

Assignment-8

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\$

```
package Stringpack;
import java.util.Scanner;
public class Main1 {
public static void main(String[] args) {
// TODO Auto-generated method stub
UserMainCode1 umc=new UserMainCode1();
String out=UserMainCode1.formString();
System.out.println(out);
}
}
class UserMainCode1{
public static String formString() {
Scanner scanner=new Scanner(System.in);
int len=scanner.nextInt();
String[] array=new String[len];
for(int j=0;j<len;j++) {
array[j]=scanner.next();
}
int num=scanner.nextInt()-1;
String ans="";
for(int i=0;i<len;i++) {
if (array[i].length()>num) {
ans=ans+array[i].charAt(num);
}
else {
ans=ans+"$";
}
}
```

```

}
return ans;
}}

```

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

```

package Stringpack;
import java.util.Scanner;

class UserMainCode3{
    public static String reverseSubstring(String str,int num1,int num2
)
    {

        StringBuffer sb = new StringBuffer(str);
        sb.reverse();
        String s = sb.substring(num1, num1 + num2);
        return s;
    }
}

public class Main2{

    public static void main(String[] args) {
        // TODO Auto-generated method stub
        Scanner sc=new Scanner(System.in);
        System.out.println("enter the values");
        String str=sc.nextLine();
        int num1=sc.nextInt();
        int num2=sc.nextInt();
        String str1=UserMainCode3.reverseSubstring(str, num1, num2);
        System.out.println(str1);

    }

}

```

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

```
package Stringpack;
import java.util.Scanner;

class UserMainCode2{
    public static String getMiddleChars(String str)
    {
        StringBuffer sb=new StringBuffer();
        if(str.length()%2==0)
        {
            System.out.println(str.substring((str.length()/2)-
1,(str.length()/2)+1));
        }
        return sb.toString();
    }
}

public class Main3 {

    public static void main(String[] args) {
        // TODO Auto-generated method stub
        Scanner sc=new Scanner(System.in);
        System.out.println("enter the even string");
        String str=sc.nextLine();

        String str1=UserMainCode2.getMiddleChars(str);
        System.out.println(str1);

    }

}
```

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

```
package Stringpack;
```

```
import java.util.Scanner;
```

```
class UserMainCode4{
    public static String getCombo(String Long , String Short)
    {
        StringBuffer s5=new StringBuffer();
        int q=Long.length();
        int w=Short.length();
        if(q>w){
            s5.append(Long).append(Short).append(Long);
        }
        else
        {
            s5.append(Short).append(Long).append(Short);
        }
        return s5.toString();
    }
}

public class Main4 {

    public static void main(String[] args) {
        // TODO Auto-generated method stub
        Scanner sc=new Scanner(System.in);
        System.out.println("enter the String");
        String s1=sc.nextLine();
        String s2=sc.nextLine();
        String s5=UserMainCode4.getCombo(s1 , s2);
        System.out.println(s5);

    }

}
```

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

```
package Stringpack;
```

```
import java.util.Scanner;
```

```
class UserMainCode6{
    public static String repeatString(String N)
    {
        Scanner sc=new Scanner(System.in);
        StringBuffer sb=new StringBuffer();
        String s=sc.next();
        int n=sc.nextInt();
        for(int i=0;i<n;i++)
            sb.append(s);
        return sb.toString();
    }
}

public class main5 {

    public static void main(String[] args) {
        // TODO Auto-generated method stub
        Scanner sc=new Scanner(System.in);
        String s=sc.next();
        String str=UserMainCode6.repeatString(s);
        System.out.println(str);
    }

}
```

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

```

package Stringpack;

import java.util.Scanner;

    public class Main6
    {

        public static void main(String[] args) {
            Scanner in=new Scanner(System.in);
            String s1=in.nextLine();
            System.out.println(UserMainCode.getSpecialChar(s1));
            in.close();
        }
        class UserMainCode{
            public static String getSpecialChar(String s1){
                int x=s1.length();
                StringBuffer sb=new StringBuffer();
                for(int i=0;i<x;i++){
                    char c=s1.charAt(i);
                    if(!Character.isAlphabetic(c))
                        sb.append(c);
                }
                return sb.toString();
            }
        }
    }

