Count the number of vowels in a given string

```
import java.util.Scanner;

public class Demo {
      static int countVowels(String s) {
            int count = 0;
            for (int i = 0; i < s.length(); i++) {
                  char c = s.charAt(i);
                  if (c == 'a' || c == 'e' || c == 'i' || c == 'o' || c
== 'u' || c == 'A' || c == 'E' || c == 'O' || c == 'I'
                              || c == 'U') {
                        count++;
                  }
            }
            return count;
      }

      public static void main(String[] args) {
            String s = "abcbea";
            System.out.println(countVowels(s));
      }
}
```

Character count

```java
public class Demo {
        static int characterCount(String s, String t) {
                int count = 0;
                char key = t.charAt(0);
                for (int i = 0; i < s.length(); i++) {
                        if (s.charAt(i) == key)
                                count++;
                }
                return count;
        }

        public static void main(String[] args) {
                String s = "How are you";
                String t ="o";
                System.out.println(characterCount(s,t));
        }
}
```

String Reverse

```java
import java.util.Scanner;

public class Demo2 {
	static String reverse(String s) {
		String t = "";
		for (int i = s.length() - 1; i >= 0; i--) {
			t = t + s.charAt(i);
		}
		return t;
	}

	public static void main(String[] args) {
		String s = "TAPACADEMY";
		System.out.println(reverse(s));
	}
}
```

Palindrome

```java
import java.util.Scanner;

public class Demo2 {
    static boolean isPalindrome(String s) {
        int i = 0;
        int j = s.length() - 1;
        while (i <= j) {
            if (s.charAt(i) != s.charAt(j)) {
                return false;
            }
            i++;
            j--;
        }
        return true;
    }

    public static void main(String[] args) {
        String s = "level";
        System.out.println(isPalindrome(s));
    }
}
```

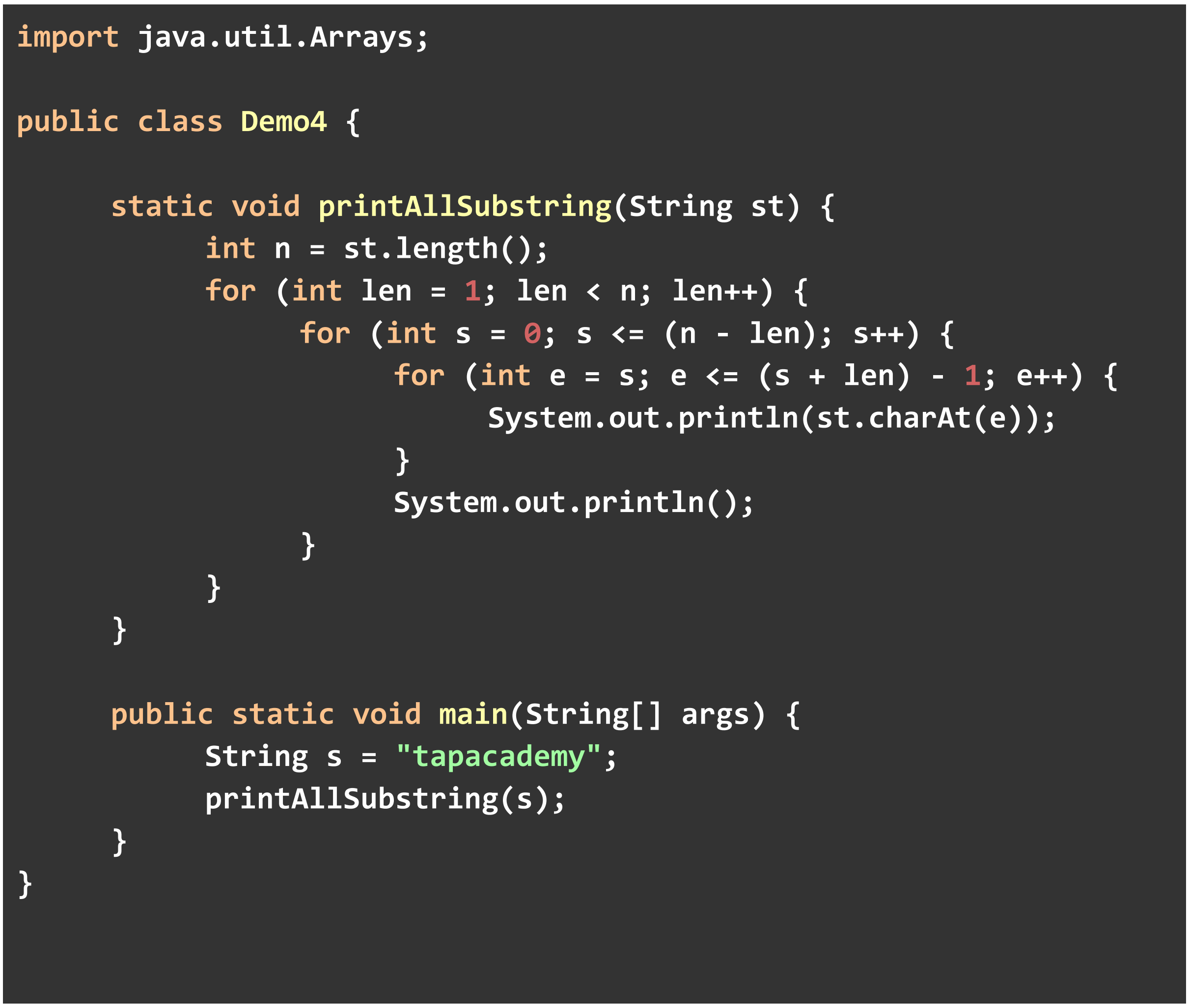
Space count

```
public class Demo4 {
      static int spaceCount(String s) {
            int count = 0;
            for (int i = 0; i < s.length(); i++) {
                  if (s.charAt(i) == ' ') {
                        count++;
                  }
            }
            return count;
      }

public static void main(String[] args) {
      String s = "How are you";
    spaceCount(s);
}
}
```

