32             if (ch[i] == 'a' || ch[i] == 'e' || ch[i] == 'i' || ch[i] == 'o' || ch[i] == 'u')
33             {
34                 a--;
35                 ch[i]=c[a];
36             }
37         }
38         String s1= new String(ch);
39         System.out.println("Reverse of vowel is: "+s1);
40     }
41 }
42
43
```

Console

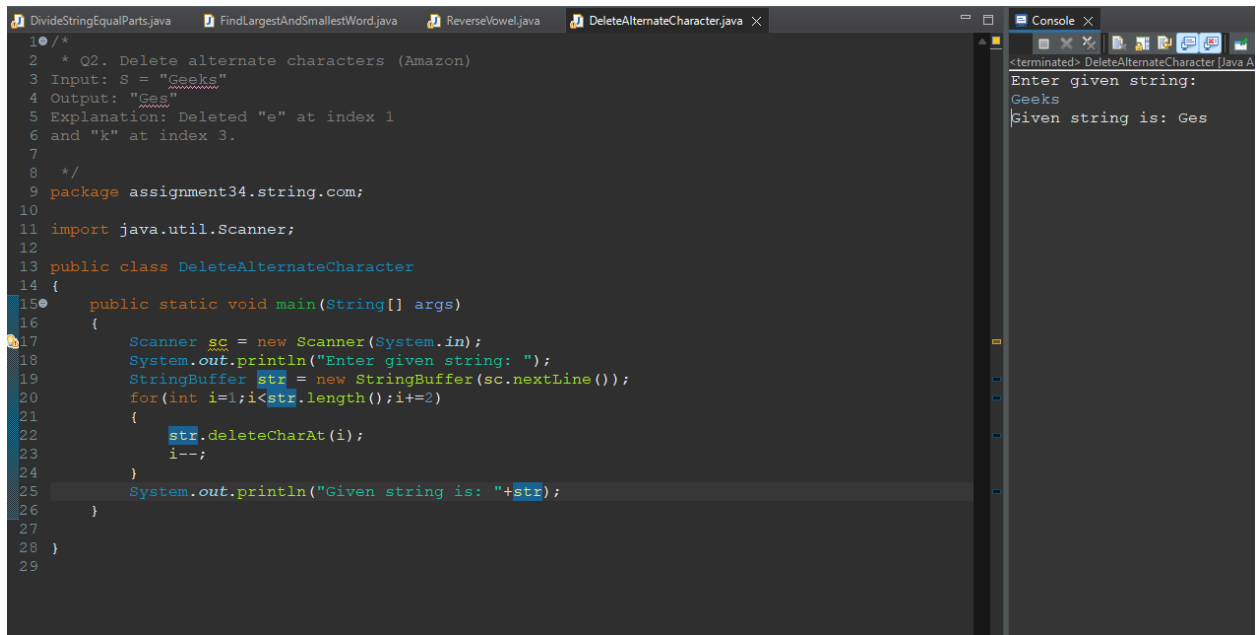
```
<terminated> ReverseVowel [Java Application] C:\Users\
Enter given string:
practice
Reverse of vowel is: prectica
```

Q2. Delete alternate characters (Amazon)

Input: S = "Geeks"

Output: "Ges"

Explanation: Deleted "e" at index 1
and "k" at index 3.



```
1  /*
2   * Q2. Delete alternate characters (Amazon)
3   Input: S = "Geeks"
4   Output: "Ges"
5   Explanation: Deleted "e" at index 1
6   and "k" at index 3.
7   */
8
9   package assignment34.string.com;
10
11   import java.util.Scanner;
12
13   public class DeleteAlternateCharacter
14   {
15       public static void main(String[] args)
16       {
17           Scanner sc = new Scanner(System.in);
18           System.out.println("Enter given string: ");
19           StringBuffer str = new StringBuffer(sc.nextLine());
20           for(int i=1; i<str.length(); i+=2)
21           {
22               str.deleteCharAt(i);
23               i--;
24           }
25           System.out.println("Given string is: "+str);
26       }
27   }
28
29
```

Console Output:

```
<terminated> DeleteAlternateCharacter [Java A
Enter given string:
Geeks
Given string is: Ges
```

Q3. Extract the integers (Zoho)

Input:

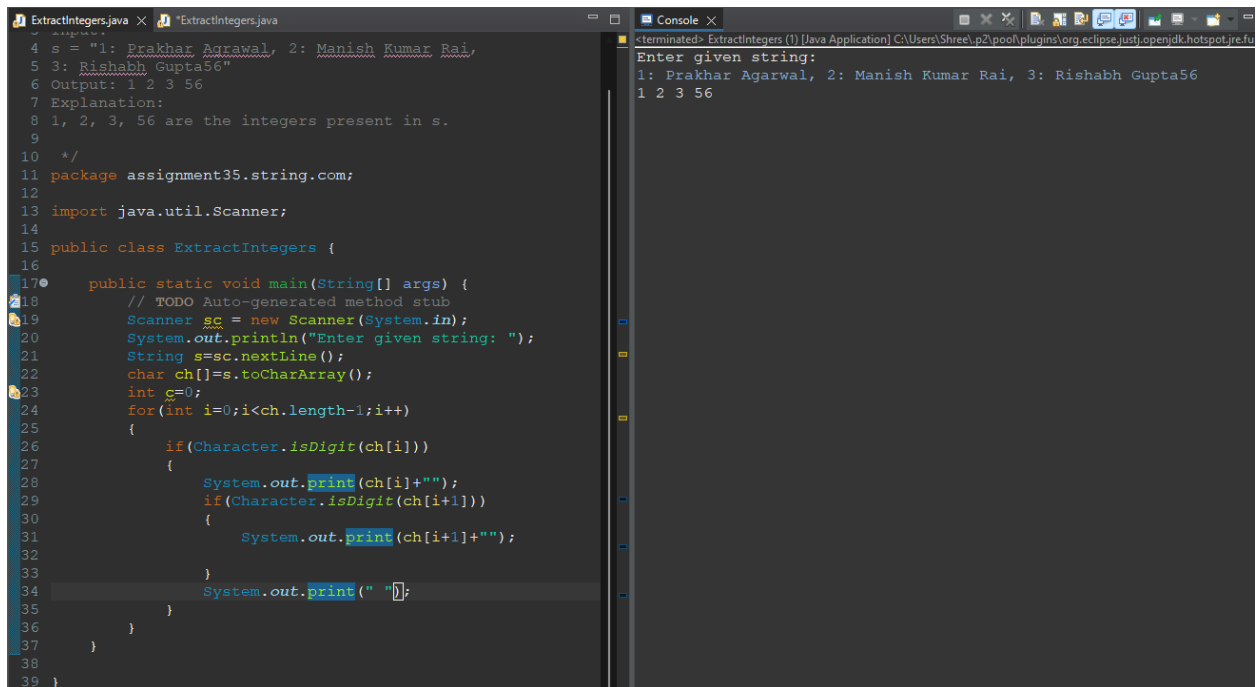
s = "1: Prakhar Agrawal, 2: Manish Kumar Rai,

3: Rishabh Gupta56"

Output: 1 2 3 56

Explanation:

1, 2, 3, 56 are the integers present in s.



```
ExtractIntegers.java x *ExtractIntegers.java
4 s = "1: Prakhar Agrawal, 2: Manish Kumar Rai,
5 3: Rishabh Gupta56"
6 Output: 1 2 3 56
7 Explanation:
8 1, 2, 3, 56 are the integers present in s.
9
10 */
11 package assignment35.string.com;
12
13 import java.util.Scanner;
14
15 public class ExtractIntegers {
16
17     public static void main(String[] args) {
18         // TODO Auto-generated method stub
19         Scanner sc = new Scanner(System.in);
20         System.out.println("Enter given string: ");
21         String s=sc.nextLine();
22         char ch[]=s.toCharArray();
23         int c=0;
24         for(int i=0;i<ch.length-1;i++)
25         {
26             if(Character.isDigit(ch[i]))
27             {
28                 System.out.print(ch[i]+"");
29                 if(Character.isDigit(ch[i+1]))
30                 {
31                     System.out.print(ch[i+1]+"");
32                 }
33                 System.out.print(" ");
34             }
35         }
36     }
37 }
38
39 }
```

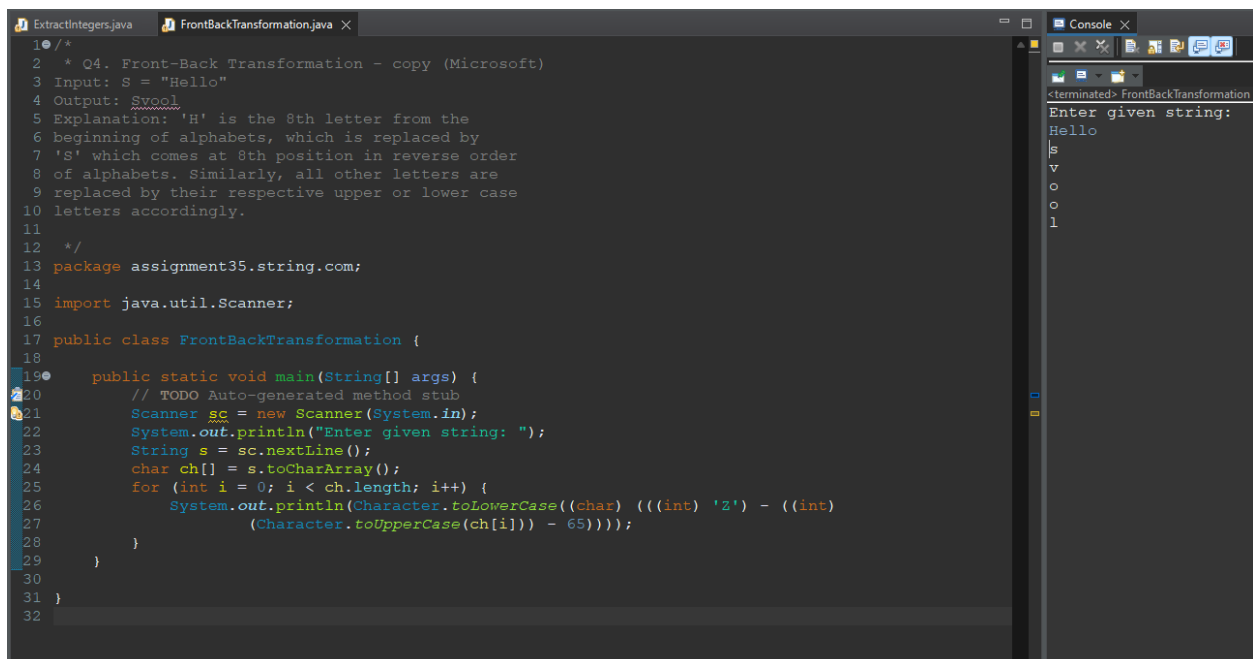
```
<terminated> ExtractIntegers (1) [Java Application] C:\Users\Shree\p2\pool\plugins\org.eclipse.jdt.openjdk.hotspot.jre.fu
Enter given string:
1: Prakhar Agarwal, 2: Manish Kumar Rai, 3: Rishabh Gupta56
1 2 3 56
```

Q4. Front-Back Transformation - copy (Microsoft)

Input: S = "Hello"

Output: Svool

Explanation: 'H' is the 8th letter from the beginning of alphabets, which is replaced by 'S' which comes at 8th position in reverse order of alphabets. Similarly, all other letters are replaced by their respective upper or lower case letters accordingly.