```

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

```

package Stringpack;

import java.util.Scanner;
import java.util.StringTokenizer;

class UserMainCodes{

```

```

        public static String negativeString(String s) {
            String newstring = "";
            int l = s.length();
            for(int i = 0; i < l; i++)
            {
                if(i-1 >= 0 && Character.isLetter(s.charAt(i-1)) ||
i+2 < l && Character.isLetter(s.charAt(i+2)))
                {
                    newstring += s.charAt(i);
                    continue;
                }
                else if(i+1 < l && s.substring(i,
i+2).equals("is"))
                {
                    newstring += "is not";
                    i++;
                } else
                    newstring += s.charAt(i);
            }
            return newstring;
        }
    }

    public class Main7 {

        public static void main(String[] args) {

            Scanner scanner=new Scanner(System.in);
            System.out.println("Enter the String:");
            String s=scanner.nextLine();
            String ans=UserMainCodes.negativeString(s);
            System.out.println(ans);

        }
    }

```

8. Name Shrinking

Write a program that accepts a string as input and converts the first two names into dot-separated initials and prints 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 getFormattedString which accepts a string. The return type (String) should return the shrunked 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

```

package Stringpack;
import java.util.Scanner;
import java.util.StringTokenizer;

class UserMain{

public static String getFrmatedString(String s1) {
StringBuffer sb = new StringBuffer();
StringTokenizer st = new StringTokenizer(s1, " ");
String s2 = st.nextToken();
String s3 = st.nextToken();
String s4 = st.nextToken();
sb.append(s4).append(" ");
sb.append(s3.substring(0, 1));
sb.append(".");
sb.append(s2.substring(0, 1));
System.out.println(sb);
return s1.toString();
}
}

public class Main8 {

    public static void main(String[] args) {
        // TODO Auto-generated method stub
        Scanner sc = new Scanner(System.in);
        String s1 = sc.nextLine();

        System.out.println(UserMain.getFrmatedString(s1));
    }

}

```

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!


```

package Stringpack;

import java.util.Scanner;
import java.util.StringTokenizer;

class UserMains{
public static String printCapitalized(String s1){
    StringBuffer sb=new StringBuffer();
    StringTokenizer t=new StringTokenizer(s1," ");
    while(t.hasMoreTokens())
    {
        String s2=t.nextToken();
        String s3=s2.substring(0,1);
        String s4=s2.substring(1, s2.length());
        sb.append(s3.toUpperCase()).append(s4).append(" ");
    }
    return sb.toString();
}
}

public class Main9 {

    public static void main(String[] args) {
        // TODO Auto-generated method stub

        Scanner sc = new Scanner(System.in);
        String s1= sc.nextLine();
        System.out.println(UserMains.printCapitalized(s1));

    }

}

```

10.Occurance Count

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

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

```

package Stringpack;

import java.util.Scanner;

class User{

    public static int wordCount() {
        Scanner sc = new Scanner(System.in);
        int count = 0;
        System.out.println("Enter a sentence");
        String s = sc.nextLine();
        System.out.println("please type the word for which you
want to perform wordCount");
        String s1 = sc.nextLine();
        String s3 = s.toLowerCase();
        String[] words = s.split(" ");
        System.out.println("count of word you entered is:");
        for(int i=0;i<words.length;i++) {
            if(words[i].equals(s1)) {
                count++;
            }
        }
        return count;
    }
}

public class Main10 {

    public static void main(String[] args) {
        User sim = new User();
        int y = sim.wordCount();
        System.out.println(y);
    }
}

```

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

```

package Stringpack;

```

```
import java.util.Scanner;

class flush {
    public static String moveX(String s) {
        String str=new String();
        String s1 = s.replaceAll("[x]", "");
        String s2 = s.replaceAll("[^x]", "");
        System.out.println(s1 + s2);
        return s1;
    }
}

public class Main11 {

    public static void main(String[] args) {
        // TODO Auto-generated method stub
        Scanner sc = new Scanner(System.in);
        String s = sc.next();
        System.out.println(flush.moveX(s));
    }

}
```