```
public class StringOps {
  ////////////////////////////////////////////////////////
  //////                                  ///////
  //////          Reminder:                ///////
  //////      allowed methods              ///////
  //////                                  ///////
  //////      1.charAt(int index)          ///////
  //////      2.length()                  ///////
  //////      3.substring(int start)       ///////
  //////      4.substring(int start,int ends)  ///////
  //////      5.indexOf(String str)        ///////
  //////                                  ///////
  //////      The rest are not allowed !   ///////
  //////      if you want to use a different   ///////
  //////      method, and you can implement    ///////
  //////      it using material from the course  ///////
  //////      you need to implement a version of  ///////
  //////      the function by yourself.        ///////
  //////                                  ///////
  //////      see example for substring        ///////
  //////      in Recitation 3 question 5       ///////
  //////                                  ///////
  ////////////////////////////////////////////////////////
  public static void main(String[] args) {

  }

  public static String capVowelsLowRest (String string) {
    String strrutern = "";
    for(int i = 0; i <= ((string.length()) - 1); i++)
    {
     if(string.charAt(i) == 'a' || string.charAt(i) == 'e' || string.charAt(i) == 'i' ||
string.charAt(i) == 'o' || string.charAt(i) == 'u')
     {
      strrutern = strrutern + ((char)(string.charAt(i) - 32));
     }
     else if(string.charAt(i) >= 'A' && string.charAt(i) <= 'Z')
     {
      if(string.charAt(i) == 'A' || string.charAt(i) == 'E' || string.charAt(i) == 'I' ||
string.charAt(i) == 'O' || string.charAt(i) == 'U')
      {
        strrutern = strrutern + (string.charAt(i));
      }
      else
      {
        strrutern = strrutern + ((char)(string.charAt(i) + 32));
      }
     }
```

```
      else
      {
       strrutern = strrutern + (string.charAt(i));
      }
    }
    return strrutern;
  }

  public static String camelCase (String string) {
   String strrutern = "";
   int k = 0;
   while((string.charAt(k)) == ' '){
   k++;
   }
   if(string.charAt(k) >= 'A' && string.charAt(k) <= 'Z'){
     strrutern = strrutern + ((char)((string.charAt(k)) + 32));
   }else{
     strrutern = strrutern + (string.charAt(k));
   }
   for( int i = k + 1; i <= (string.length() - 1); i++){
     if(((string.charAt(i))) != ' '){
       if(string.charAt(i-1) == ' '){
         if(string.charAt(i) >= 'A' && string.charAt(i) <= 'Z'){
           strrutern = strrutern + (string.charAt(i));
         }else{
           strrutern = strrutern + ((char)((string.charAt(i)) - 32));
         }
       }else{
         if(string.charAt(i) >= 'A' && string.charAt(i) <= 'Z'){
           strrutern = strrutern + ((char)((string.charAt(i)) + 32));
         }else{
           strrutern = strrutern + (string.charAt(i));
         }


       }
     }
   }

  return strrutern;

}

  public static int[] allIndexOf (String string, char chr) {
     int count = 0;
     for(int i = 0; i <= ((string.length()) - 1); i++){
       if(string.charAt(i) == chr ){
         count = count + 1;
```

```
        }
      }

      int [] array = new int[count];
      int counttwo = 0;
      for(int i = 0; i <= ((string.length()) - 1); i++){
        if(string.charAt(i) == chr ){
          array [counttwo] = i;
          counttwo++;

        }
      }

      return array;
    }

}


public class ArrayOps {
   public static void main(String[] args) {

   }

   public static int findMissingInt (int [] array) {
      int value = 0;
      for(int i = 0; i <= (array.length);){
        for(int k = 0; k <= array.length; k++){
          if(k == array.length){
            value = i;
            break;
          }else{
            if(i == (array[k])){
              i++;
            }
          }
        }
       break;
      }
      return value;
   }

   public static int secondMaxValue(int [] array) {
      int max = 0;
      int secondmax = 0;
      int count = 0;
      for(int i = 0; i <= (array.length - 1); i++){
       max = Math.max((array [i]), max);
      }
```

```
    for(int i = 0; i <= (array.length - 1); i++){
      if((array [i]) < max){
        secondmax = Math.max(array [i] , secondmax);
      }else{
       count++;
      }
    }
    if (count > 1) {
      return max;
    }else{
      return secondmax;
    }
}

public static boolean containsTheSameElements(int [] array1,int [] array2) {
  boolean thefirst = true;
  boolean thesecond = true;
  for(int i = 0; i <= ((array1.length) - 1); i++){
      for(int k = 0; k <= (array2.length ); ){
        if(k == array2.length){
          thefirst = false;
          break;
        }else{
          if((array1 [i]) == (array2 [k])){
            break;
          }else{
            k++;
          }
        }
      }

  }
  for(int i = 0; i <= ((array2.length) - 1); i++){
    for(int k = 0; k <= (array1.length ); ){
      if(k == array1.length){
        thesecond = false;
        break;
      }else{
        if((array2 [i]) == (array1 [k])){
          break;
        }else{
          k++;
        }
      }
    }

  }
  if(thesecond == true && thefirst == true ){
```

```
      return true;
     }else{
      return false;
     }

   }


   public static boolean isSorted(int [] array) {
    boolean bigtosmall = true;
    boolean smalltobig = true;
    for(int i = 0; i <= ((array.length)- 2); i++){
      if((array [i]) >= (array [i + 1] )){
      }else{
      bigtosmall = false;
      break;
      }
    }
    for(int i = 0; i <= ((array.length)- 2); i++){
      if((array[i]) <= (array [i+1] )) {
      }else{
      smalltobig = false;
      break;
      }
    }
    if (smalltobig == false && bigtosmall == false){
      return false;
    }else{
      return true;
    }
  }

}
```