Word Count

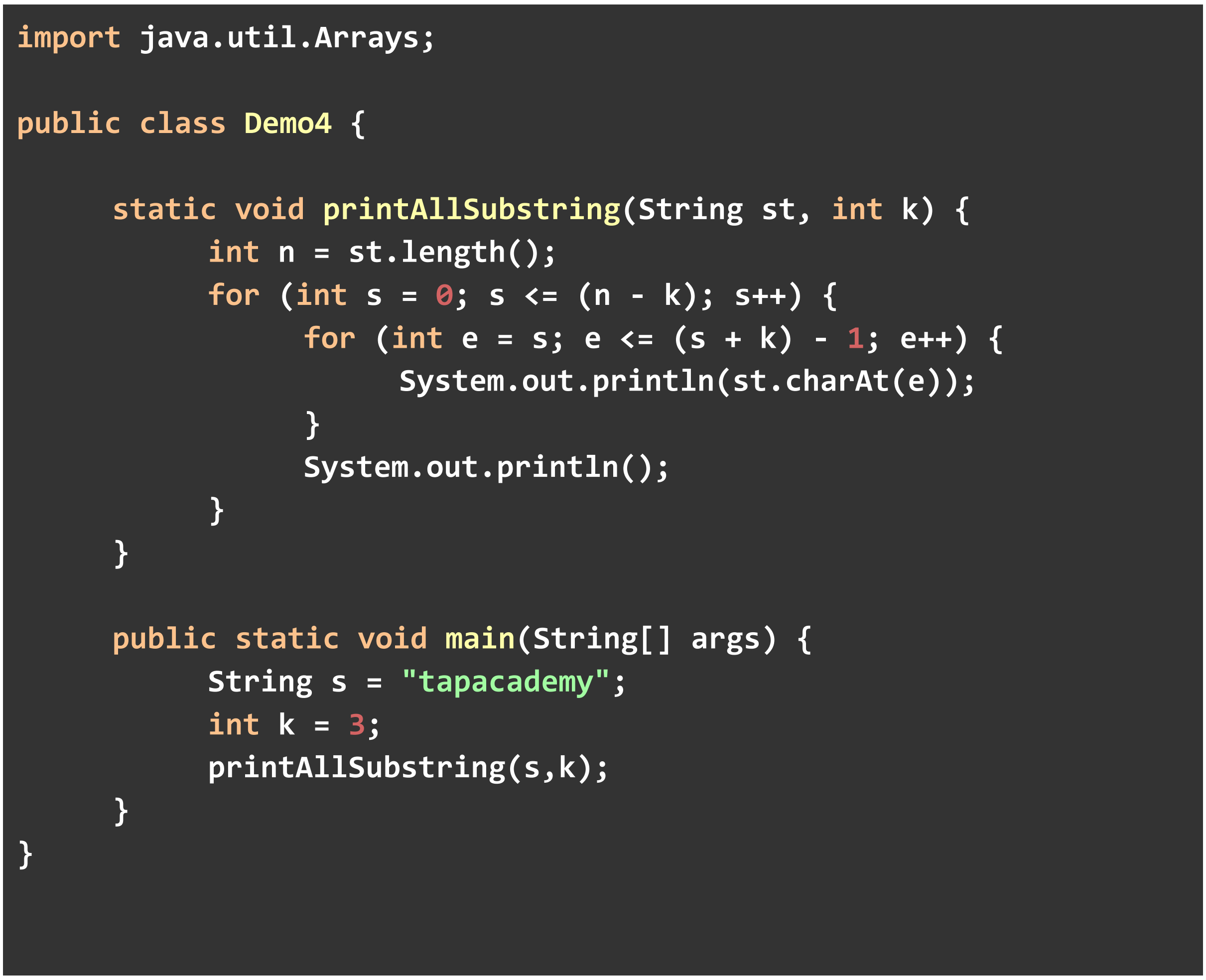
```
public class Demo4 {
      static int wordCount(String s) {
            int count = 0;
            for (int i = 0; i < s.length(); i++) {
                  if (s.charAt(i) == ' ') {
                        count++;
                  }
            }
            return count + 1;
      }

public static void main(String[] args) {
      String s = "How are you";
    wordCount(s);
}
}
```