```
1  /*
2   * Q4. Front-Back Transformation - copy (Microsoft)
3   Input: S = "Hello"
4   Output: Svool
5   Explanation: 'H' is the 8th letter from the
6   beginning of alphabets, which is replaced by
7   'S' which comes at 8th position in reverse order
8   of alphabets. Similarly, all other letters are
9   replaced by their respective upper or lower case
10  letters accordingly.
11  */
12
13  package assignment35.string.com;
14
15  import java.util.Scanner;
16
17  public class FrontBackTransformation {
18
19      public static void main(String[] args) {
20          // TODO Auto-generated method stub
21          Scanner sc = new Scanner(System.in);
22          System.out.println("Enter given string: ");
23          String s = sc.nextLine();
24          char ch[] = s.toCharArray();
25          for (int i = 0; i < ch.length; i++) {
26              System.out.println(Character.toLowerCase((char) (((int) 'Z') - ((int)
27                  (Character.toUpperCase(ch[i])) - 65))));
28          }
29      }
30  }
31  }
32  }
```

Console

```
<terminated> FrontBackTransformation
Enter given string:
Hello
|s
v
o
o
l
```

Q5. Uncommon characters (Zoho)

Input:

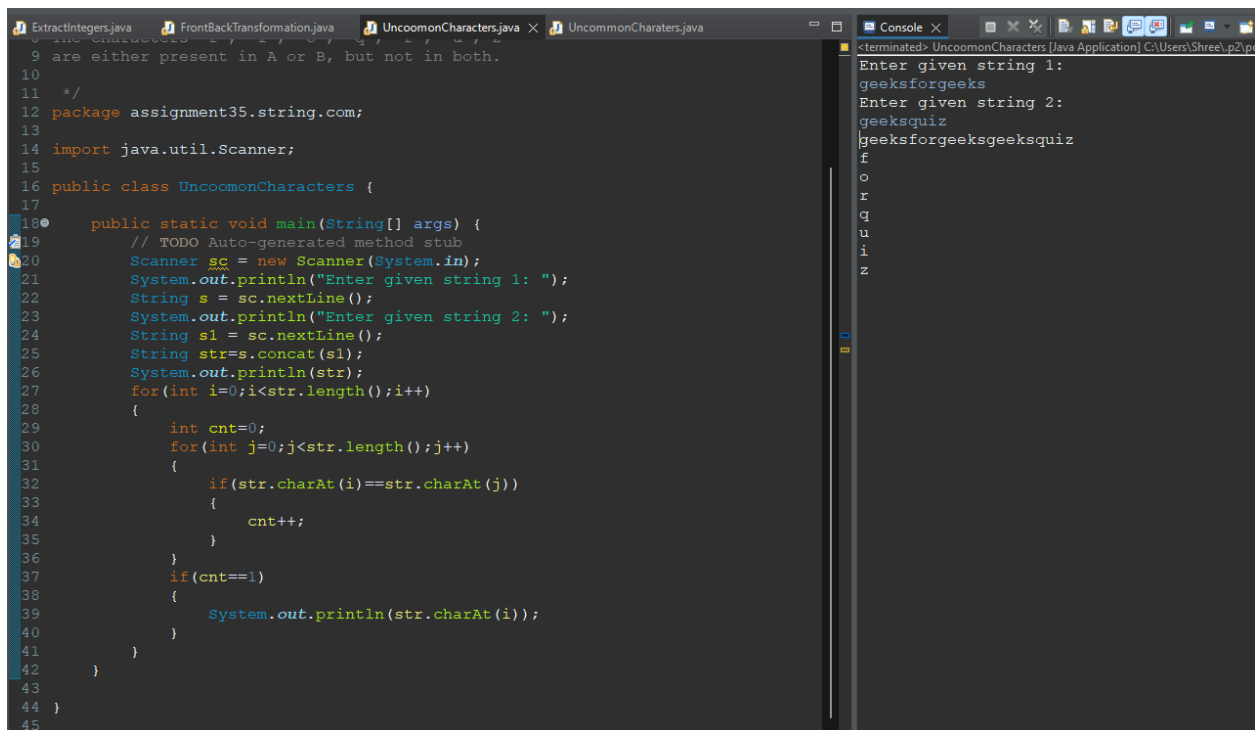
A = geeksforgeeks

B = geeksquiz

Output: fioqruz

Explanation:

The characters 'f', 'i', 'o', 'q', 'r', 'u', 'z' are either present in A or B, but not in both.



```
9 are either present in A or B, but not in both.
10
11 */
12 package assignment35.string.com;
13
14 import java.util.Scanner;
15
16 public class UncoomonCharacters {
17
18     public static void main(String[] args) {
19         // TODO Auto-generated method stub
20         Scanner sc = new Scanner(System.in);
21         System.out.println("Enter given string 1: ");
22         String s = sc.nextLine();
23         System.out.println("Enter given string 2: ");
24         String s1 = sc.nextLine();
25         String str=s.concat(s1);
26         System.out.println(str);
27         for(int i=0;i<str.length();i++)
28         {
29             int cnt=0;
30             for(int j=0;j<str.length();j++)
31             {
32                 if(str.charAt(i)==str.charAt(j))
33                 {
34                     cnt++;
35                 }
36             }
37             if(cnt==1)
38             {
39                 System.out.println(str.charAt(i));
40             }
41         }
42     }
43 }
44
45
```

```
<terminated> UncoomonCharacters [Java Application] C:\Users\Shree\p2\p
Enter given string 1:
geeksforgeeks
Enter given string 2:
geeksquiz
geeksforgeeksgeeksquiz
f
o
r
q
u
i
z
```

Q6. Remove common characters and concatenate (Oracle)

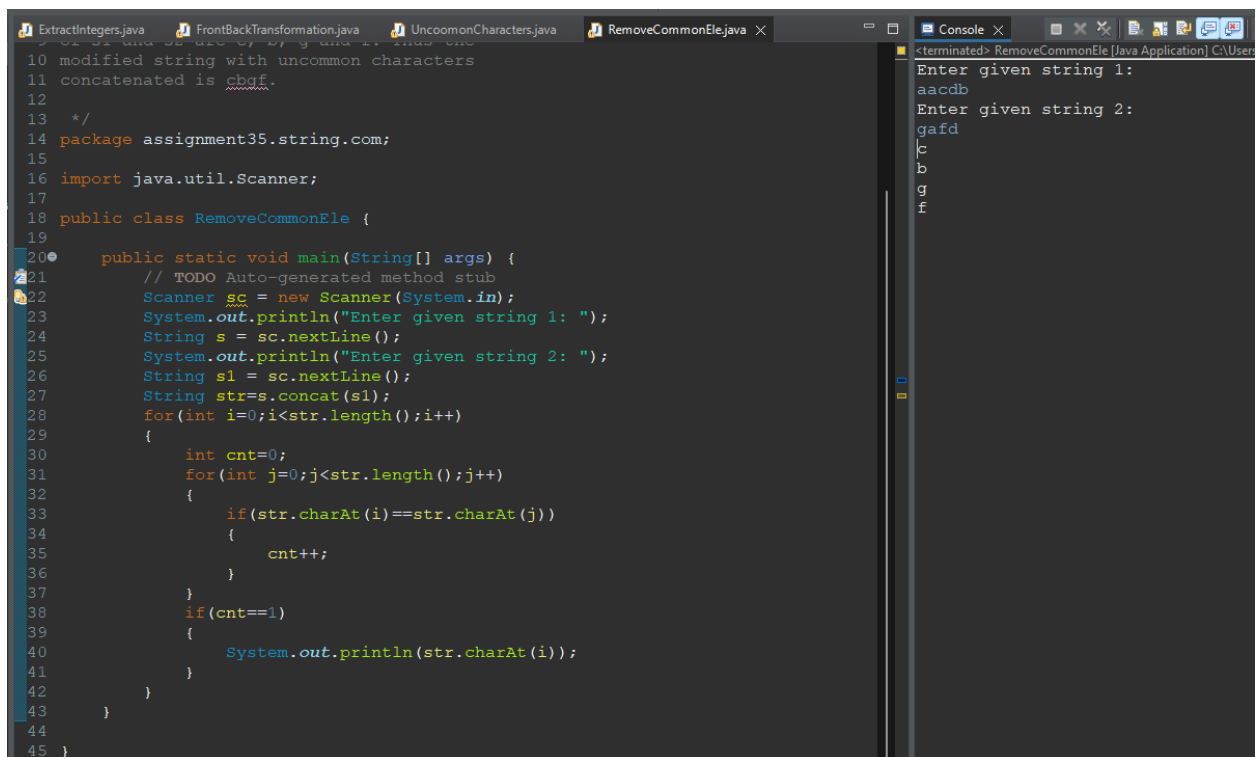
Input:

s1 = aacdb

s2 = gafd

Output: cbgf

Explanation: The common characters of s1 and s2 are: a, d. The uncommon characters of s1 and s2 are c, b, g and f. Thus the modified string with uncommon characters concatenated is cbgf.



```
10 modified string with uncommon characters
11 concatenated is cbgf.
12
13 */
14 package assignment35.string.com;
15
16 import java.util.Scanner;
17
18 public class RemoveCommonEle {
19
20     public static void main(String[] args) {
21         // TODO Auto-generated method stub
22         Scanner sc = new Scanner(System.in);
23         System.out.println("Enter given string 1: ");
24         String s = sc.nextLine();
25         System.out.println("Enter given string 2: ");
26         String s1 = sc.nextLine();
27         String str=s.concat(s1);
28         for(int i=0;i<str.length();i++)
29         {
30             int cnt=0;
31             for(int j=0;j<str.length();j++)
32             {
33                 if(str.charAt(i)==str.charAt(j))
34                 {
35                     cnt++;
36                 }
37             }
38             if(cnt==1)
39             {
40                 System.out.println(str.charAt(i));
41             }
42         }
43     }
44
45 }
```

Console

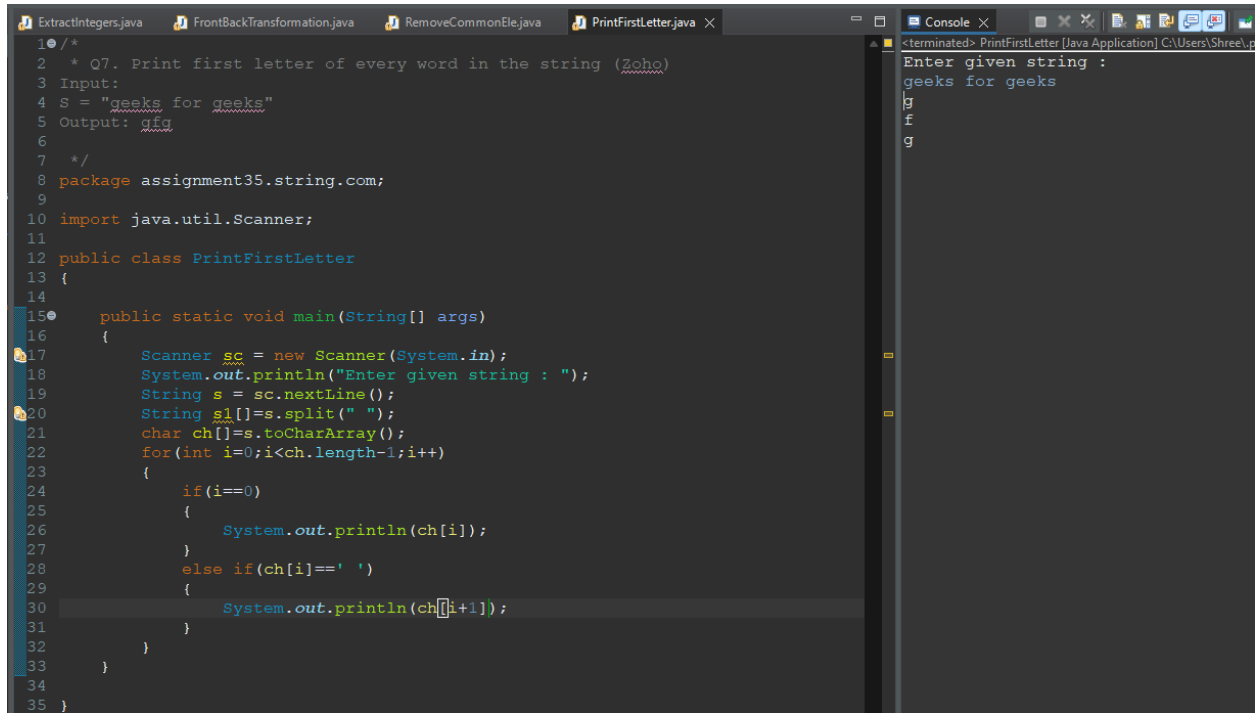
```
<terminated> RemoveCommonEle [Java Application] C:\Users\
Enter given string 1:
aacdb
Enter given string 2:
gafd
c
b
g
f
```

Q7. Print first letter of every word in the string (Zoho)

Input:

S = "geeks for geeks"

Output: gfg



The screenshot shows a Java IDE with a file named `PrintFirstLetter.java`. The code is as follows:

```
1 /*
2  * Q7. Print first letter of every word in the string (Zoho)
3  Input:
4  S = "geeks for geeks"
5  Output: gfg
6
7  */
8  package assignment35.string.com;
9
10 import java.util.Scanner;
11
12 public class PrintFirstLetter
13 {
14
15     public static void main(String[] args)
16     {
17         Scanner sc = new Scanner(System.in);
18         System.out.println("Enter given string : ");
19         String s = sc.nextLine();
20         String s1[] = s.split(" ");
21         char ch[] = s.toCharArray();
22         for(int i=0; i<ch.length-1; i++)
23         {
24             if(i==0)
25             {
26                 System.out.println(ch[i]);
27             }
28             else if(ch[i] == ' ')
29             {
30                 System.out.println(ch[i+1]);
31             }
32         }
33     }
34 }
35 }
```

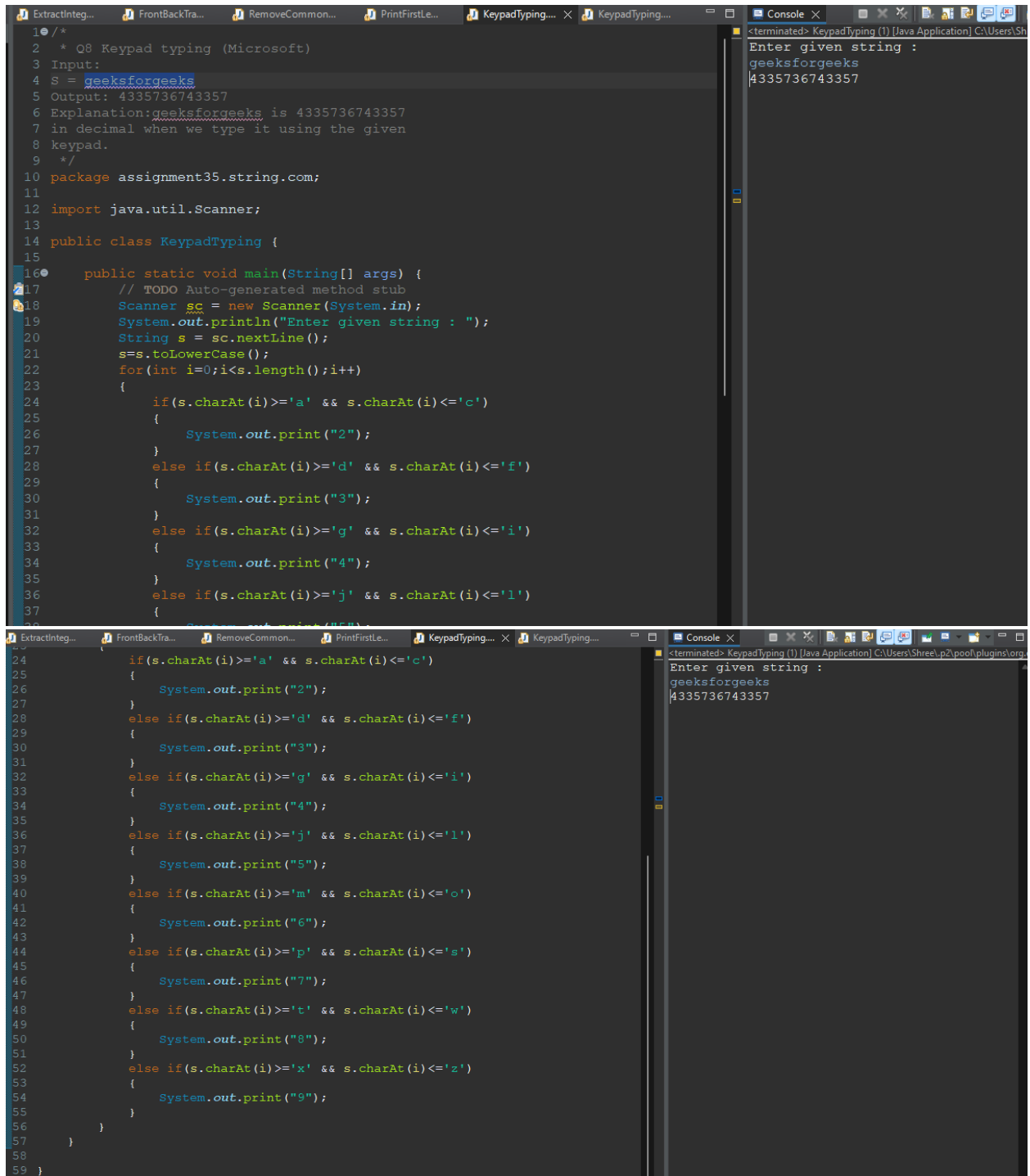
The console output on the right shows the execution of the program:

```
<terminated> PrintFirstLetter [Java Application] C:\Users\Shree\p
Enter given string :
geeks for geeks
g
f
g
```

Q8 Keypad typing (Microsoft)

Input: S = geeksforgeeks Output: 4335736743357

Explanation: geeksforgeeks is 4335736743357 in decimal when we type it using the given keypad.



```
10 /*
11  * Q8 Keypad typing (Microsoft)
12  * Input:
13  * S = geeksforgeeks
14  * Output: 4335736743357
15  * Explanation: geeksforgeeks is 4335736743357
16  * in decimal when we type it using the given
17  * keypad.
18  */
19 package assignment35.string.com;
20
21 import java.util.Scanner;
22
23 public class KeypadTyping {
24
25     public static void main(String[] args) {
26         // TODO Auto-generated method stub
27         Scanner sc = new Scanner(System.in);
28         System.out.println("Enter given string : ");
29         String s = sc.nextLine();
30         s = s.toLowerCase();
31         for(int i=0; i<s.length(); i++)
32         {
33             if(s.charAt(i) >= 'a' && s.charAt(i) <= 'c')
34             {
35                 System.out.print("2");
36             }
37             else if(s.charAt(i) >= 'd' && s.charAt(i) <= 'f')
38             {
39                 System.out.print("3");
40             }
41             else if(s.charAt(i) >= 'g' && s.charAt(i) <= 'i')
42             {
43                 System.out.print("4");
44             }
45             else if(s.charAt(i) >= 'j' && s.charAt(i) <= 'l')
46             {
47                 System.out.print("5");
48             }
49             else if(s.charAt(i) >= 'm' && s.charAt(i) <= 'o')
50             {
51                 System.out.print("6");
52             }
53             else if(s.charAt(i) >= 'p' && s.charAt(i) <= 's')
54             {
55                 System.out.print("7");
56             }
57             else if(s.charAt(i) >= 't' && s.charAt(i) <= 'w')
58             {
59                 System.out.print("8");
60             }
61             else if(s.charAt(i) >= 'x' && s.charAt(i) <= 'z')
62             {
63                 System.out.print("9");
64             }
65         }
66     }
67 }
```

Console Output:

```
<terminated> KeypadTyping (1) [Java Application] C:\Users\Sh...
Enter given string :
geeksforgeeks
4335736743357
```

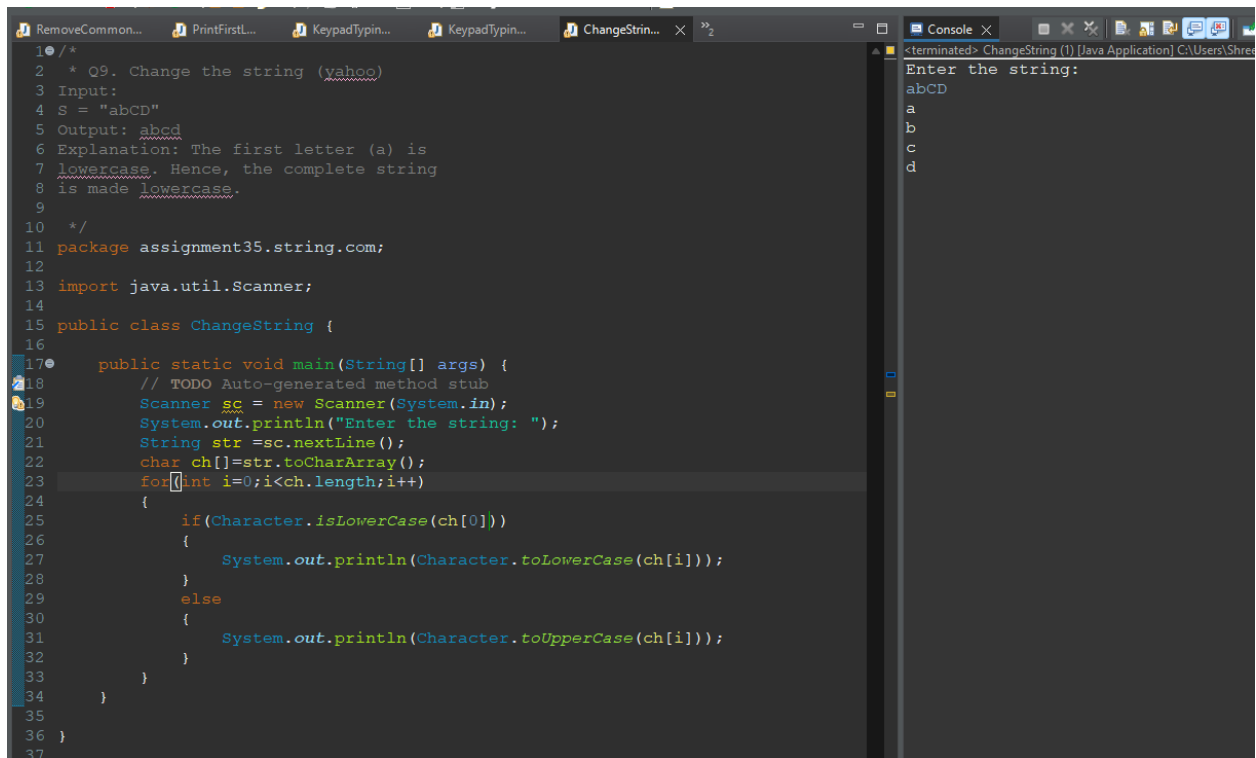

Q9. Change the string (yahoo)

Input:

S = "abCD"

Output: abcd

Explanation: The first letter (a) is lowercase. Hence, the complete string is made lowercase.



