Assignment-8

By: santhanam.L

1. Playing with String - I

Given a string array and non negative integer (n) apply the following rules.

1. Pick nth character from each String element in the String array and form a new String.
2. If nth character not available in a particular String in the array consider $ as the character. 3. Return the newly formed string.

Include a class UserMainCode with a static method formString which accepts the string and integer. The return type is the string formed based on rules.

Create a Class Main which would be used to accept the string and integer and call the static method present in UserMainCode.

Input and Output Format:

Input consists of a an integer which denotes the size of the array

followed by the array of strings and an integer (n).

Output consists of a string .

Refer sample output for formatting specifications.

Sample Input 1:

4

ABC

XYZ

EFG

MN

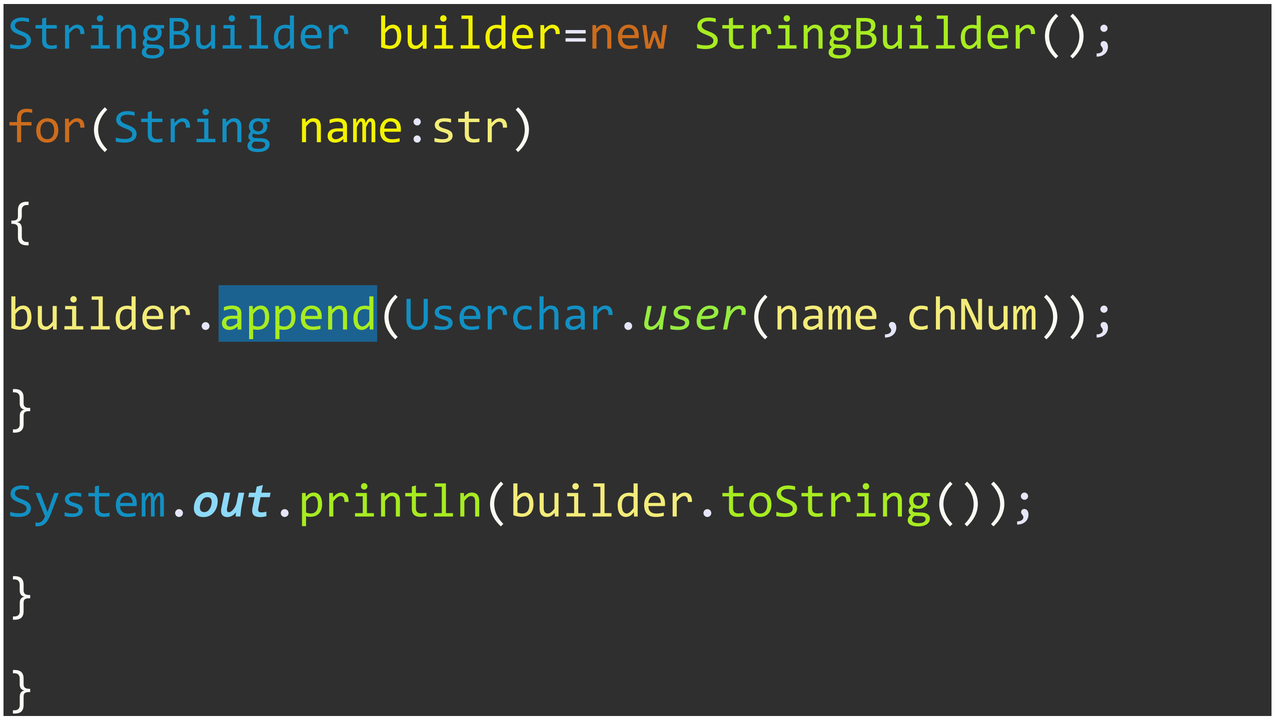
3

Sample Output 1:

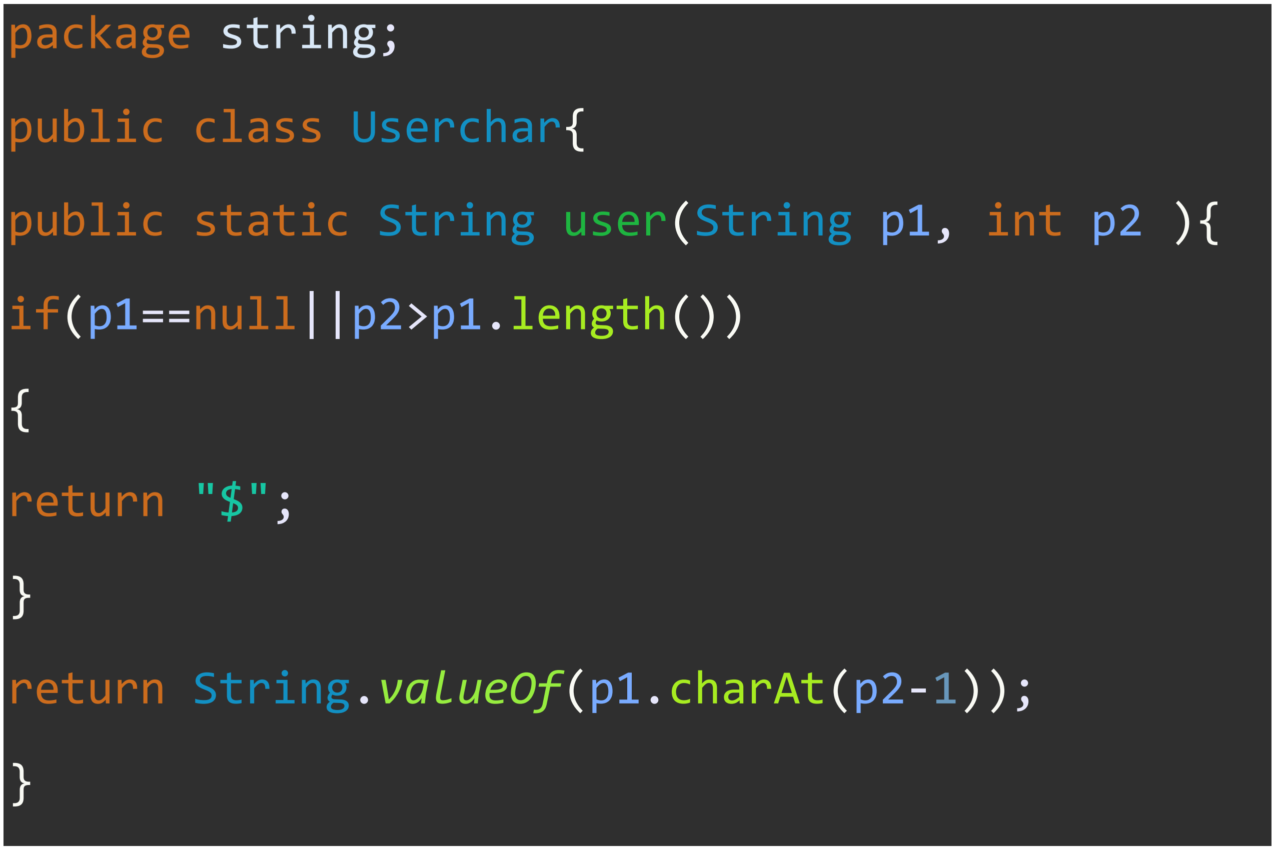
CZG$

UserMainCode:

```
package string;
import java.util.Scanner;
public class Nthchar {
public static void main(String[] args) {
Scanner scan=new Scanner(System.in);
System.out.println("Enter the +ve Number: ");
int num=scan.nextInt();
String [] str=new String[num];
for(int i=0;i<str.length;i++)
{
str[i]=scan.next();
}
System.out.println("chose char number");
int chNum=scan.nextInt();
```



```
StringBuilder builder=new StringBuilder();
for(String name:str)
{
builder.append(Userchar.user(name,chNum));
}
System.out.println(builder.toString());
}
}
```



```
package string;
public class Userchar{
public static String user(String p1, int p2 ){
if(p1==null||p2>p1.length())
{
return "$";
}
return String.valueOf(p1.charAt(p2-1));
}
}
```

Output:

```
Enter the +ve Number:

6

df

gfa

hjn

ytd

oie

zyl

chose char number

3

$andel
```

2. Reverse SubString

Given a string, startIndex and length, write a program to extract the substring from right to left. Assume the last character has index 0.

Include a class UserMainCode with a static method "reverseSubstring" that accepts 3 arguments and returns a string. The 1st argument corresponds to

the string, the second argument corresponds to the startIndex and the third argument corresponds to the length.

Create a class Main which would get a String and 2 integers as input and

call the static method reverseSubstring present in the UserMainCode.

Input and Output Format:

The first line of the input consists of a string.

The second line of the input consists of an integer that corresponds to the startIndex.

The third line of the input consists of an integer that corresponds to

the length of the substring.

Sample Input:

rajasthan

2

3

Sample Output:

Hts

Ans:

```java
package string;

import java.util.Scanner;

public class ReverseWord {

public static void main(String[] args) {

Scanner scan=new Scanner(System.in);

System.out.println("enter the Word :");

String word=scan.nextLine();

System.out.println("Enter the Start String Number: ");

int num1=scan.nextInt();

System.out.println("Enter the num of end String:");

int num2=scan.nextInt();

System.out.println(ReverseString.reverse(word,num1,num2));

}

}
```

```
package string;

public class ReverseString {

public static String reverse (String p1,int
p2,int p3)

{

StringBuffer builder =new StringBuffer(p1);

System.out.println(builder.reverse());

String Str=builder.substring(p2, p3);

return Str;

}

}
```

Output:

```
enter the Word :

santhanam

Enter the Start String Number:

2

Enter the num of end String:
```

```
5

manahtnas

nah
```

3. Fetching Middle Characters from String

Write a program to read a string of even length and to fetch two middle most characters from the input string and return it as string output.

Include a class UserMainCode with a static method getMiddleChars which accepts a string of even length as input . The return type is a string which should be the middle characters of the string.

Create a class Main which would get the input as a string and call the static method getMiddleChars present in the UserMainCode.

Input and Output Format:

Input consists of a string of even length.

Output is a string .

Refer sample output for formatting specifications.

Sample Input 1:

this

Sample Output 1:

Hi

Ans:

```
package string;

import java.util.Scanner;

public class FetchingString {

public static void main(String[] args) {

Scanner scan=new Scanner(System.in);

System.out.println("Enter the Word finishing Even:");

String Word=scan.nextLine();

System.out.println(WordFetch.Fetching(Word));

scan.close();

}

}
```

```
package string;

public class WordFetch {

public static String Fetching(String Str)

{
```

```
StringBuffer buffer=new StringBuffer();

if(Str.length()%2==0)

{

buffer.append(Str.substring((Str.length()/2)-1,
(Str.length()/2)+1));

}

return buffer.toString();

}

}
```

Output:

```
Enter the Word finishing Even:

this

hi

sandel

nd
```

4.String processing – Long + Short + Long

Obtain two strings S1,S2 from user as input. Your program should form a

string of “long+short+long”, with the shorter string inside of the

longer String.

Include a class UserMainCode with a static method getCombo which accepts two string variables. The return type is the string.

Create a Class Main which would be used to accept two Input strings and call the static method present in UserMainCode.

Input and Output Format:

Input consists of two strings with maximum size of 100 characters.

Output consists of an string.

Refer sample output for formatting specifications.

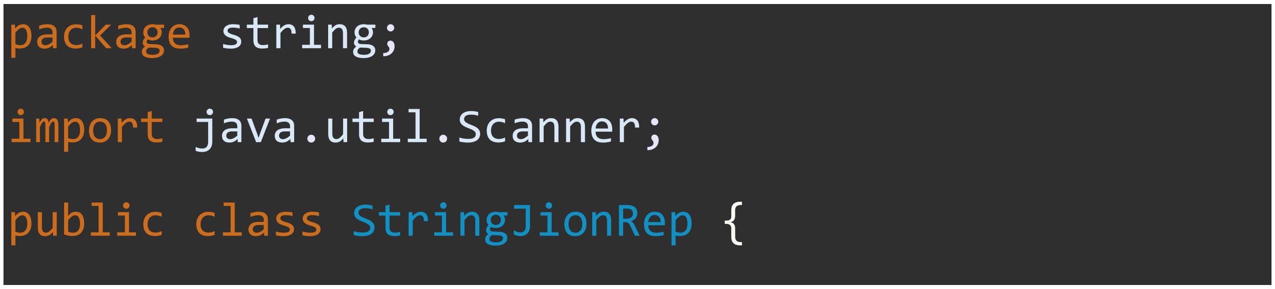
Sample Input 1:

Hello

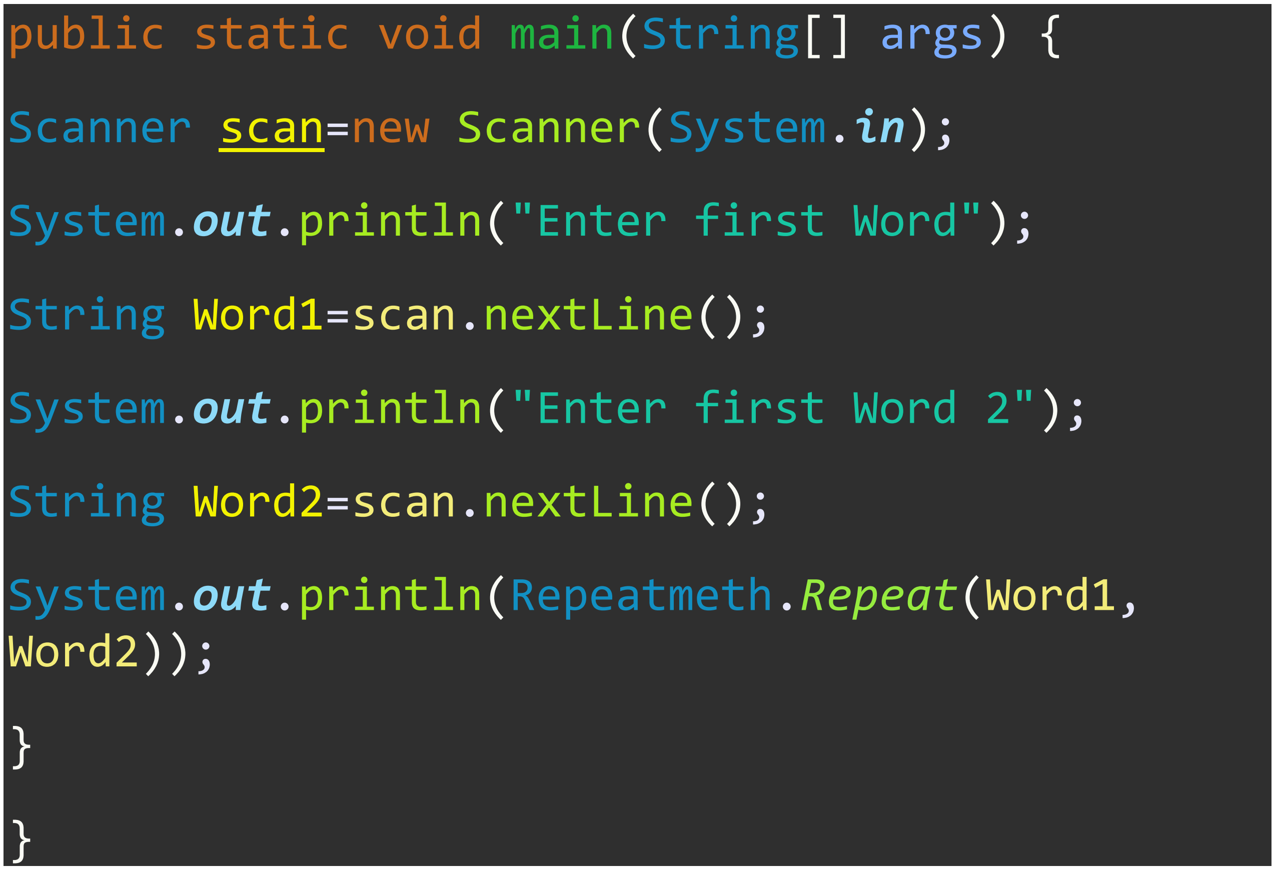
Hi

Sample Output 1:

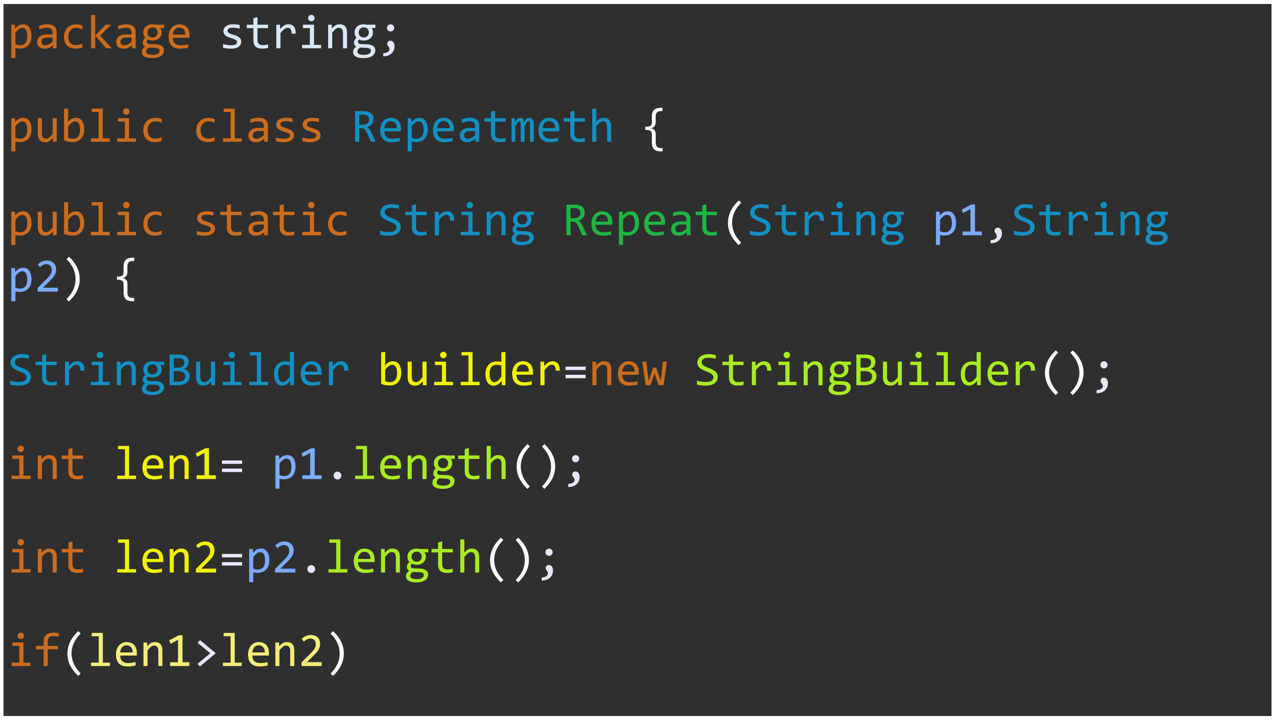
HelloHiHello Ans:



```
package string;

import java.util.Scanner;

public class StringJionRep {
```



```
public static void main(String[] args) {
Scanner scan=new Scanner(System.in);
System.out.println("Enter first Word");
String Word1=scan.nextLine();
System.out.println("Enter first Word 2");
String Word2=scan.nextLine();
System.out.println(Repeatmeth.Repeat(Word1,
Word2));
}
}
```



```
package string;
public class Repeatmeth {
public static String Repeat(String p1,String
p2) {
StringBuilder builder=new StringBuilder();
int len1= p1.length();
int len2=p2.length();
if(len1>len2)
{
builder.append(p1).append(p2).append(p1);
}
else
{
builder.append(p2).append(p1).append(p2);
}
return builder.toString();
}
}
```

Output:

```
package string;
public class Repeatmeth {
public static String Repeat(String p1,String
p2) {
StringBuilder builder=new StringBuilder();
int len1= p1.length();
int len2=p2.length();
if(len1>len2)

{

builder.append(p1).append(p2).append(p1);

}

else

{

builder.append(p2).append(p1).append(p2);

}

return builder.toString();

}

}
```

5.Strings Processing - Replication

Write a program to read a string and also a number N. Return the replica of original string for n given time.

