**1.**    **Check Sum of Odd Digits**

Write a program to read a number , calculate the sum of odd digits (values) present in the given number.  
  
Include a class **UserMainCode** with a static method **checkSum** which accepts a positive integer . The return type should be 1 if the sum is odd . In case the sum is even return -1 as output.  
  
Create a class **Main** which would get the input as a positive integer and call the static method **checkSum** present in the UserMainCode.  
  
**Input and Output Format:**  
Input consists of a positive integer n.  
Refer sample output for formatting specifications.

**Sample Input 1:**

56895

**Sample Output 1:**

Sum of odd digits is odd.

**Sample Input 2:**

84228

**Sample Output 2:**

Sum of odd digits is even.

**2.**  **Sum of Squares of Even Digits**

Write a program to read a number , calculate the sum of squares of even digits (values) present in the given number.

Include a class **UserMainCode** with a static method **sumOfSquaresOfEvenDigits** which accepts a positive integer . The return type (integer) should be the sum of squares of the even digits.

Create a class **Main** which would get the input as a positive integer and call the static method sumOfSquaresOfEvenDigits present in the UserMainCode.

**Input and Output Format:**

Input consists of a positive integer n.

Output is a single integer .

Refer sample output for formatting specifications.

**Sample Input 1:**

56895

**Sample Output 1:**

100

**3**. **Reversing a Number**

Write a program to read a positive number as input and to get the reverse of the given number and return it as output.

Include a class **UserMainCode** with a static method **reverseNumber** which accepts a positive integer .

The return type is an integer value which is the reverse of the given number.

Create a **Main** class which gets the input as a integer and call the static method **reverseNumber** present in the **UserMainCode**

**Input and Output Format:**

Input consists of a positive integer.

Output is an integer .

Refer sample output for formatting specifications.

**Sample Input 1:**

543

**Sample Output 1:**

345

**Sample Input 1:**

1111

**Sample Output 1:**

1111

**4.**  **Unique Number**

Write a program that accepts an Integer as input and finds whether the number is Unique or not. Print Unique if the number is “Unique”, else print “Not Unique”.

**Note:** A Unique number is a positive integer (without leading zeros) with no duplicate digits.For example 7, 135, 214 are all unique numbers whereas 33, 3121, 300 are not.

Include a class **UserMainCode** with a static method **getUnique** which accepts an integer. The return type (Integer) should return 1 if the number is unique else return -1.

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

**Input and Output Format:**

Input consists of an integer .

Output consists of a String (“Unique” or “Not Unique”).

Refer sample output for formatting specifications.

**Sample Input 1:**

123

**Sample Output 1:**

Unique

**Sample Input 2:**

33

**Sample Output 2:**

Not Unique

**5**. **Digits**

Write a program to read a non-negative integer n, that returns the count of the occurances of 7 as digit.  
  
Include a class UserMainCode with a static method **countSeven** which accepts the integer value. The return type is integer which is the count value.  
  
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 integer.  
Output consists of integer.  
Refer sample output for formatting specifications.  
  
**Sample Input 1:**  
717  
  
**Sample Output 1:**  
2  
  
**Sample Input 2:**  
4534  
  
**Sample Output 2:**  
0