```
1  /*
2   * Q9. Change the string (yahoo)
3   Input:
4   S = "abCD"
5   Output: abcd
6   Explanation: The first letter (a) is
7   lowercase. Hence, the complete string
8   is made lowercase.
9
10  */
11  package assignment35.string.com;
12
13  import java.util.Scanner;
14
15  public class ChangeString {
16
17      public static void main(String[] args) {
18          // TODO Auto-generated method stub
19          Scanner sc = new Scanner(System.in);
20          System.out.println("Enter the string: ");
21          String str = sc.nextLine();
22          char ch[] = str.toCharArray();
23          for(int i=0; i<ch.length; i++)
24          {
25              if(Character.isLowerCase(ch[i]))
26              {
27                  System.out.println(Character.toLowerCase(ch[i]));
28              }
29              else
30              {
31                  System.out.println(Character.toUpperCase(ch[i]));
32              }
33          }
34      }
35  }
36  }
37
```

Console Output:

```
<terminated> ChangeString (1) [Java Application] C:\Users\Shree
Enter the string:
abCD
a
b
c
d
```

Q10. Good or Bad string (Amazon)

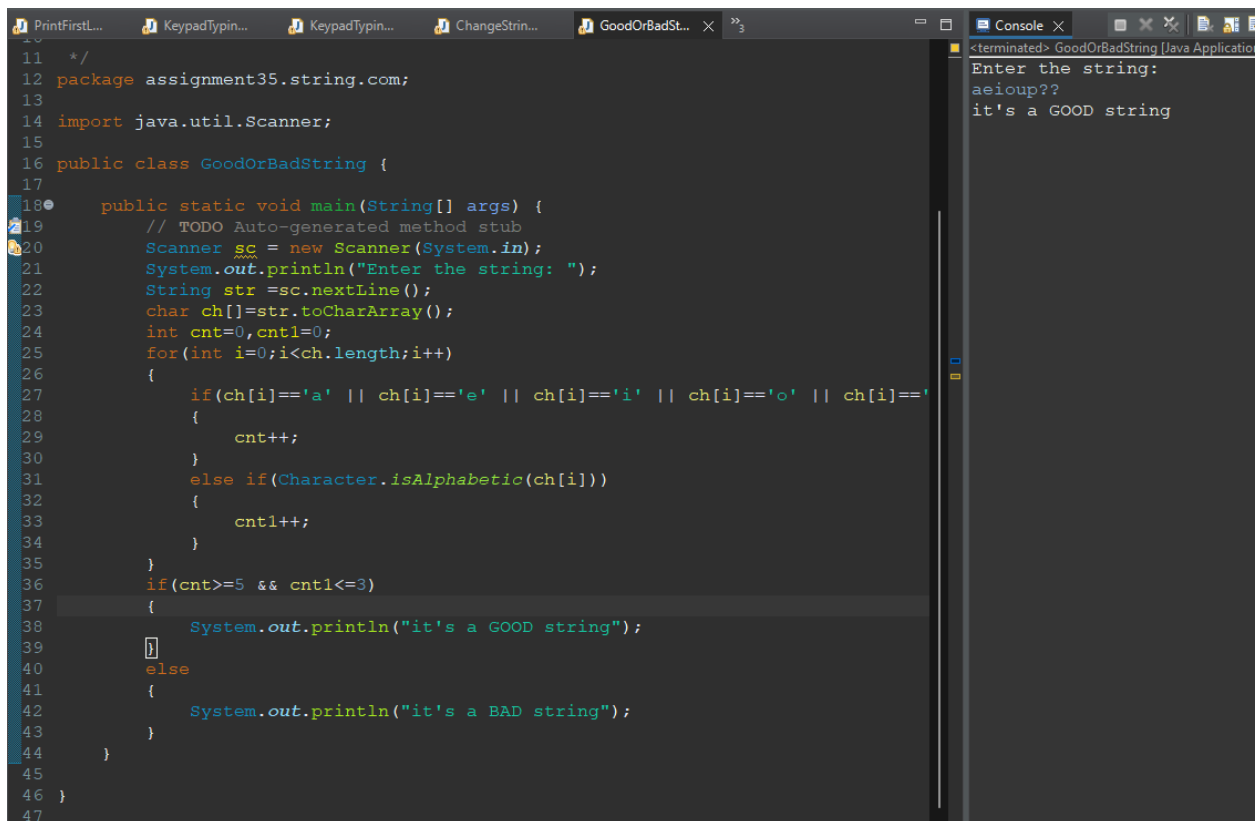
Input:

S = aeioup??

Output:

1

Explanation: The String doesn't contain more than 3 consonants or more than 5 vowels together. So, it's a GOOD string.



```
11  /*
12  package assignment35.string.com;
13
14  import java.util.Scanner;
15
16  public class GoodOrBadString {
17
18  public static void main(String[] args) {
19      // TODO Auto-generated method stub
20      Scanner sc = new Scanner(System.in);
21      System.out.println("Enter the string: ");
22      String str = sc.nextLine();
23      char ch[] = str.toCharArray();
24      int cnt = 0, cnt1 = 0;
25      for(int i = 0; i < ch.length; i++)
26      {
27          if(ch[i] == 'a' || ch[i] == 'e' || ch[i] == 'i' || ch[i] == 'o' || ch[i] == 'u')
28          {
29              cnt++;
30          }
31          else if(Character.isAlphabetic(ch[i]))
32          {
33              cnt1++;
34          }
35      }
36      if(cnt >= 5 && cnt1 <= 3)
37      {
38          System.out.println("it's a GOOD string");
39      }
40      else
41      {
42          System.out.println("it's a BAD string");
43      }
44  }
45
46  }
```

Console Output:

```
<terminated> GoodOrBadString [Java Application]
Enter the string:
aeioup??
it's a GOOD string
```

Q11. Twice Counter

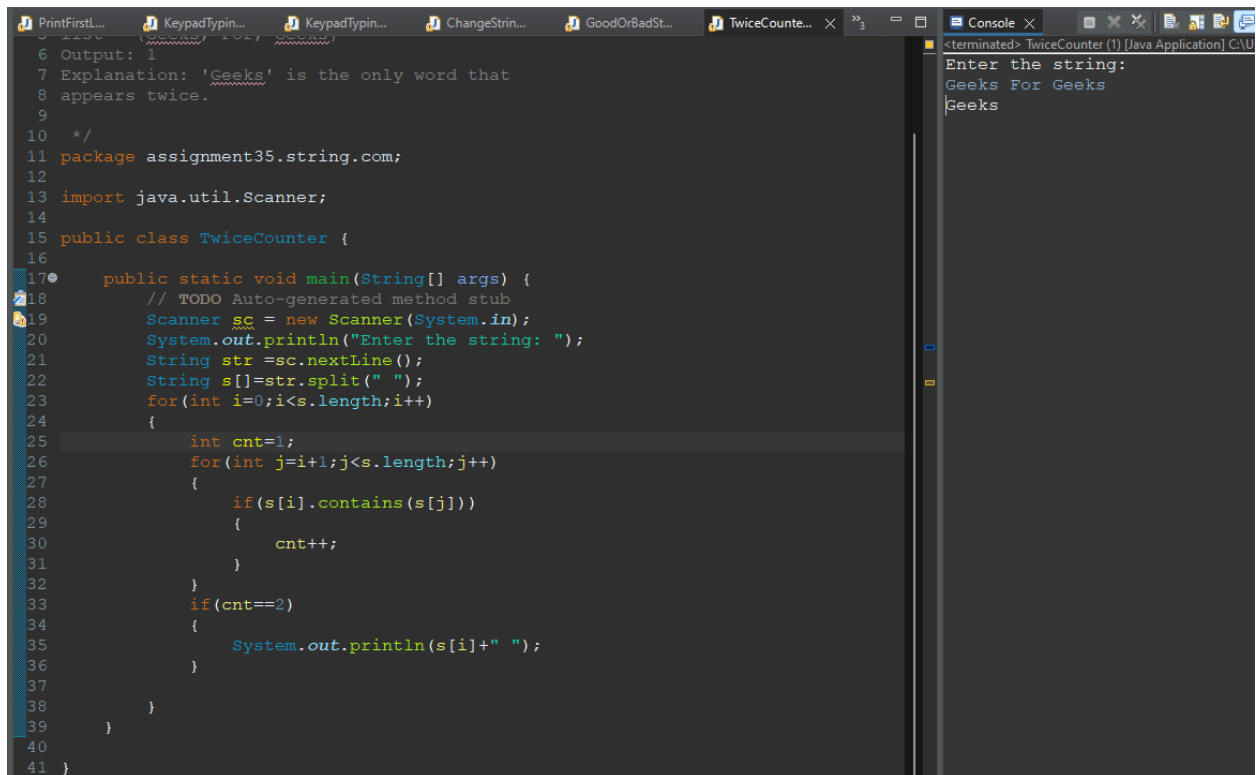
Input:

N = 3

list = {Geeks, For, Geeks}

Output: 1

Explanation: 'Geeks' is the only word that appears twice.



The screenshot shows a Java IDE with a code editor on the left and a console on the right. The code editor displays the following Java code:

```
10  */
11  package assignment35.string.com;
12
13  import java.util.Scanner;
14
15  public class TwiceCounter {
16
17  public static void main(String[] args) {
18      // TODO Auto-generated method stub
19      Scanner sc = new Scanner(System.in);
20      System.out.println("Enter the string: ");
21      String str = sc.nextLine();
22      String s[] = str.split(" ");
23      for(int i=0; i<s.length; i++)
24      {
25          int cnt=1;
26          for(int j=i+1; j<s.length; j++)
27          {
28              if(s[i].contains(s[j]))
29              {
30                  cnt++;
31              }
32          }
33          if(cnt==2)
34          {
35              System.out.println(s[i]+" ");
36          }
37      }
38  }
39  }
40
41 }
```

The console on the right shows the output of the program:

```
<terminated> TwiceCounter (1) [Java Application] C:\U
Enter the string:
Geeks For Geeks
Geeks
```

Q12. Rearrange a string

Example 1:

Input: S = "AC2BEW3"

Output: "ABCEW5"

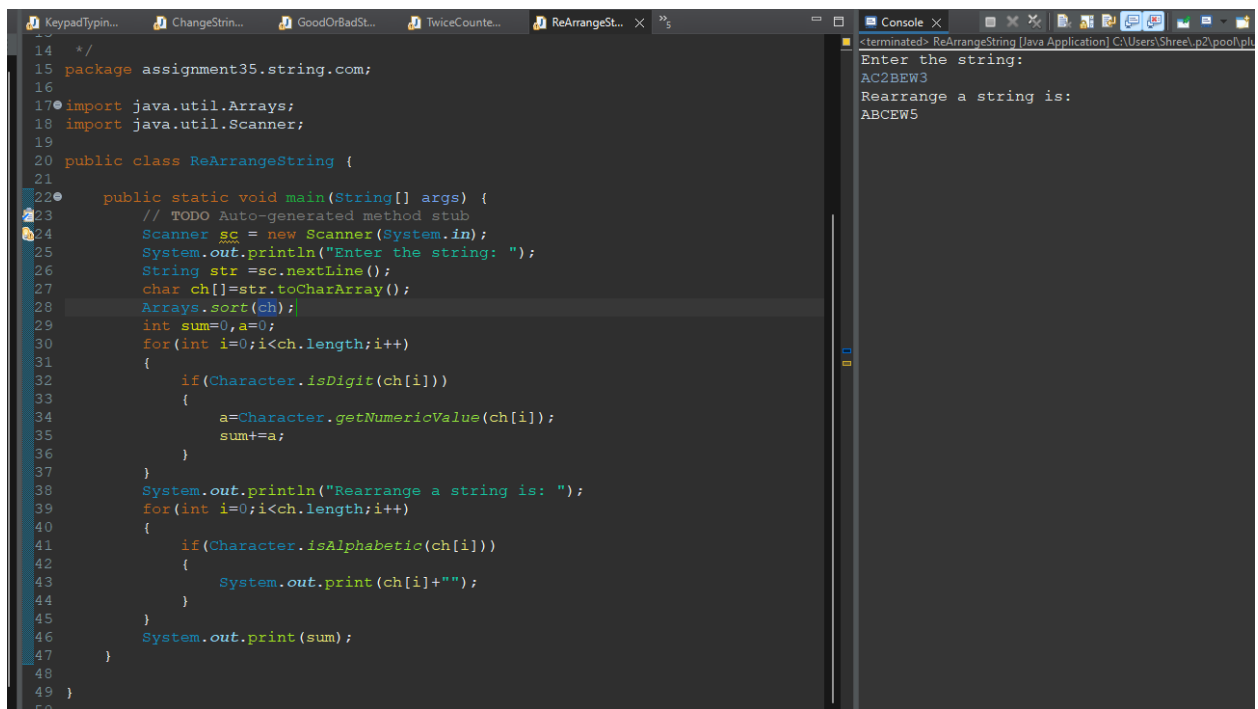
Explanation: $2 + 3 = 5$ and we print all alphabets in the lexicographical order.

Example 2:

Input: S = "ACCBA10D2EW30"

Output: "AABCCDEW6"

Explanation: $0+1+2+3 = 6$ and we print all alphabets in the lexicographical order.