Include a class UserMainCode with a static method repeatString which accepts the the string and the number n. The return type is the string based on the problem statement.

Create a Class Main which would be used to accept the string and integer

and call the static method present in UserMainCode.

Input and Output Format:

Input consists of a string and integer.

Output consists of a string.

Refer sample output for formatting specifications.

Sample Input 1:

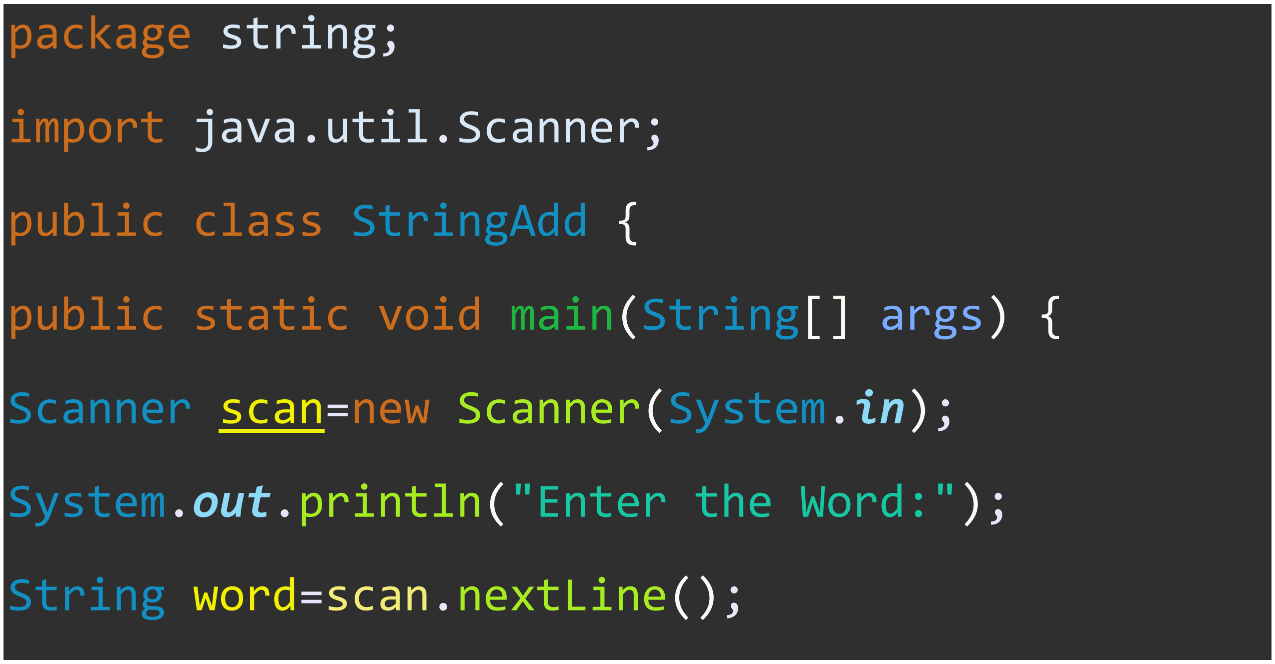
Lily

2

Sample Output 1:

LilyLily

Ans:



```
package string;

import java.util.Scanner;

public class StringAdd {

public static void main(String[] args) {

Scanner scan=new Scanner(System.in);

System.out.println("Enter the Word:");

String word=scan.nextLine();
System.out.println("Enter the want to repeatof Number:");

int num=scan.nextInt();

String res=ValidString(word,num);

System.out.println(res);

}

private static String ValidString(String p1,int p2) {

StringBuilder builder=new StringBuilder();

String name=p1;

int rep=p2;

for(int i=0;i<rep;i++)

{

builder.append(name +"  ");

}

return builder.toString();

}
```

Output:

```
Enter the Word:

hari

Enter the want to repeatof Number:

3

hari hari hari
```

6. Flush Characters

Write a program to read a string from the user and remove all the alphabets and spaces from the String, and only store special characters and digit in the output String. Print the output string.

Include a class UserMainCode with a static method getSpecialChar which accepts a string. The return type (String) should return the character removed string.

Create a Class Main which would be used to accept a string and call the static method present in UserMainCode.

Input and Output Format:

Input consists of a strings.

Output consists of an String (character removed string).

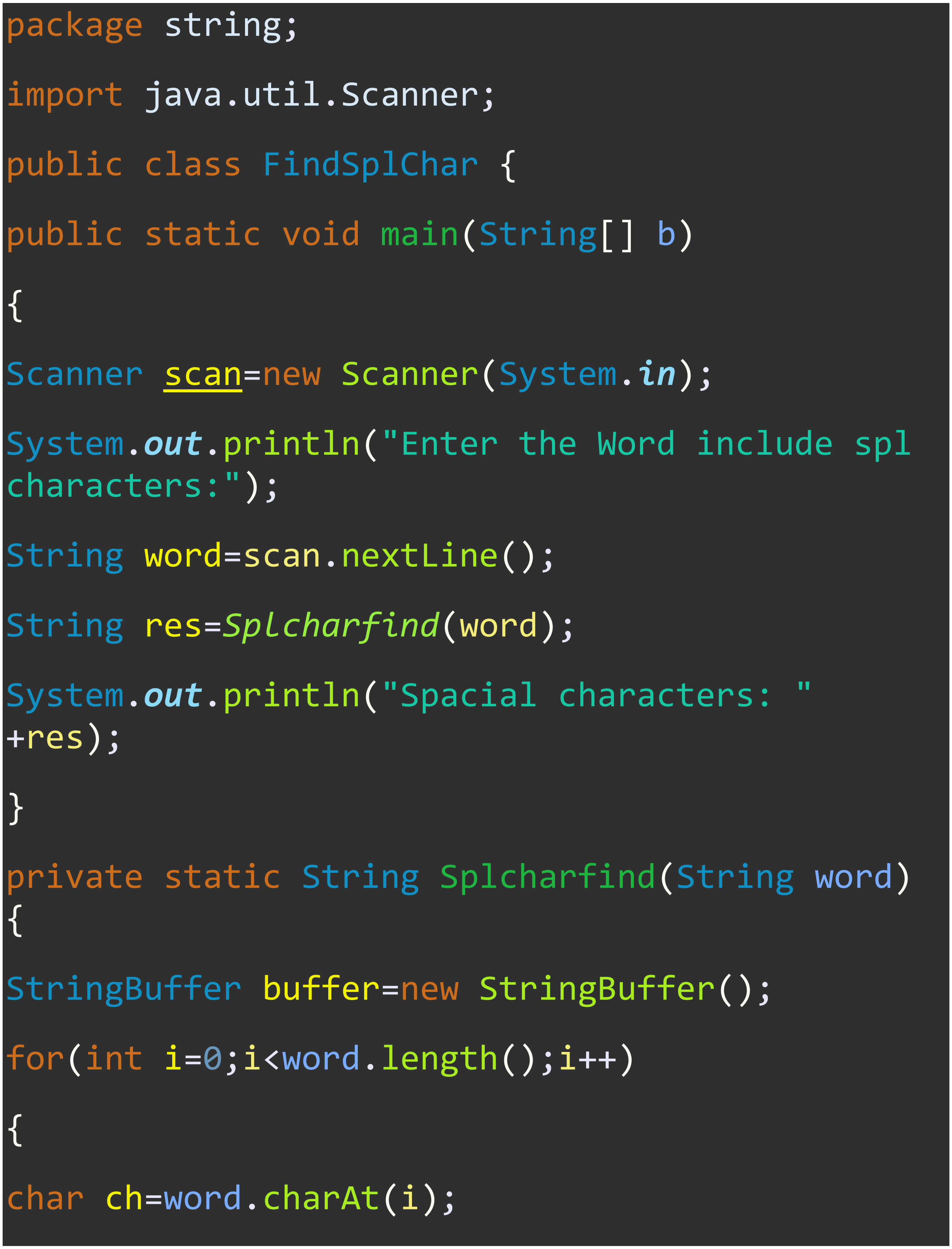
Refer sample output for formatting specifications.