Print All Substring



```
import java.util.Arrays;

public class Demo4 {

    static void printAllSubstring(String st) {
        int n = st.length();
        for (int len = 1; len < n; len++) {
            for (int s = 0; s <= (n - len); s++) {
                for (int e = s; e <= (s + len) - 1; e++) {
                    System.out.println(st.charAt(e));
                }
                System.out.println();
            }
        }
    }

    public static void main(String[] args) {
        String s = "tapacademy";
        printAllSubstring(s);
    }
}
```

Print substring of length 3



```java
import java.util.Arrays;

public class Demo4 {

    static void printAllSubstring(String st, int k) {
        int n = st.length();
        for (int s = 0; s <= (n - k); s++) {
            for (int e = s; e <= (s + k) - 1; e++) {
                System.out.println(st.charAt(e));
            }
            System.out.println();
        }
    }

    public static void main(String[] args) {
        String s = "tapacademy";
        int k = 3;
        printAllSubstring(s,k);
    }
}
```

Print Longest non repeating substring

```java
import java.util.Arrays;

public class Demo4 {
    static boolean isPalindrome(String s) {
        int i = 0;
        int j = s.length() - 1;
        while (i <= j) {
            if (s.charAt(i) != s.charAt(j)) {
                return false;
            }
            i++;
            j--;
        }
        return true;
    }

    static String printLongestPalinSubstring(String st) {
        int n = st.length();
        String res = "";
        for (int len = 1; len < n; len++) {
            for (int s = 0; s <= (n - len); s++) {
                String temp = "";
                for (int e = s; e <= (s + len) - 1; e++) {
                    temp = temp + st.charAt(e);
                }
                if (isPalindrome(temp)) {
                    if (temp.length() > res.length())
                        res = temp;
                }
            }
        }
        return res;
    }

    public static void main(String[] args) {
        String s = "tapacademy";
        System.out.println(printLongestPalinSubstring(s));
    }
}
```

Repeated Strings

https://www.hackerrank.com/challenges/repeated string/problem

```
import java.io.*;
import java.math.*;
import java.security.*;
import java.text.*;
import java.util.*;
import java.util.concurrent.*;
import java.util.regex.*;

class Result {
    public static long repeatedString(String s, long n) {
        int countA = 0;
        for (int i = 0; i < s.length(); i++) {
            if (s.charAt(i) == 'a') {
                countA++;
            }
        }
        long count1 = n / s.length() * countA;
        long count2 = 0;
        for (int i = 0; i < n % s.length(); i++) {
            if (s.charAt(i) == 'a') {
                count2++;
            }
        }
        return count1 + count2;
    }
}

public class Solution {
    public static void main(String[] args) throws IOException {
        BufferedReader bufferedReader = new BufferedReader(new
InputStreamReader(System.in));
        BufferedWriter bufferedWriter = new BufferedWriter(new
FileWriter(System.getenv("OUTPUT_PATH")));
        String s = bufferedReader.readLine();
        long n =
Long.parseLong(bufferedReader.readLine().trim());
           long result = Result.repeatedString(s, n);
           bufferedWriter.write(String.valueOf(result));
           bufferedWriter.newLine();
           bufferedReader.close();
           bufferedWriter.close();
      }
}
```