```
14  */
15  package assignment35.string.com;
16
17  import java.util.Arrays;
18  import java.util.Scanner;
19
20  public class RearrangeString {
21
22      public static void main(String[] args) {
23          // TODO Auto-generated method stub
24          Scanner sc = new Scanner(System.in);
25          System.out.println("Enter the string: ");
26          String str = sc.nextLine();
27          char ch[] = str.toCharArray();
28          Arrays.sort(ch);
29          int sum = 0, a = 0;
30          for (int i = 0; i < ch.length; i++)
31          {
32              if (Character.isDigit(ch[i]))
33              {
34                  a = Character.getNumericValue(ch[i]);
35                  sum += a;
36              }
37          }
38          System.out.println("Rearrange a string is: ");
39          for (int i = 0; i < ch.length; i++)
40          {
41              if (Character.isAlphabetic(ch[i]))
42              {
43                  System.out.print(ch[i] + "");
44              }
45          }
46          System.out.print(sum);
47      }
48  }
49  }
```

```
<terminated> RearrangeString [Java Application] C:\Users\Shree\p2\pool\pl
Enter the string:
AC2BEW3
Rearrange a string is:
ABCEW5
```

Q13. Easy string (google)

Example 1:

Input: S = "aaABbBb"

Output: "3a3b"

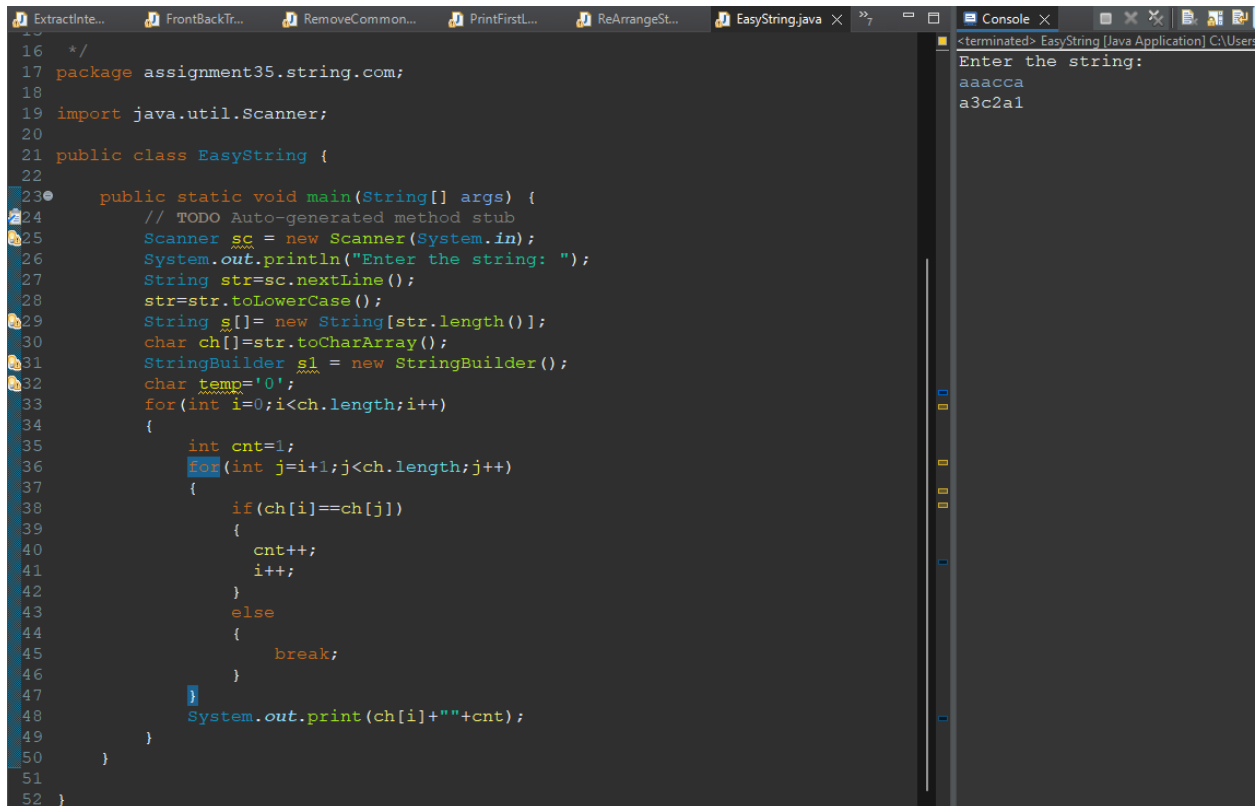
Explanation: As 'a' appears 3 times consecutively and 'b' also 3 times, and 'b' and 'B' considered as same.

Example 2:

Input: S = "aaacca"

Output: "3a2c1a"

Explanation: As 'a' appears 3 times consecutively and 'c' also 2 times, and then 'a' 1 time.



```
16  */
17  package assignment35.string.com;
18
19  import java.util.Scanner;
20
21  public class EasyString {
22
23      public static void main(String[] args) {
24          // TODO Auto-generated method stub
25          Scanner sc = new Scanner(System.in);
26          System.out.println("Enter the string: ");
27          String str=sc.nextLine();
28          str=str.toLowerCase();
29          String s[]= new String[str.length()];
30          char ch[]=str.toCharArray();
31          StringBuilder sb = new StringBuilder();
32          char temp='0';
33          for(int i=0;i<ch.length;i++)
34          {
35              int cnt=1;
36              for(int j=i+1;j<ch.length;j++)
37              {
38                  if(ch[i]==ch[j])
39                  {
40                      cnt++;
41                      i++;
42                  }
43                  else
44                  {
45                      break;
46                  }
47              }
48              System.out.print(ch[i]+" "+cnt);
49          }
50      }
51
52  }
```

Console

<terminated> EasyString [Java Application] C:\Users\...
Enter the string:
aaacca
a3c2a1

Q14. Special array reversal (google)

Example 1:

Input: S = "A&B"

Output: "B&A"

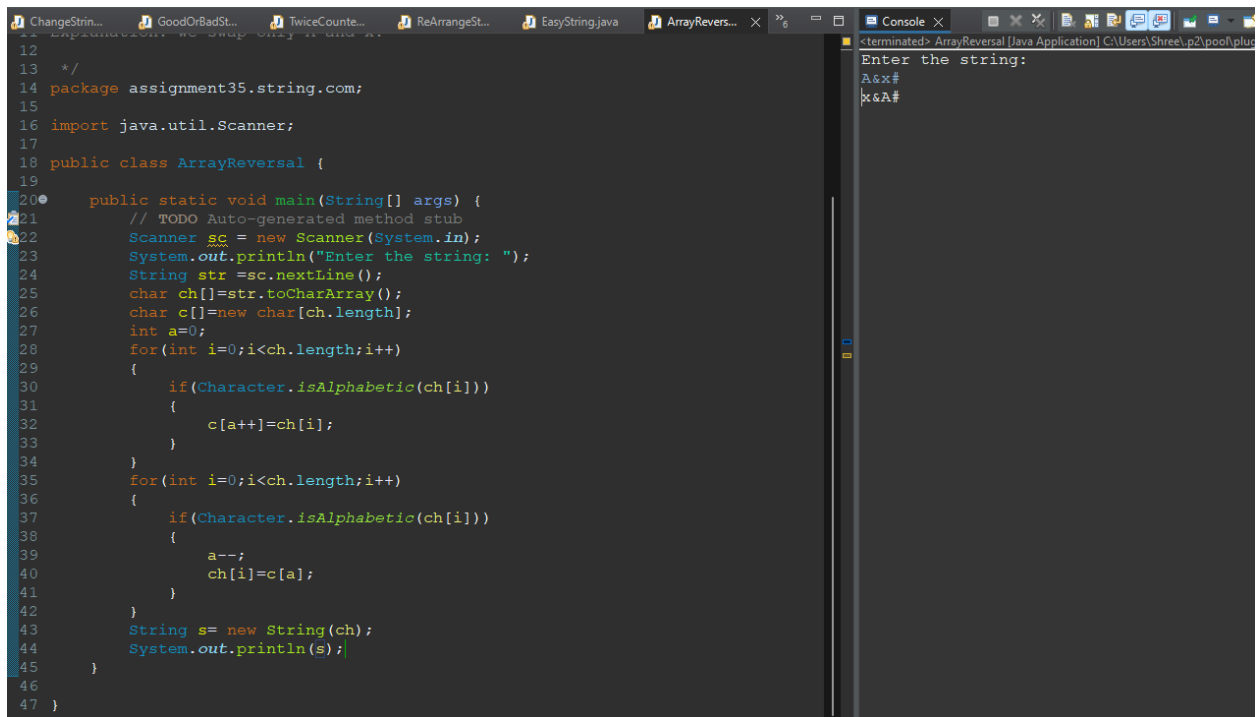
Explanation: As we ignore '&' and then reverse, so answer is "B&A".

Example 2:

Input: S = "A&x#"

Output: "x&A#"

Explanation: we swap only A and x.



```
12
13
14 package assignment35.string.com;
15
16 import java.util.Scanner;
17
18 public class ArrayReversal {
19
20     public static void main(String[] args) {
21         // TODO Auto-generated method stub
22         Scanner sc = new Scanner(System.in);
23         System.out.println("Enter the string: ");
24         String str = sc.nextLine();
25         char ch[] = str.toCharArray();
26         char c[] = new char[ch.length];
27         int a = 0;
28         for (int i = 0; i < ch.length; i++)
29         {
30             if (Character.isAlphabetic(ch[i]))
31             {
32                 c[a++] = ch[i];
33             }
34         }
35         for (int i = 0; i < ch.length; i++)
36         {
37             if (Character.isAlphabetic(ch[i]))
38             {
39                 a--;
40                 ch[i] = c[a];
41             }
42         }
43         String s = new String(ch);
44         System.out.println(s);
45     }
46
47 }
```

Console Output:

```
<terminated> ArrayReversal [Java Application] C:\Users\Shree\p2\pool\plug
Enter the string:
A&x#
x&A#
```

Q15. Find largest word in dictionary (MicroSoft)

Example 1:

Input: d = {"ale", "apple", "monkey", "plea"}

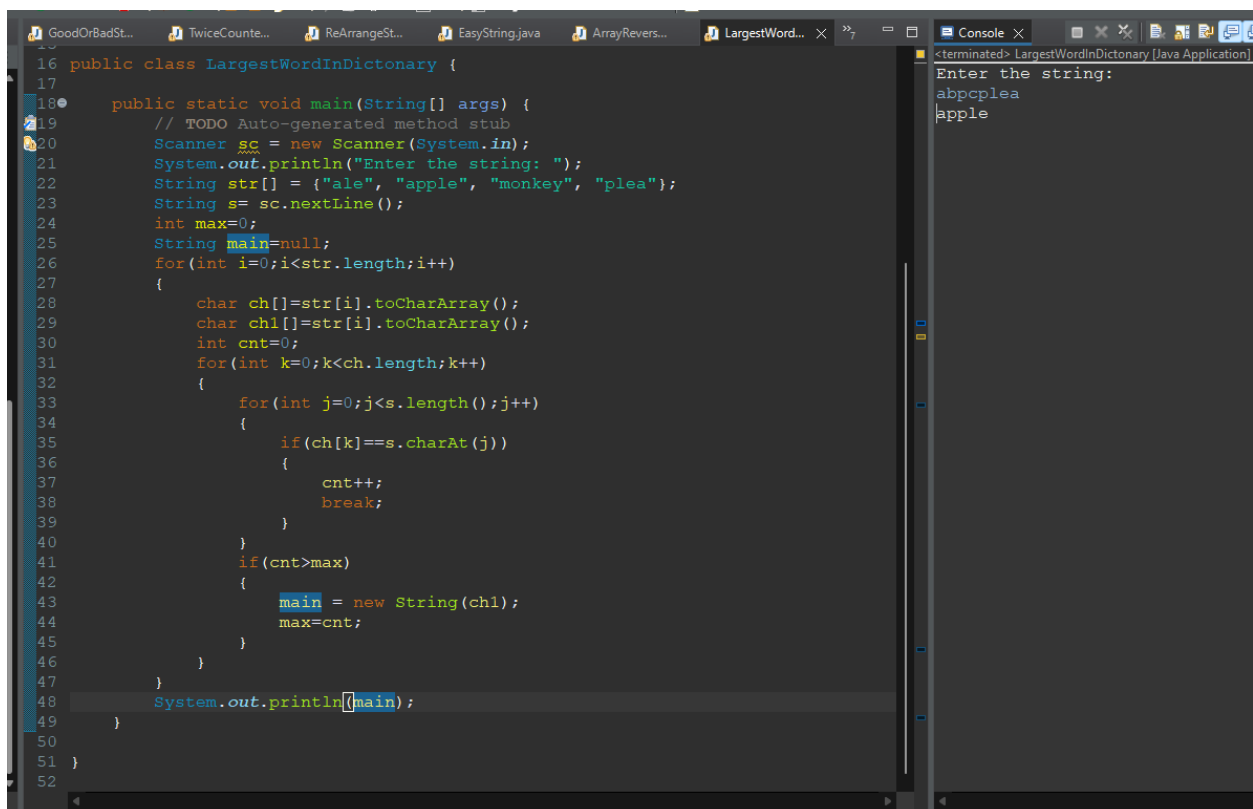
S = "abpcplea"

Output: "apple"

Explanation: After deleting "b", "c"

"a" S became "apple" which is present

in d.



```
16 public class LargestWordInDictionary {
17
18     public static void main(String[] args) {
19         // TODO Auto-generated method stub
20         Scanner sc = new Scanner(System.in);
21         System.out.println("Enter the string: ");
22         String str[] = {"ale", "apple", "monkey", "plea"};
23         String s = sc.nextLine();
24         int max=0;
25         String main=null;
26         for(int i=0;i<str.length;i++)
27         {
28             char ch[]=str[i].toCharArray();
29             char chl[]=str[i].toCharArray();
30             int cnt=0;
31             for(int k=0;k<ch.length;k++)
32             {
33                 for(int j=0;j<s.length();j++)
34                 {
35                     if(ch[k]==s.charAt(j))
36                     {
37                         cnt++;
38                         break;
39                     }
40                 }
41                 if(cnt>max)
42                 {
43                     main = new String(chl);
44                     max=cnt;
45                 }
46             }
47         }
48         System.out.println(main);
49     }
50 }
51 }
52 }
```

Console Window:

```
<terminated> LargestWordInDictionary [Java Application]
Enter the string:
abpcplea
apple
```

Q16. Odd Even Problem (oracle)

Given a string S of lowercase english characters, find out whether the summation of X and Y is

even or odd, where X is the count of characters which occupy even positions in english

alphabets and have positive even frequency, and Y is the count of characters which occupy odd

positions in english alphabets and have positive odd frequency.

Note: Positive means greater than zero.

Example 1:

Input: S = "abbbcc"

Output: "ODD"

Explanation: $X = 0$ and $Y = 1$ so $(X + Y)$ is

ODD. 'a' occupies 1st place(odd) in english

alphabets and its frequency is odd(1), 'b'

occupies 2nd place(even) but its frequency

is odd(3) so it doesn't get counted and 'c'

occupies 3rd place(odd) but its frequency

is even(2) so it also doesn't get counted.

Example 2:

Input: S = "nobitaa"

Output: "EVEN"

Explanation: $X = 0$ and $Y = 2$ so $(X + Y)$ is

EVEN.


```
ExtractInte... FrontBackTr... RemoveCommon... ReArrangeSt... EasyString.java LargestWord... EvenOddProbl... X";
1  /*
2   * Q16. Odd Even Problem (oracle)
3   Given a string S of lowercase english characters, find out whether the summation of X and Y is
4   even or odd, where X is the count of characters which occupy even positions in english
5   alphabets and have positive even frequency, and Y is the count of characters which occupy odd
6   positions in english alphabets and have positive odd frequency.
7   Note: Positive means greater than zero.
8   Example 1:
9   Input: S = "abbbcc"
10  Output: "ODD"
11  Explanation: X = 0 and Y = 1 so (X + Y) is
12  ODD. 'a' occupies 1st place(odd) in english
13  alphabets and its frequency is odd(1), 'b'
14  occupies 2nd place(even) but its frequency
15  is odd(3) so it doesn't get counted and 'c'
16  occupies 3rd place(odd) but its frequency
17  is even(2) so it also doesn't get counted.
18  Example 2:
19  Input: S = "nobitaa"
20  Output: "EVEN"
21  Explanation: X = 0 and Y = 2 so (X + Y) is
22  EVEN.
23
24  */
25  package assignment35.string.com;
26
27  import java.util.Scanner;
28
29  public class EvenOddProblem {
30
31  public static void main(String[] args) {
32      // TODO Auto-generated method stub
33      Scanner sc = new Scanner(System.in);
34      System.out.println("Enter the string: ");
35      String str=sc.nextLine();
36      char ch[]=str.toCharArray();
37      int x=0,y=0;
38      // TODO Auto-generated method stub
39  }
```

```
<terminated> EvenOddProblem [Java Appli
Enter the string:
abbbcc
odd
```

```
ExtractInte... FrontBackTr... RemoveCommon... ReArrangeSt... EasyString.java LargestWord... EvenOddProbl... X";
36  char ch[]=str.toCharArray();
37  int x=0,y=0;
38  for(int i=0;i<ch.length;i++)
39  {
40      int cnt=1;
41      for(int j=i+1;j<ch.length;j++)
42      {
43          if(ch[i]==ch[j])
44          {
45              cnt++;
46              ch[j]='\0';
47          }
48      }
49      if(ch[i]!='\0')
50      {
51          if(ch[i]%2==0 && cnt%2==0)
52          {
53              x++;
54          }
55          else
56          {
57              y++;
58          }
59      }
60  }
61  if((x+y)%2==0)
62  {
63      System.out.println("Even");
64  }
65  else
66  {
67      System.out.println("Odd");
68  }
69  }
70
71 }
```

```
<terminated> EvenOddProblem [Java Appli
Enter the string:
abbbcc
odd
```

Q17. Replace a word

Example 1:Input: S = "xxforxx xx for xx" ,oldW = "xx" ,ewW = "Geeks"

Output: "geeksforgeeks geeks for geeks"

Explanation: Replacing each "xx" with " Geeks" in S.

Example 2:

Input:

S = "india is the xx country"

oldW = "xx"

newW = "best"

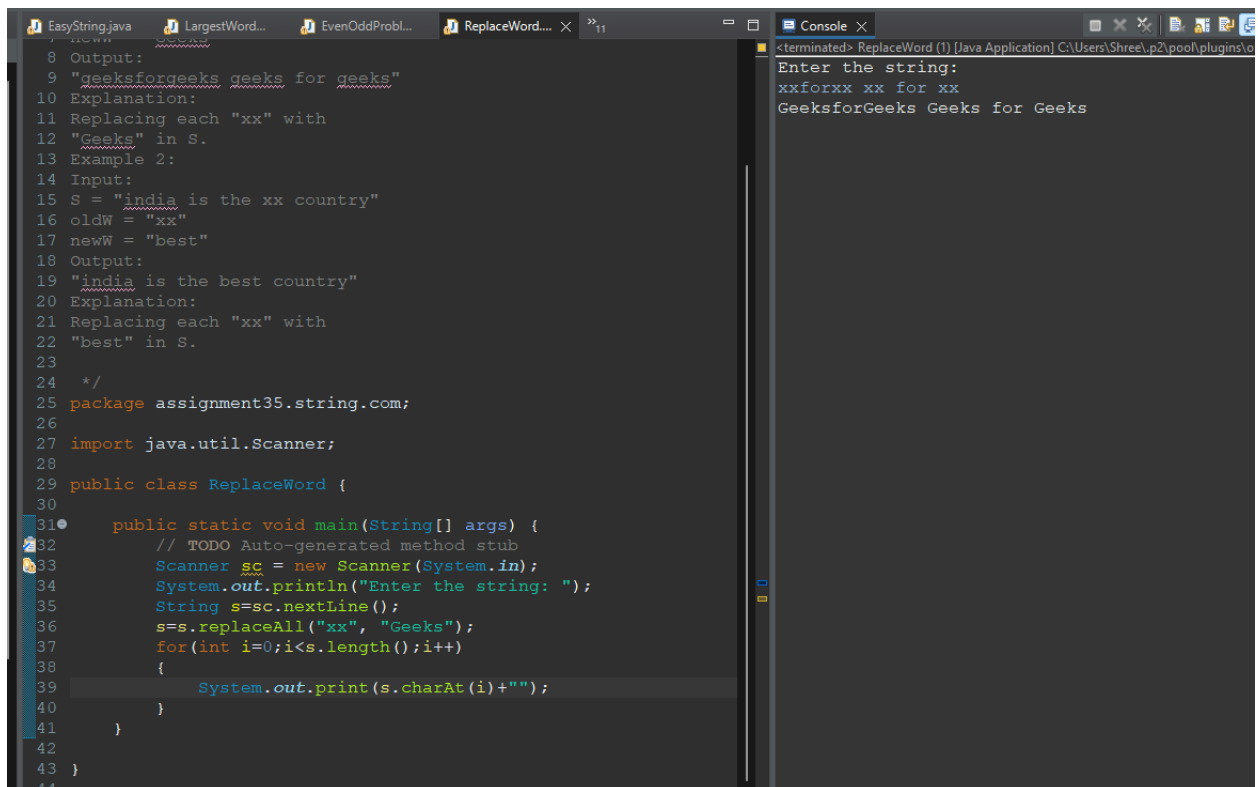
Output:

"india is the best country"

Explanation:

Replacing each "xx" with

"best" in S.