Sample Input : cogniz$#45Ant

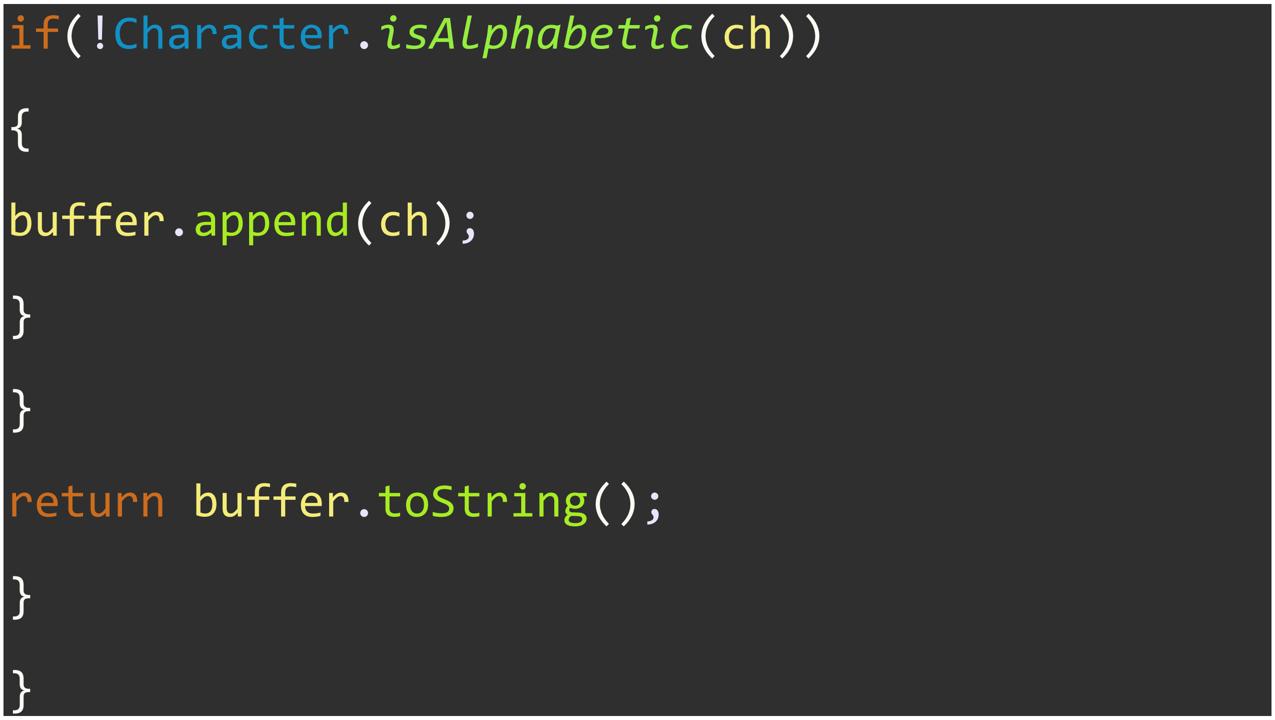
Sample Output :

$#45

Ans:

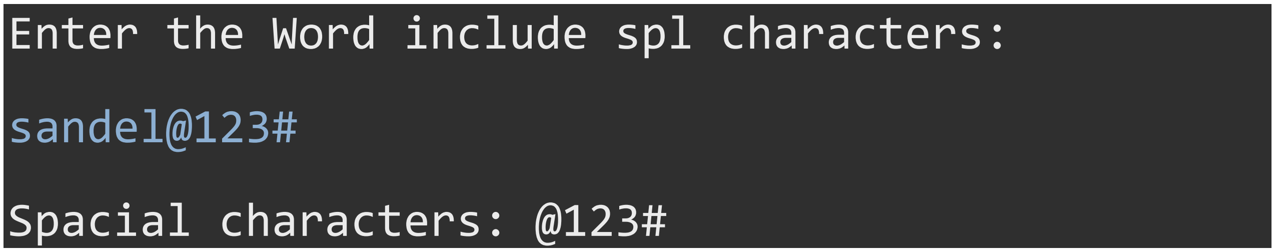


```java
package string;
import java.util.Scanner;
public class FindSplChar {
public static void main(String[] b)
{
Scanner scan=new Scanner(System.in);
System.out.println("Enter the Word include spl
characters:");
String word=scan.nextLine();
String res=Splcharfind(word);
System.out.println("Spacial characters: "
+res);
}
private static String Splcharfind(String word)
{
StringBuffer buffer=new StringBuffer();
for(int i=0;i<word.length();i++)
{
char ch=word.charAt(i);
```



```
if(!Character.isAlphabetic(ch))

{

buffer.append(ch);

}

}

return buffer.toString();

}

}
```

Output:



```
Enter the Word include spl characters:

sandel@123#

Spacial characters: @123#
```

7.Negative String

Given a string input, write a program to replace every appearance of the word "is" by "is not".

If the word "is" is immediately preceeded or followed by a letter no change should be made to the string .

Include a class UserMainCode with a static method “negativeString” that accepts a String

arguement and returns a String.

Create a class Main which would get a String as input and call the static method negativeString present in the UserMainCode.

Input and Output Format:

Input consists of a String.

Output consists of a String.