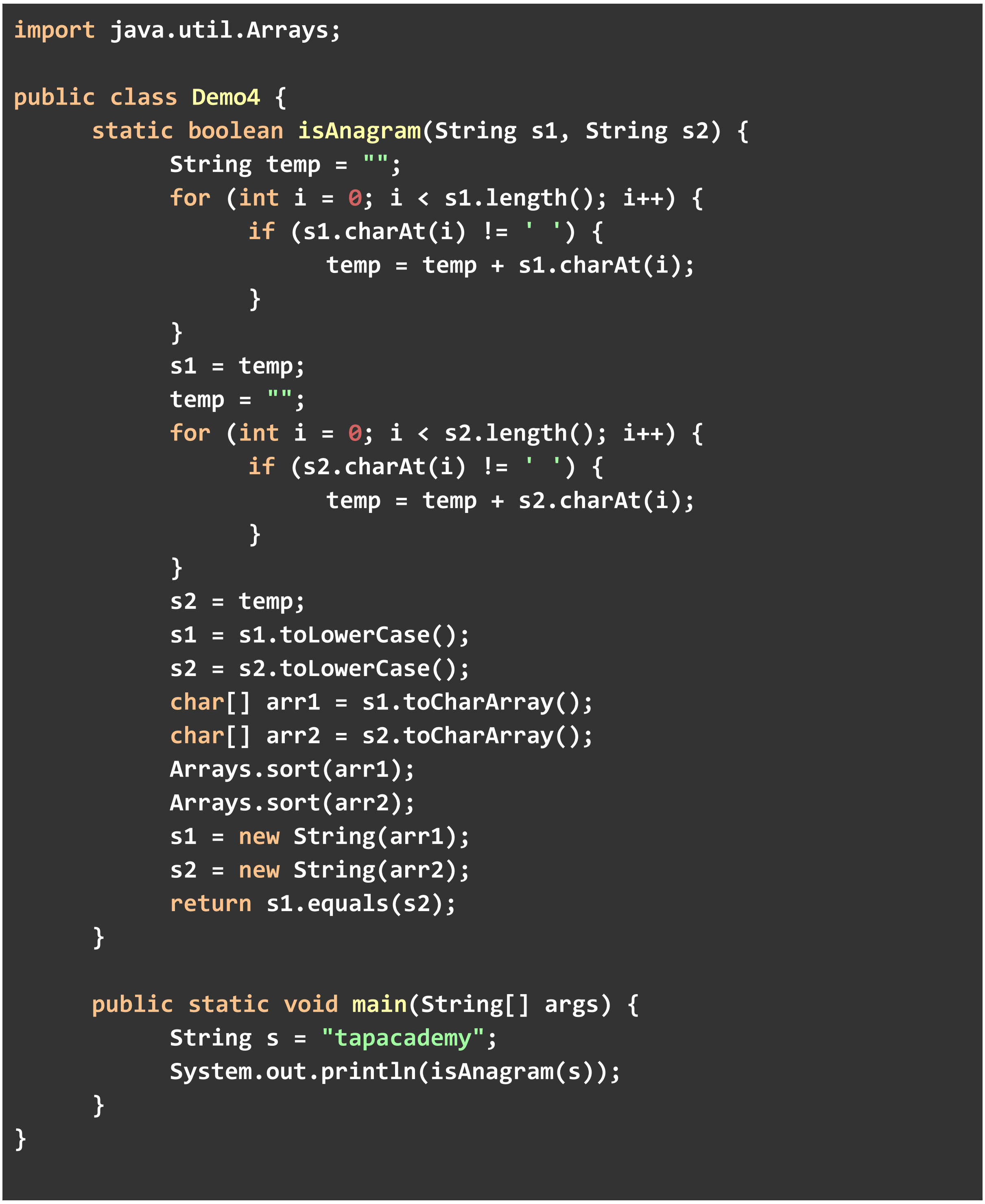
Game of thrones

https://www.hackerrank.com/challenges/game-of-thrones/problem

```
import java.io.*;
import java.math.*;
import java.security.*;
import java.text.*;
import java.util.*;
import java.util.concurrent.*;
import java.util.regex.*;

class Result {
    public static String gameOfThrones(String s) {
        char[] ar = s.toCharArray();
        Arrays.sort(ar);
        int i = 0;
        int errorCount = 0;
        while (i < ar.length) {
            if (i < ar.length - 1 && ar[i] == ar[i + 1]) {
                i = i + 2;
            } else {
                i++;
                errorCount++;
            }
        }
        if (errorCount <= 1) {
            return "YES";
        } else {
            return "NO";
        }
    }
}

public class Solution {
    public static void main(String[] args) throws IOException {
        BufferedReader bufferedReader = new BufferedReader(new
InputStreamReader(System.in));
        BufferedWriter bufferedWriter = new BufferedWriter(new
FileWriter(System.getenv("OUTPUT_PATH")));
        String s = bufferedReader.readLine();
        String result = Result.gameOfThrones(s);
        bufferedWriter.write(result);
        bufferedWriter.newLine();
```

```
            bufferedReader.close();
            bufferedWriter.close();
      }
}
```