```
EasyString.java LargestWord... EvenOddProbl... ReplaceWord... x 22_11 Console x
8 Output:
9 "geeksforgeeks geeks for geeks"
10 Explanation:
11 Replacing each "xx" with
12 "Geeks" in S.
13 Example 2:
14 Input:
15 S = "india is the xx country"
16 oldW = "xx"
17 newW = "best"
18 Output:
19 "india is the best country"
20 Explanation:
21 Replacing each "xx" with
22 "best" in S.
23
24 */
25 package assignment35.string.com;
26
27 import java.util.Scanner;
28
29 public class ReplaceWord {
30
31     public static void main(String[] args) {
32         // TODO Auto-generated method stub
33         Scanner sc = new Scanner(System.in);
34         System.out.println("Enter the string: ");
35         String s=sc.nextLine();
36         s=s.replaceAll("xx", "Geeks");
37         for(int i=0;i<s.length();i++)
38         {
39             System.out.print(s.charAt(i)+"");
40         }
41     }
42
43 }
44
<terminated> ReplaceWord (1) [Java Application] C:\Users\Shree\p2\poo\plugins\o
Enter the string:
xxforxx xx for xx
GeeksforGeeks Geeks for Geeks
```

Q18. Ordering of strings (walmart)

You will be given N number of strings. You have to find the lexicographically smallest string and

the lexicographically largest string among these strings.

Example 1:

Input:

N = 3

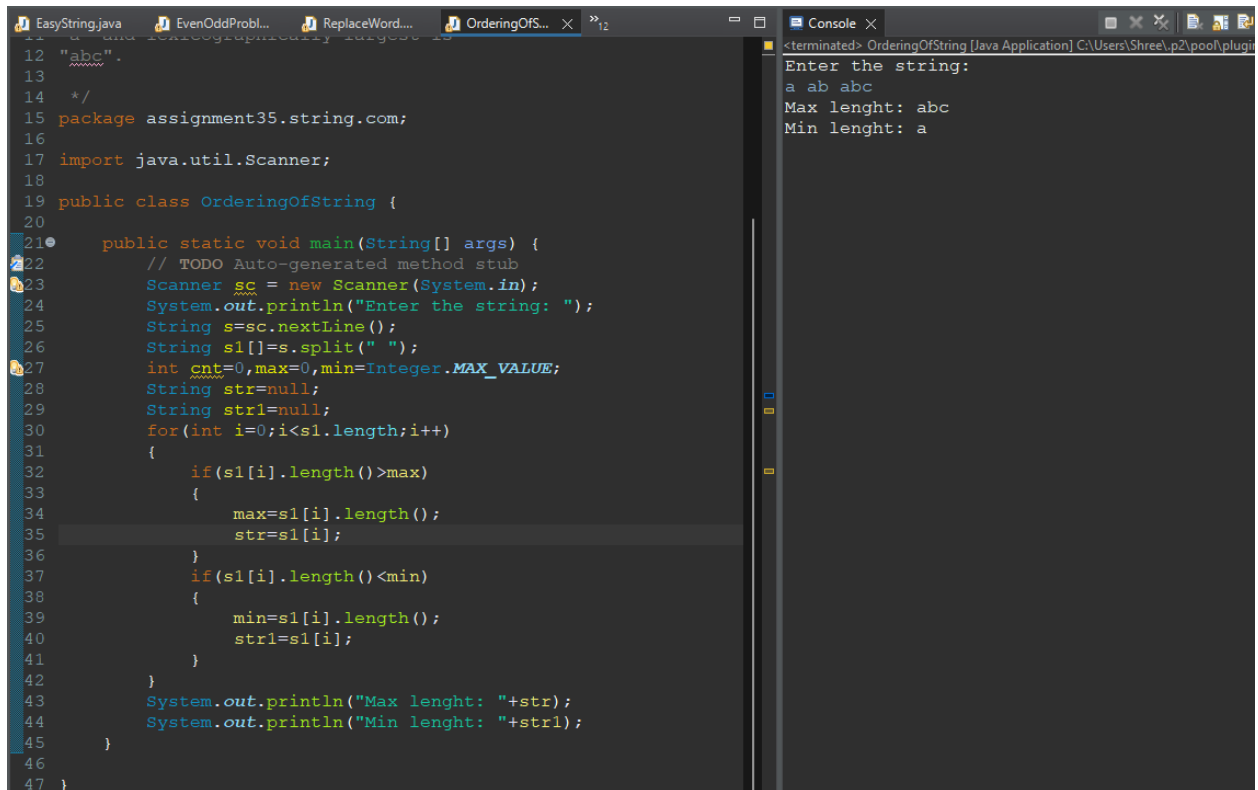
strings = a , ab , abc

Output: a abc

Explanation: Lexicographically smallest is

"a" and lexicographically largest is

"abc".



The screenshot shows a Java IDE with a file named 'OrderingOfString.java'. The code implements a solution to find the lexicographically smallest and largest strings from a given input. It uses a Scanner to read the input, splits it into an array of strings, and then iterates through the array to find the maximum and minimum strings based on their length and lexicographical order. The console output shows the input 'a ab abc' and the resulting 'Max lenght: abc' and 'Min lenght: a'.

```
11 // Lexicographically Largest and Smallest String
12 "abc".
13
14 */
15 package assignment35.string.com;
16
17 import java.util.Scanner;
18
19 public class OrderingOfString {
20
21     public static void main(String[] args) {
22         // TODO Auto-generated method stub
23         Scanner sc = new Scanner(System.in);
24         System.out.println("Enter the string: ");
25         String s=sc.nextLine();
26         String s1[]=s.split(" ");
27         int cnt=0,max=0,min=Integer.MAX_VALUE;
28         String str=null;
29         String str1=null;
30         for(int i=0;i<s1.length;i++)
31         {
32             if(s1[i].length()>max)
33             {
34                 max=s1[i].length();
35                 str=s1[i];
36             }
37             if(s1[i].length()<min)
38             {
39                 min=s1[i].length();
40                 str1=s1[i];
41             }
42         }
43         System.out.println("Max lenght: "+str);
44         System.out.println("Min lenght: "+str1);
45     }
46 }
47 }
```

Console Output:

```
<terminated> OrderingOfString [Java Application] C:\Users\Shree\p2\poo\plugin
Enter the string:
a ab abc
Max lenght: abc
Min lenght: a
```

Q19. Same characters in two strings

Given two strings A and B of equal length, find how many times the corresponding position in

the two strings hold exactly the same character. The comparison should not be case sensitive.

Example 1:

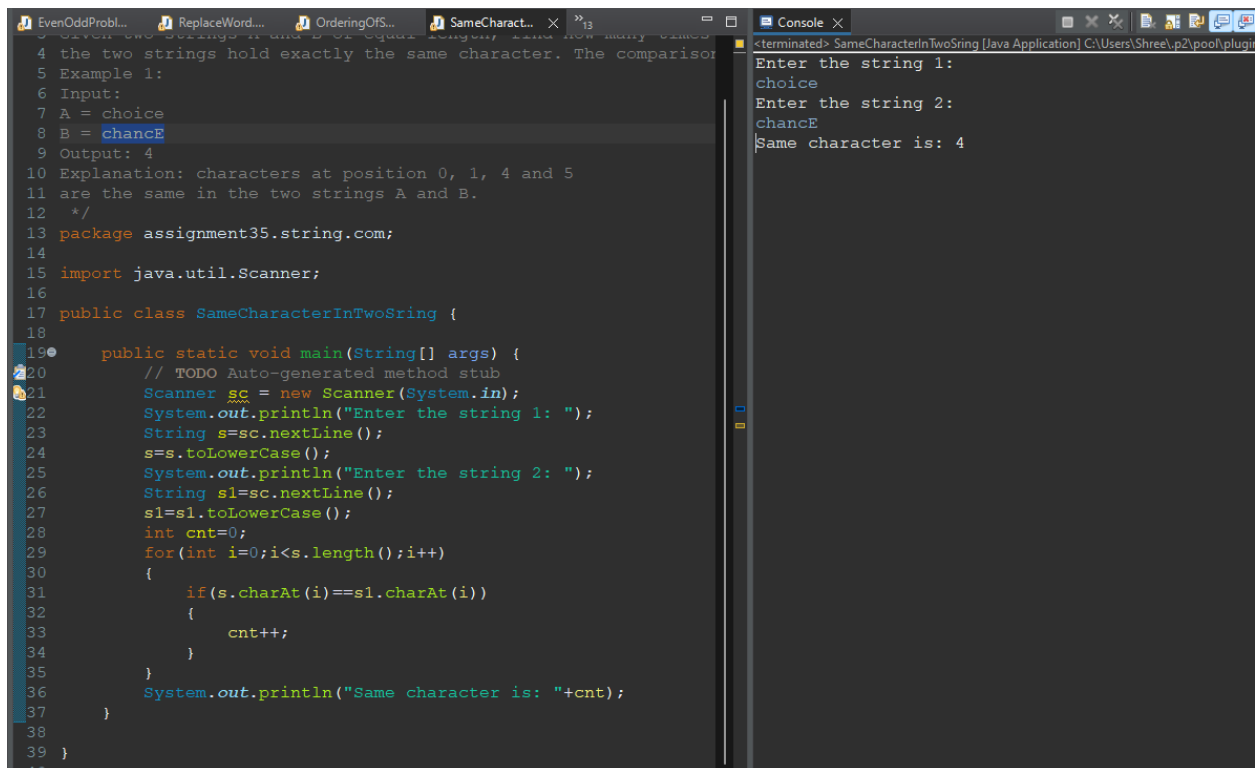
Input:

A = choice

B = chance

Output: 4

Explanation: characters at position 0, 1, 4 and 5 are the same in the two strings A and B.



```
4 the two strings hold exactly the same character. The comparison should not be case sensitive.
5 Example 1:
6 Input:
7 A = choice
8 B = chance
9 Output: 4
10 Explanation: characters at position 0, 1, 4 and 5 are the same in the two strings A and B.
11
12 */
13 package assignment35.string.com;
14
15 import java.util.Scanner;
16
17 public class SameCharacterInTwoString {
18
19     public static void main(String[] args) {
20         // TODO Auto-generated method stub
21         Scanner sc = new Scanner(System.in);
22         System.out.println("Enter the string 1: ");
23         String s=sc.nextLine();
24         s=s.toLowerCase();
25         System.out.println("Enter the string 2: ");
26         String s1=sc.nextLine();
27         s1=s1.toLowerCase();
28         int cnt=0;
29         for(int i=0;i<s.length();i++)
30         {
31             if(s.charAt(i)==s1.charAt(i))
32             {
33                 cnt++;
34             }
35         }
36         System.out.println("Same character is: "+cnt);
37     }
38 }
39
40
```

```
<terminated> SameCharacterInTwoString [Java Application] C:\Users\Shree\p2\pool\plugin
Enter the string 1:
choice
Enter the string 2:
chance
Same character is: 4
```

Q20. Maximum number of characters between any two same character

Given a string containing lower and uppercase alphabets, the task is to find the maximum number of characters between any two same characters in the string.

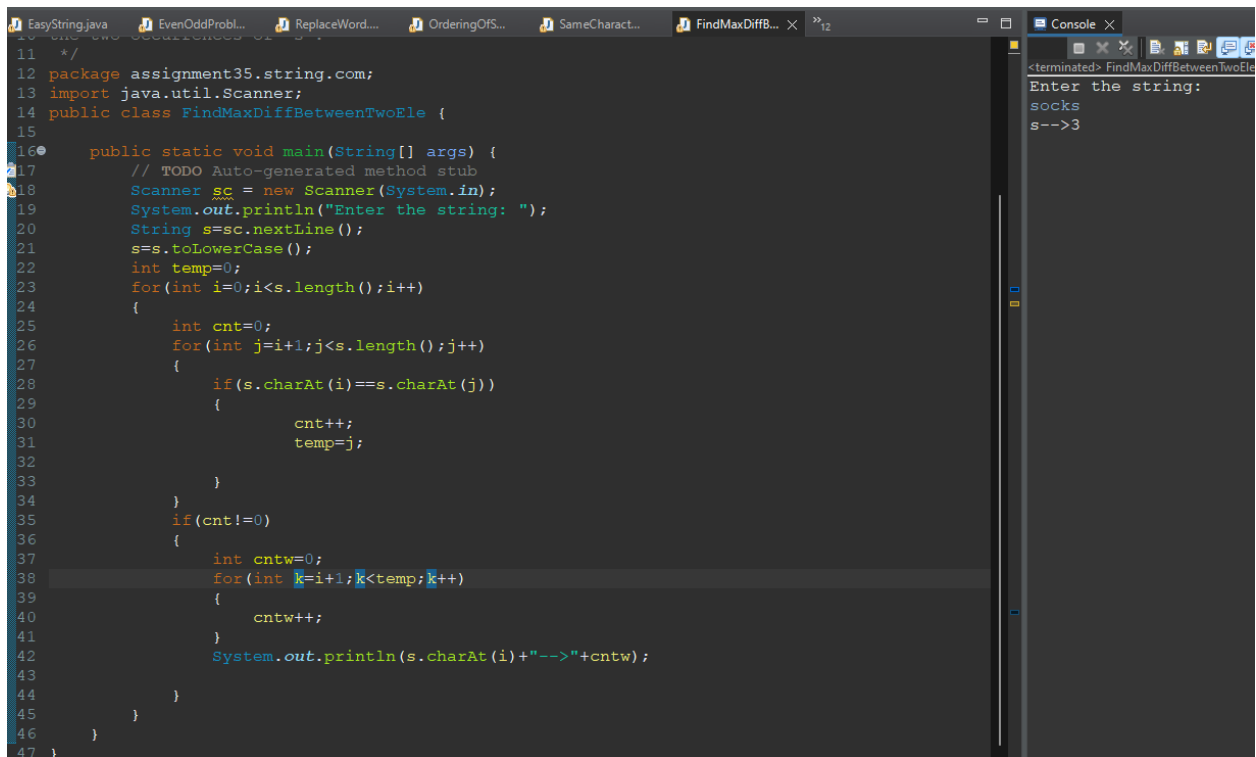
Example 1:

Input:

S = "socks"

Output: 3

Explanation: There are 3 characters between the two occurrences of 's'.



