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 Assignment8;

import java.util.Scanner;

public class Ques1 {

 static String fromString() {
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter the size of the array: ");
        int size=sc.nextInt();
        String str[]=new String[size];
        System.out.println("Enter the strings: ");
        for(int i=0;i<size;i++)
        {
           str[i]=sc.next();
        }
      System.out.println("Enter the non negative integer: ");
      int n=sc.nextInt()-1;
      String result="";
      for(int j=0;j<str.length;j++) {
       if(str[j].length()>n)
        result=result+str[j].charAt(n);
       else
        result=result+"$";
      }
      return result;
 }

}

package Assignment8;

public class UserMainCode {

 public static void main(String[] args) {
  // TODO Auto-generated method stub
      Ques1 ques1=new Ques1();
      System.out.println(Ques1.fromString());
 }

}
```

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 Assignment8.ques2;

public class ReverseSubstring {
    static String reverseSubstring(String str,int startindex,int endindex) {
     StringBuffer sb=new StringBuffer(str);
     sb.reverse();
     String str1=sb.substring(startindex,endindex);
     return str1;
    }
}

package Assignment8.ques2;

import java.util.Scanner;

public class MainReverseSubstring {
   public static void main(String[] args)
   {
 System.out.println("Enter the string, start index and end index: ");
 Scanner sc=new Scanner(System.in);
 String str=sc.next();
 int startindex=sc.nextInt();
 int endindex=sc.nextInt();
 ReverseSubstring reversersubstring=new ReverseSubstring();
 System.out.println(ReverseSubstring.reverseSubstring(str, startindex, endindex));
   }
}
```

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 Assignment8.ques2;

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

package Assignment8.ques2;

import java.util.Scanner;

public class MainMiddleChars {

 public static void main(String[] args) {
  // TODO Auto-generated method stub
     Scanner sc=new Scanner(System.in);
     System.out.println("Enter the string: ");
     String str=sc.nextLine();
     String str1=MiddleCharacter.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 Assignment8.ques2;

public class StringCombo {
static String getCombo(String str1, String str2) {
 StringBuffer sb=new StringBuffer();
 int len1=str1.length();
 int len2=str2.length();
 if(len1>len2)
  System.out.print(str1+str2+str1);
 else if(len2>len1)
  System.out.println(str2+str1+str2);
 else
  System.out.println(str1+""+str2);
 return sb.toString();
}
}

package Assignment8.ques2;

import java.util.Scanner;

public class MainStringCombo {

 public static void main(String[] args) {
  // TODO Auto-generated method stub
     Scanner sc=new Scanner(System.in);
     System.out.println("Enter two strings");
     String s=sc.next();
     String s1=sc.next();
     StringCombo stringCombo=new StringCombo();
     System.out.println(StringCombo.getCombo(s, s1));
 }
}
```

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 Assignment8.ques2;

public class StringReplication {
 static String repeatString(String str, int n) {
   StringBuffer sb=new StringBuffer();
   for(int i=1;i<=n;i++) {
    sb.append(str);

   }
   return sb.toString();
  }
}

package Assignment8.ques2;

import java.util.Scanner;

public class MainStringReplication {

 public static void main(String[] args) {
  // TODO Auto-generated method stub
     Scanner sc=new Scanner(System.in);
     System.out.println("Enter a string and the number of times to repeat: ");
     String s=sc.next();
     int n=sc.nextInt();
     StringReplication stringReplication=new StringReplication();
     StringReplication.repeatString(s, n);
     System.out.println(StringReplication.repeatString(s, n));
 }

}
```

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 Assignment8.ques2;

public class FlushChar {
static String getSpecialChars(String s) {
 return s.replaceAll("[A-Za-z]", "");
}
}

package Assignment8.ques2;

import java.util.Scanner;

public class MainFlushChar {

 public static void main(String[] args) {
  // TODO Auto-generated method stub
       Scanner sc=new Scanner(System.in);
       System.out.println("Enter the mixture of alpha, integers and special chars");
       String str=sc.next();
       FlushChar flushChar=new FlushChar();
       System.out.println(FlushChar.getSpecialChars(str));
 }

}
```

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 Assignment8.ques2;

public class ReplaceIs {
  public static String negativeString(String s1)
  {
   String str=s1.replace(" is "," is not ");
  return str;
  }
}

package Assignment8.ques2;

import java.util.Scanner;

public class MainReplaceIs {

 public static void main(String[] args) {
  // TODO Auto-generated method stub
  Scanner s=new Scanner(System.in);
  System.out.println("Enter a Sentence");
  String s1=s.nextLine();
  System.out.println(ReplaceIs.negativeString(s1));
  s.close();

}
}
```

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
---->

```
package Assignment8.ques2;
```

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

public class FullName {
 public static void main(String[] args) {
     Scanner sc = new Scanner(System.in);
     String s1 = sc.nextLine();
     getvalues(s1);
   }
   public static void getvalues(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);

}
}
```

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 Assignment8.ques2;

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

public class StartCase {
 public static void main(String[] args) {
    // TODO Auto-generated method stub
  System.out.println("Enter sentence: ");
    Scanner sc = new Scanner(System.in);
    String s1= sc.nextLine();
```

```
        System.out.println(capsStart(s1));
    }
    public static String capsStart(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();
    }
}
```

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
```

---->

```
package Assignment8.ques2;

import java.util.Scanner;

public class OccuranceCount {
 static int countWords(String str, String word)

  {
     // split the string by spaces in a
     String a[] = str.split(" ");

     // search for pattern in a
     int count = 0;
     for (int i = 0; i < a.length; i++)
     {
     // if match found increase count
     if (word.equals(a[i]))
         count++;
     }

     return count;
 }


public static void main(String[] args) {
   // TODO Auto-generated method stub
 Scanner sc=new Scanner(System.in);
 System.out.println("Enter the sentence");
 String s1=sc.next();
 System.out.println("Enter the word");
 String s2=sc.nextLine();
 OccuranceCount oc=new OccuranceCount();
 System.out.println(oc.countWords(s1,s2));




}
}
```

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 Assignment8.ques2;
import java.util.Scanner;
public class MoveXtoLast {

 public static void main(String[] args) {
  // TODO Auto-generated method stub
  Scanner sc = new Scanner(System.in);
     String s = sc.next();
     String s1 = s.replaceAll("[x]", "");
     String s2 = s.replaceAll("[^x]", "");

     System.out.println(s1 + s2);
```

```
    }

  }
```