Anagram



```java
import java.util.Arrays;

public class Demo4 {
      static boolean isAnagram(String s1, String s2) {
            String temp = "";
            for (int i = 0; i < s1.length(); i++) {
                  if (s1.charAt(i) != ' ') {
                        temp = temp + s1.charAt(i);
                  }
            }
            s1 = temp;
            temp = "";
            for (int i = 0; i < s2.length(); i++) {
                  if (s2.charAt(i) != ' ') {
                        temp = temp + s2.charAt(i);
                  }
            }
            s2 = temp;
            s1 = s1.toLowerCase();
            s2 = s2.toLowerCase();
            char[] arr1 = s1.toCharArray();
            char[] arr2 = s2.toCharArray();
            Arrays.sort(arr1);
            Arrays.sort(arr2);
            s1 = new String(arr1);
            s2 = new String(arr2);
            return s1.equals(s2);
      }

      public static void main(String[] args) {
            String s = "tapacademy";
            System.out.println(isAnagram(s));
      }
}
```

Pangram

```java
import java.util.Arrays;

public class Demo4 {
    static boolean isPangram(String s) {
        String t = "abcdefghijklmnopqrstuvwxyz";
        s = s.toLowerCase();
        int count = 0;
        for (int i = 0; i < t.length(); i++) {
            if (s.indexOf(t.charAt(i)) >= 0)
                count++;
            else
                break;
        }
        if (count == 26)
            return true;
        else
            return false;
    }

    public static void main(String[] args) {
        String s = "tapacademy";
        System.out.println(isPangram(s));
    }
}
```

Invalid Bracket

```
import java.util.Scanner;

public class Demo4 {

      public static void main(String[] args) {
            Scanner scan = new Scanner(System.in);
            String s = scan.nextLine();

            int bracket=0, count=0;

            for(int i=0; i<s.length(); i++) {
                  char c = s.charAt(i);
                  if(c == '(') {
                        bracket++;
                  }
                  else {
                        if(bracket <= 0) {
                              count++;
                        }else {
                              bracket--;
                        }
                  }
            }

            System.out.println(count + bracket);
      }
}
```

Mars Exploration:

```
import java.util.Scanner;

public class Demo4 {

    static int marsExploration(String s) {
        int count = 0;
        for(int i=0; i<s.length(); i=i+3) {
            if(s.charAt(i) != 'S') {
                count++;
            }
            if(s.charAt(i) != 'O') {
                count++;
            }
            if(s.charAt(i) != 'S') {
                count++;
            }
        }
        return count;
    }

    public static void main(String[] args) {
        Scanner scan = new Scanner(System.in);
        String s = scan.next();

        System.out.println(marsExploration(s));
    }
}
```

Camel Case

```java
import java.util.Scanner;

public class Demo4 {

        static int camelCase(String s) {
                int count = 1;
                for(int i=0; i<s.length(); i=i+3) {
                        if(s.charAt(i) >= 'A'
                                && s.charAt(i) <= 'Z') {
                                count++;
                        }
                }
                return count;
        }

        public static void main(String[] args) {
                Scanner scan = new Scanner(System.in);
                String s = scan.next();

                System.out.println(camelCase(s));
        }
}
```

Strong Password

```java
import java.util.Scanner;

public class Demo4 {

    static int minimumNumber(int n, String s) {
        int digit=1, lc=1, uc=1,sc=1;

        for(int i=0; i<s.length(); i++) {
            char c = s.charAt(i);
            if(c>='0' && c<='9') {
                digit = 0;
            }
            else if(c>='a' && c<='z') {
                lc = 0;
            }
            else if(c>='A' && c<='Z') {
                uc = 0;
            }else {
                sc=0;
            }
        }

        if((digit+lc+uc+sc) > 6-n) {
            return (digit+lc+uc+sc);
        }else {
            return 6-n;
        }
    }

    public static void main(String[] args) {
        Scanner scan = new Scanner(System.in);
        int n = scan.nextInt();
        String s = scan.next();

        System.out.println(minimumNumber(n, s));
    }
}
```