Sample Input 1: This is

just a misconception

Sample Output 1:

This is not just a misconception Sample

Input 2:

Today is misty

Sample Output 2:

Today is not misty Ans:

```
package string;

import java.util.Scanner;

import java.util.StringTokenizer;

public class NegativeSentence {

public static void main(String[] args) {

Scanner scan=new Scanner(System.in);
```

```
System.out.println("Enter the positive Sentence
include is: ");

String sent=scan.nextLine();

String res=Negsent(sent);

System.out.println(res);

}

public static String Negsent(String sent) {

StringTokenizer stoken=new
StringTokenizer(sent," ");

StringBuilder sbuilder=new StringBuilder();

while(stoken.hasMoreTokens())

{

String str=stoken.nextToken();

if(str.equals("is"))

{

sbuilder.append(str.replace("is", "is not"));

}

else

{

sbuilder.append(str);
}sbuilder.append(" ");

}

return sbuilder.toString();

}

}
```

Output:

```
Enter the positive Sentence include is:

this is a book

this is not a book
```

8. Name Shrinking

Write a program that accepts a string as input and converts the first two names into dot-separated initials and printa the output.

Input string format is 'fn mn ln'. Output string format is 'ln [mn's

1st character].[fn's 1st character]'

Include a class UserMainCode with a static method

getFormatedString which accepts a string. The return type (String)

should return the shrinked name.

Create a Class Main which would be used to accept Input String and call

the static method present in UserMainCode.

Input and Output Format:

Input consists of a string.

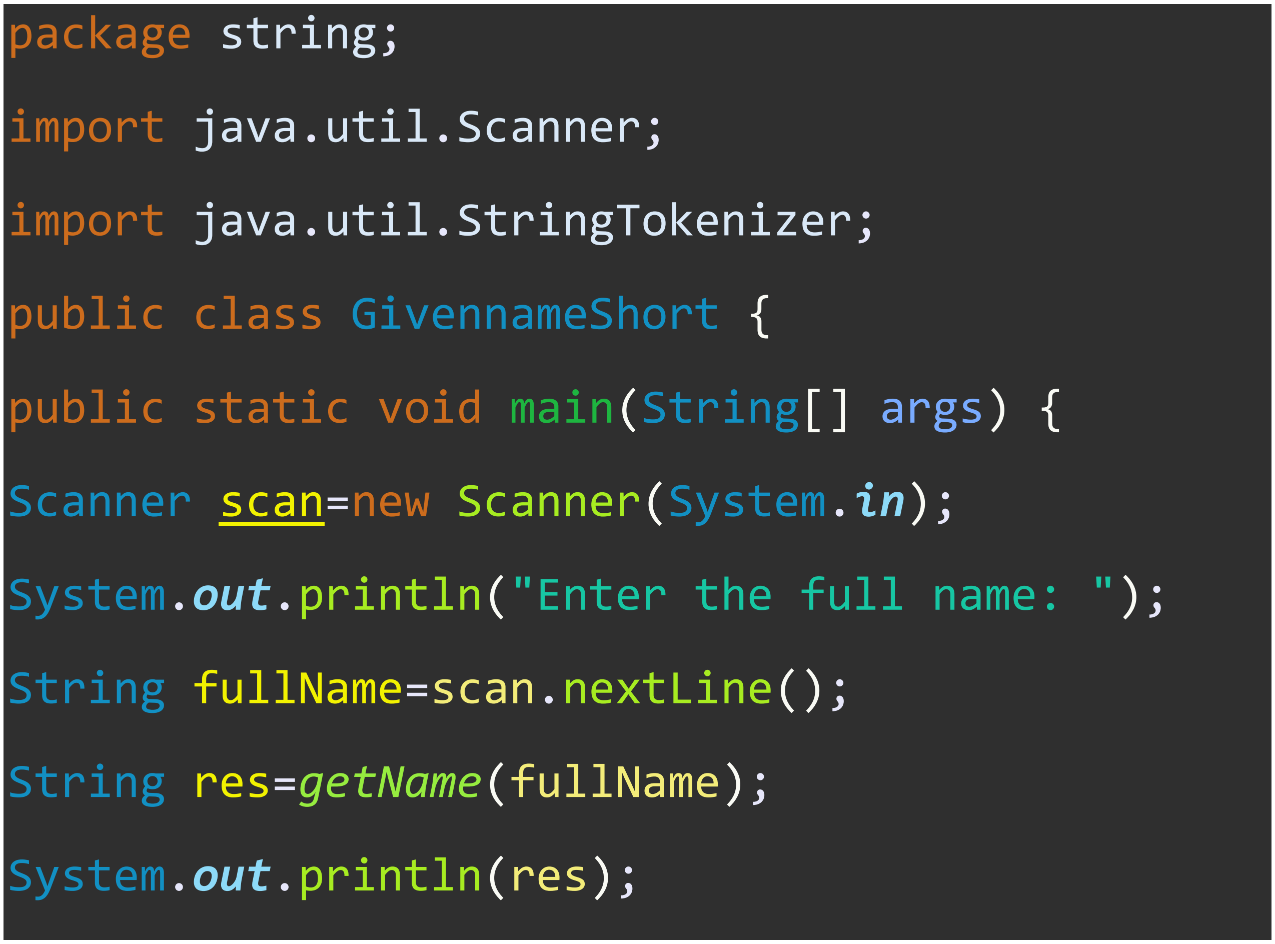
Output consists of a String.

Refer sample output for formatting specifications.

Sample Input: Sachin Ramesh Tendulkar

Sample Output:

Tendulkar R.S Ans:



```
package string;
import java.util.Scanner;
import java.util.StringTokenizer;
public class GivennameShort {
public static void main(String[] args) {
Scanner scan=new Scanner(System.in);
System.out.println("Enter the full name: ");
String fullName=scan.nextLine();
String res=getName(fullName);
System.out.println(res);
}
private static String getName(String fullName) {
StringTokenizer st=new StringTokenizer(fullName," ");
StringBuilder sb=new StringBuilder();
String s1=st.nextToken();
String s2=st.nextToken();
String s3=st.nextToken();
sb.append(s3).append(" ");
sb.append(s1.substring(0, 1)).append(".");
sb.append(s2.substring(0, 1));
return sb.toString();
}
}
```

Output:

```
Enter the full name:
Loganathan Indhirani Santhanam
Santhanam L.I
```

9.Start Case

Write a program to read a sentence in string variable and convert the

first letter of each word to capital case. Print the final string. Note:

- Only the first letter in each word should be in capital case in final

string.

Include a class UserMainCode with a static method printCapitalized which accepts a string. The return type (String) should return the capitalized string.

Create a Class Main which would be used to accept a string and call the static method present in UserMainCode.

Input and Output Format:

Input consists of a strings.

Output consists of a String (capitalized string).

Refer sample output for formatting specifications.