```
11  */
12  package assignment35.string.com;
13  import java.util.Scanner;
14  public class FindMaxDiffBetweenTwoEle {
15
16      public static void main(String[] args) {
17          // TODO Auto-generated method stub
18          Scanner sc = new Scanner(System.in);
19          System.out.println("Enter the string: ");
20          String s=sc.nextLine();
21          s=s.toLowerCase();
22          int temp=0;
23          for(int i=0;i<s.length();i++)
24          {
25              int cnt=0;
26              for(int j=i+1;j<s.length();j++)
27              {
28                  if(s.charAt(i)==s.charAt(j))
29                  {
30                      cnt++;
31                      temp=j;
32                  }
33              }
34              if(cnt!=0)
35              {
36                  int cntw=0;
37                  for(int k=i+1;k<temp;k++)
38                  {
39                      cntw++;
40                  }
41                  System.out.println(s.charAt(i)+"-->"+cntw);
42              }
43          }
44      }
45  }
46  }
47  }
```

Console

```
<terminated> FindMaxDiffBetweenTwoEle
Enter the string:
socks
s-->3
```

Q21. Last Match

Given two strings A and B, you need to find the last occurrence (1 based indexing) of string B in

string A.

Example 1:

Input:

A = abcdefghijklghifghsd

B = ghi

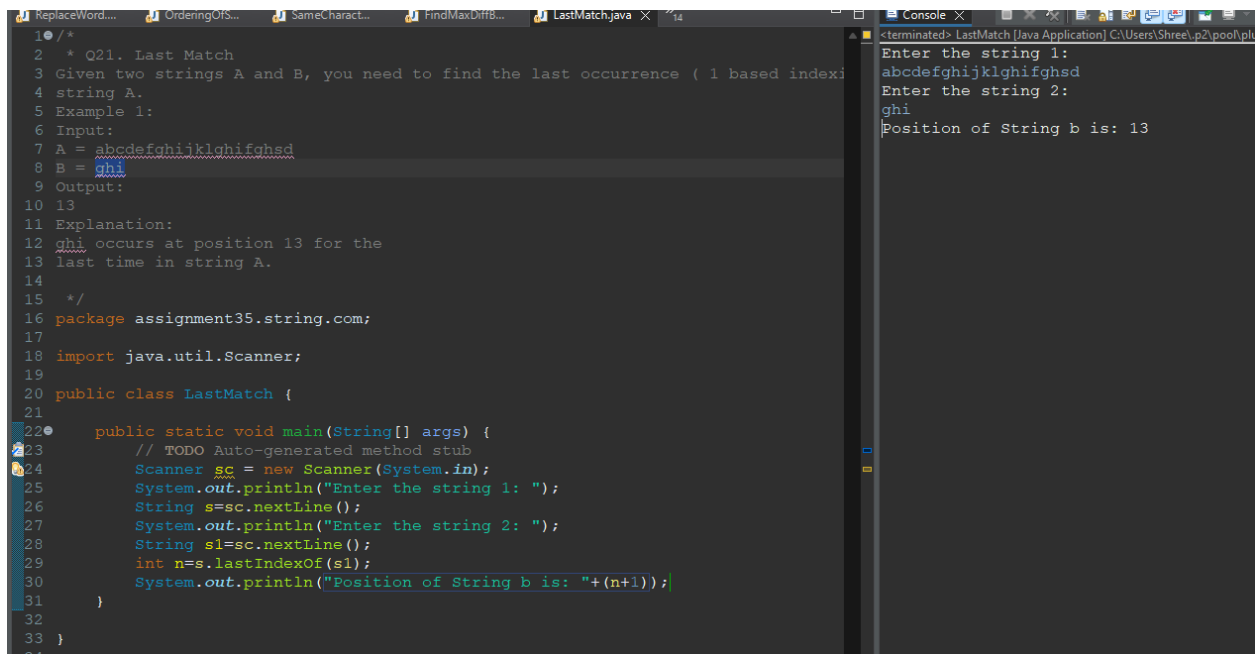
Output:

13

Explanation:

ghi occurs at position 13 for the

last time in string A.



The screenshot shows an IDE with a Java file named 'LastMatch.java' and a console window. The Java code is as follows:

```
1  /*
2   * Q21. Last Match
3   * Given two strings A and B, you need to find the last occurrence ( 1 based index)
4   * string A.
5   * Example 1:
6   * Input:
7   * A = abcdefghijklghifghsd
8   * B = ghi
9   * Output:
10  * 13
11  * Explanation:
12  * ghi occurs at position 13 for the
13  * last time in string A.
14  */
15
16 package assignment35.string.com;
17
18 import java.util.Scanner;
19
20 public class LastMatch {
21
22     public static void main(String[] args) {
23         // TODO Auto-generated method stub
24         Scanner sc = new Scanner(System.in);
25         System.out.println("Enter the string 1: ");
26         String s=sc.nextLine();
27         System.out.println("Enter the string 2: ");
28         String s1=sc.nextLine();
29         int n=s.lastIndexOf(s1);
30         System.out.println("Position of String b is: "+(n+1));
31     }
32 }
33
34
```

The console window shows the following output:

```
<terminated> LastMatch [Java Application] C:\Users\Shree\p2\pool\pl
Enter the string 1:
abcdefghijklghifghsd
Enter the string 2:
ghi
Position of String b is: 13
```

Q22. Difficulty of sentence

Given a sentence as a string S. Calculate difficulty of a given sentence.

Difficulty of sentence is defined as $5 * (\text{number of hard words}) + 3 * (\text{number of easy words})$.

A

word in the given string is considered hard if it has 4 consecutive consonants or the number of

consonants is more than the number of vowels. Else the word is easy.

Note: uppercase and lowercase characters are the same.

Example

Input: S = "Difficulty of sentence"

Output: 13

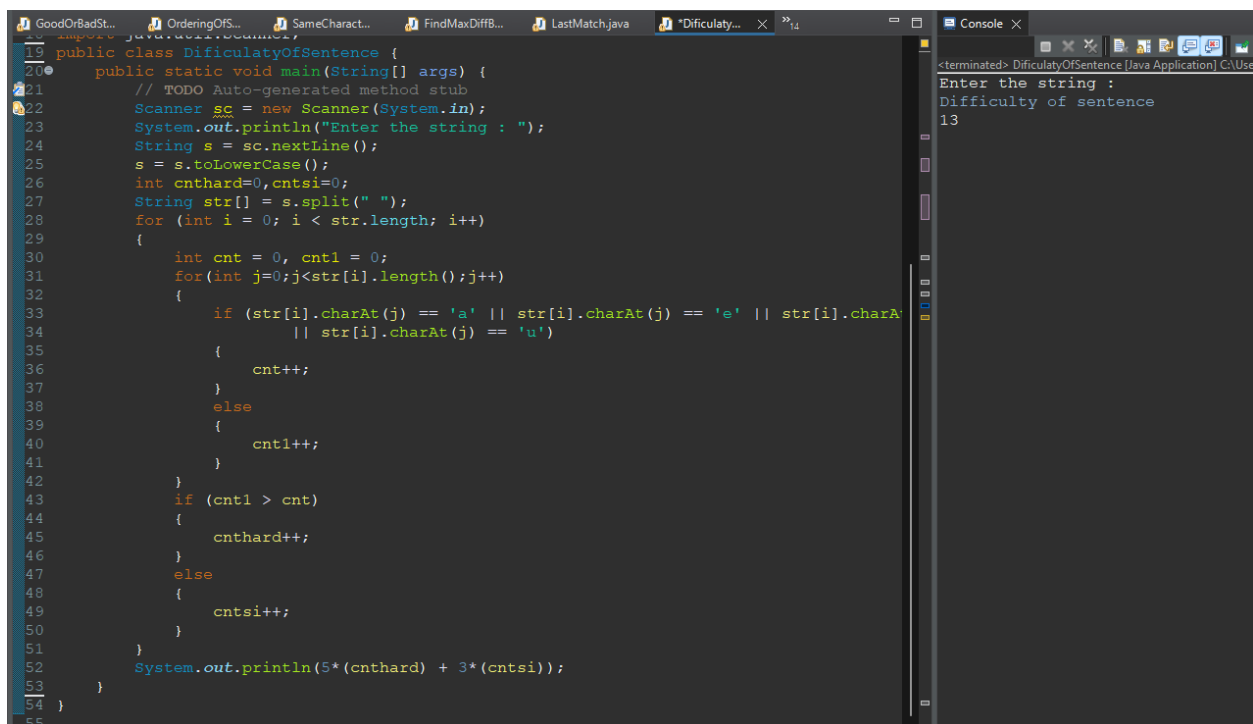
Explanation: 2 hard words + 1 easy word

Example

Input: S = "I am good"

Output: 9

Explanation: 3 easy word



```
19 public class DifficultyOfSentence {
20     public static void main(String[] args) {
21         // TODO Auto-generated method stub
22         Scanner sc = new Scanner(System.in);
23         System.out.println("Enter the string : ");
24         String s = sc.nextLine();
25         s = s.toLowerCase();
26         int cnthard=0, cntsi=0;
27         String str[] = s.split(" ");
28         for (int i = 0; i < str.length; i++)
29         {
30             int cnt = 0, cnt1 = 0;
31             for(int j=0;j<str[i].length();j++)
32             {
33                 if (str[i].charAt(j) == 'a' || str[i].charAt(j) == 'e' || str[i].charAt(j) == 'i' || str[i].charAt(j) == 'o' || str[i].charAt(j) == 'u')
34                     || str[i].charAt(j) == 'u')
35                 {
36                     cnt++;
37                 }
38                 else
39                 {
40                     cnt1++;
41                 }
42             }
43             if (cnt1 > cnt)
44             {
45                 cnthard++;
46             }
47             else
48             {
49                 cntsi++;
50             }
51         }
52         System.out.println(5*(cnthard) + 3*(cntsi));
53     }
54 }
55 }
```

Console Output:

```
<terminated> DifficultyOfSentence [Java Application] C:\Use
Enter the string :
Difficulty of sentence
13
```