Sample Input:

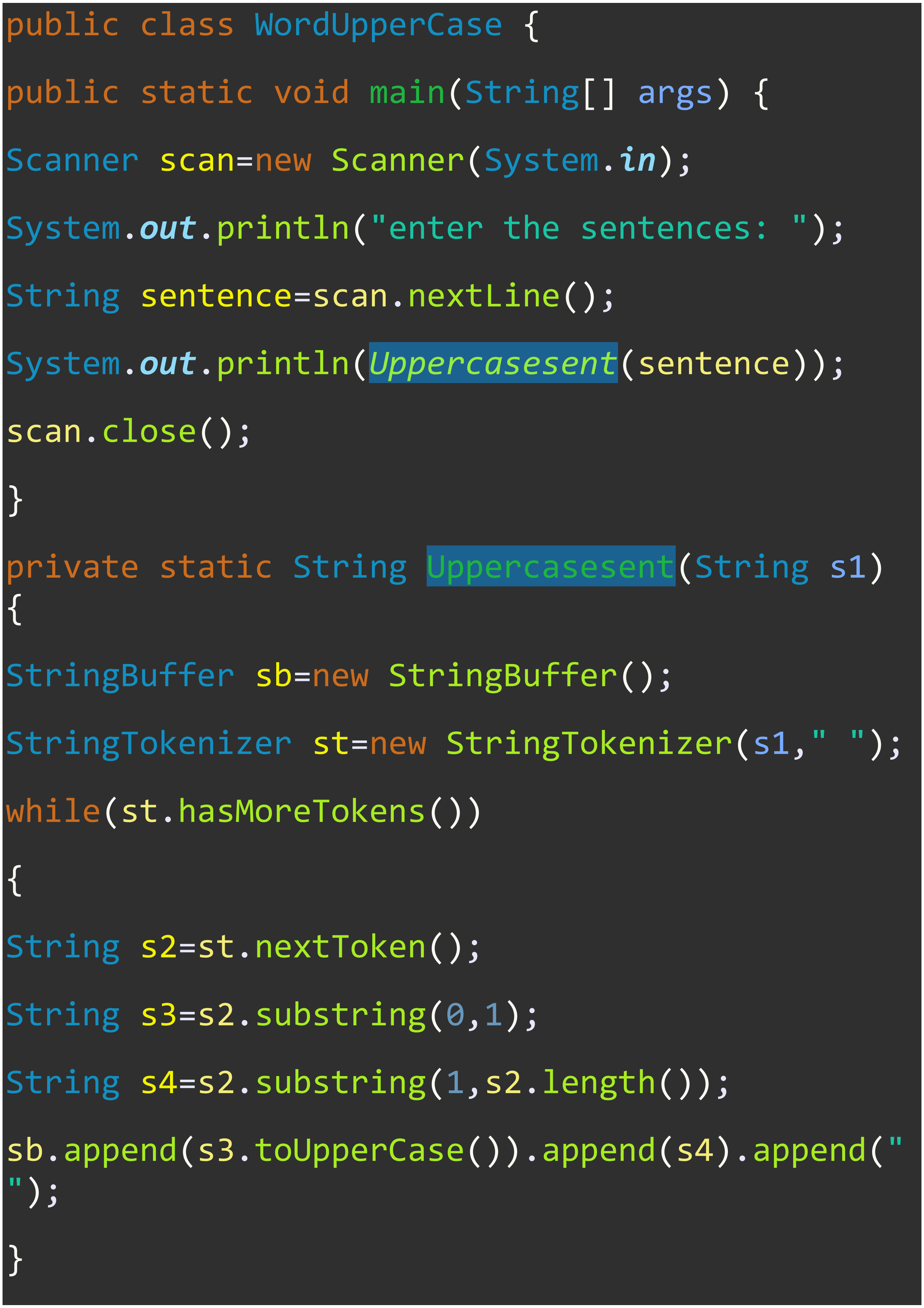
Now is the time to act!

Sample Output:

Now Is The Time To Act!

Ans:

```
package string;

import java.util.Scanner;

import java.util.StringTokenizer;
```



```
public class WordUpperCase {
public static void main(String[] args) {
Scanner scan=new Scanner(System.in);
System.out.println("enter the sentences: ");
String sentence=scan.nextLine();
System.out.println(Uppercasesent(sentence));
scan.close();
}
private static String Uppercasesent(String s1) {
StringBuffer sb=new StringBuffer();
StringTokenizer st=new StringTokenizer(s1," ");
while(st.hasMoreTokens())
{
String s2=st.nextToken();
String s3=s2.substring(0,1);
String s4=s2.substring(1,s2.length());
sb.append(s3.toUpperCase()).append(s4).append(" ");
}
return sb.toString();

}

}
```

Output:

```
enter the sentences:

do not waste your time

Do Not Waste Your Time
```

10.Occurance Count

Write a program to read a string that contains a sentence and read a word. Check the number of occurances of that word in the sentence.

Include a class UserMainCode with a static method countWords which accepts the two strings. The return type is the integer giving the count.

Note: The check is case-sensitive.

Create a Class Main which would be used to accept the two strings and call the static method present in UserMainCode.

Input and Output Format:

Input consists of two strings.

Output consists of count indicating the number of occurances.

Refer sample output for formatting specifications.

Sample Input 1:

Hello world Java is best programming language in the world world

Sample Output 1:

2

Sample Input 2:

hello world

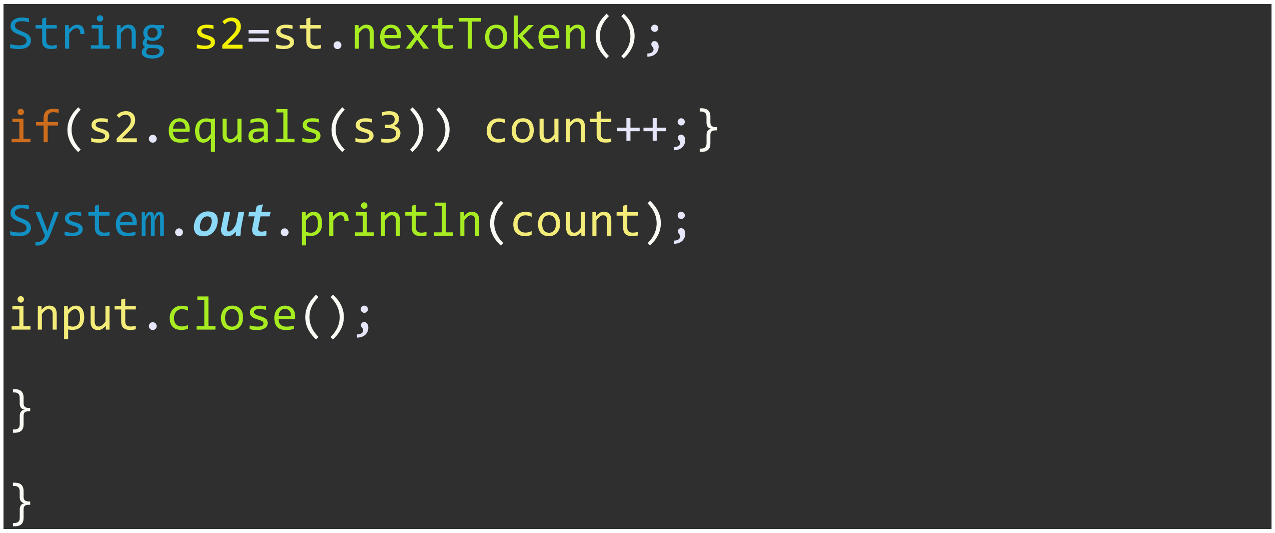
World

Sample Output 2:

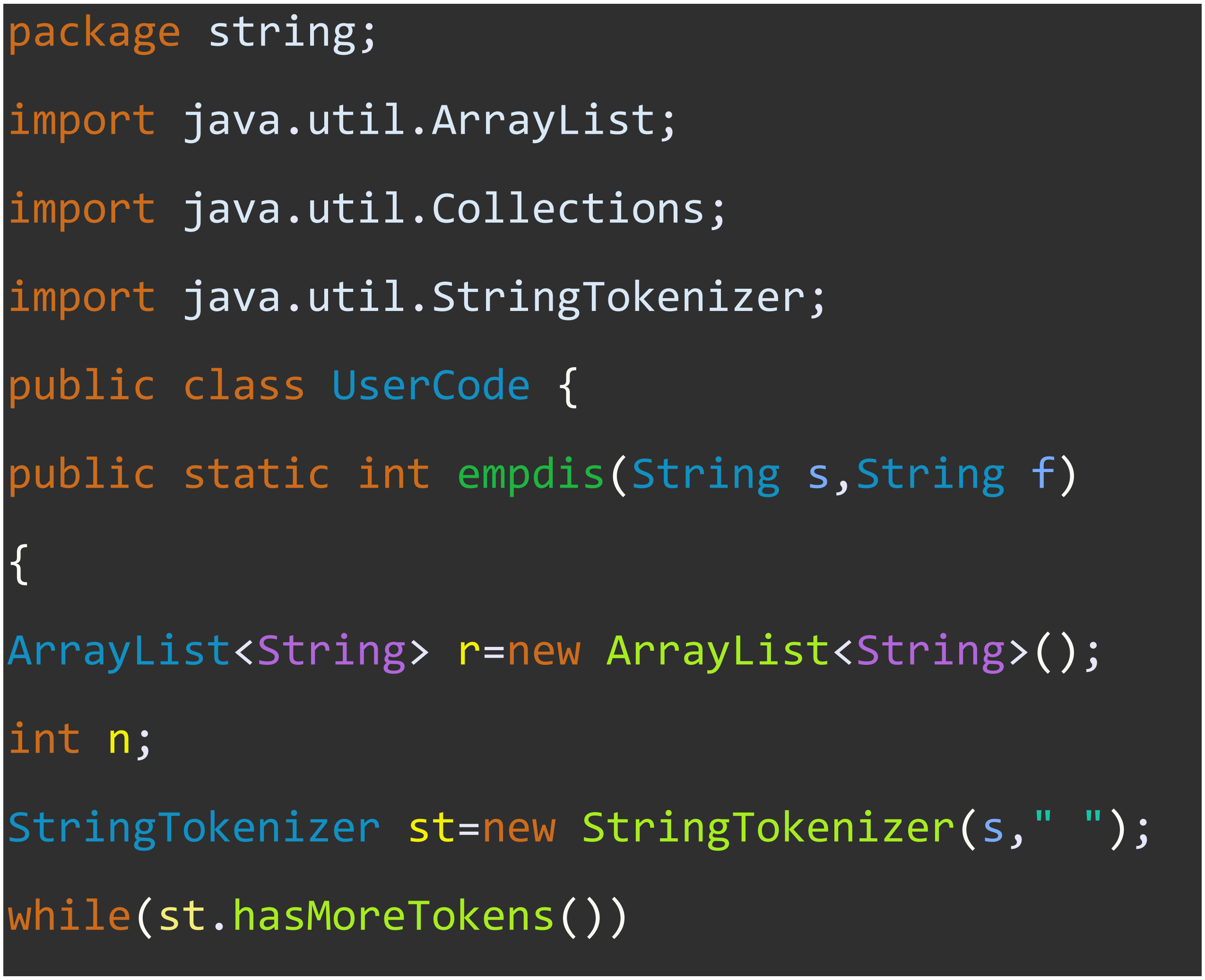
0

Ans:

```
package string;

import java.util.Scanner;

import java.util.StringTokenizer;

public class Occurrence {

public static void main(String[] args) {

Scanner input=new Scanner(System.in);

String s1=input.nextLine();

String s3=input.next(); int count=0;

StringTokenizer st=new StringTokenizer(s1," ");

while(st.hasMoreElements())

{
```



```
String s2=st.nextToken();
if(s2.equals(s3)) count++;}
System.out.println(count);
input.close();
}
}
```



```
package string;
import java.util.ArrayList;
import java.util.Collections;
import java.util.StringTokenizer;
public class UserCode {
public static int empdis(String s,String f)
{
ArrayList<String> r=new ArrayList<String>();
int n;
StringTokenizer st=new StringTokenizer(s," ");
while(st.hasMoreTokens())
{
r.add(st.nextToken());
}
n=Collections.frequency(r, f);
return n;
}
}
```

```
package string;
import java.util.Scanner;
public class UserSubMain {
public static void main(String[] args)
{
Scanner input=new Scanner(System.in);
System.out.println("enter the string: ");
String s=input.nextLine();
System.out.println("enter the word you want:
");
String f=input.next();
System.out.println(UserCode.empdis(s,f));

input.close();

}

}
```

Output:

```
enter the string:

Hello world Java is best programming language
in the World

enter the word you want:

world

1
```

11.String Processing - III

Write a program to read a string where all the lowercase 'x' chars have been moved to the end of the string.

Include a class UserMainCode with a static method moveX which accepts the string. The return type is the modified string.

Create a Class Main which would be used to accept the string and call the static method present in UserMainCode.

Input and Output Format:

Input consists of a string.

Output consists of a string.

Refer sample output for formatting specifications.

Sample Input 1:

xxhixx

Sample Output 1:

hixxxx

Sample Input 2:

XXxxtest

Sample Output 2:

XXtestxx

Ans:

```
package string;
```

```java
import java.util.Scanner;

public class Stringfilter {

public static void main(String[] args) {

Scanner input=new Scanner(System.in);

System.out.println("enter the word: ");

String word=input.nextLine();

String last=word.replaceAll("[x]", "");

String last1=word.replaceAll("[^x]", "");//x x x x x x

System.out.println(last+last1);

input.close();

}

}
```

Output:

```
enter the word:

xxxhixxhelloxx

hihelloxxxxxxx
```