Q1. ISBN Number Program

ISBN is another special number in Java. ISBN stands for the International Standard Book Number

that is carried by almost each every book. The ISBN is a ten-digit unique number. With the help of

the ISBN, we can easily find any book. The ISBN number is a legal number when 1\*Digit1 +

2\*Digit2 + 3\*Digit3 + 4\*Digit4 + 5\*Digit5 + 6\*Digit6 + 7\*Digit7 + 8\*Digit8 + 9\*Digit9 + 10\*Digit10

is divisible by 11. The digits are taken from right to left. So, if the ten-digit number is 7426985414,

Digit1 and Digit10 will be 4 and 7, respectively.

Code:

// "static void main" must be defined in a public class.

public class Main {

public static void main(String[] args) {

System.out.println("enter a 10 digits number");

Scanner sc=new Scanner(System.in);

String isbn=sc.next();

if(isbn.length()!=10) {

System.out.println("enter a not valid isbn number");

}

int sum=0;

for(int i=0; i<10; i++) {

int digit = Character.getNumericValue(isbn.charAt(i));

sum+= digit\*(i + 1);

}

if(sum%11==0) {

System.out.println("its is isbn num");

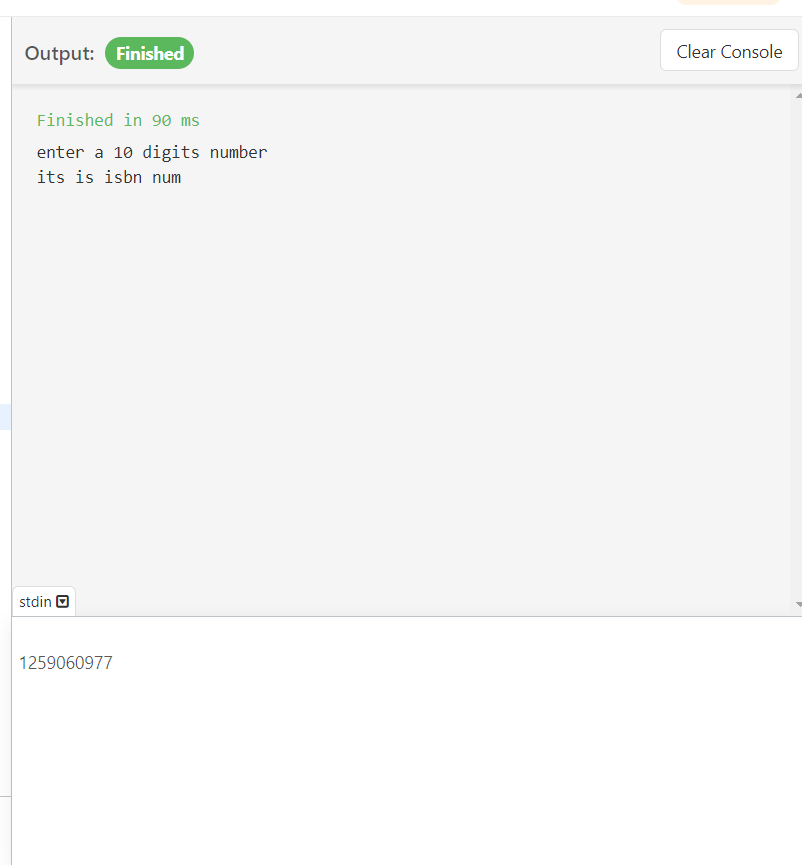
} else {

System.out.println(" its is not isbn number");

}

}

}



Q2. Prime Number: A prime number is a number p such that whenever p

divides ab, then either p divides a or p divides b. In other words, a number

that is divisible by itself only is called a prime number. For example, 2, 3, 5,

7, etc. are prime numbers.

Code:

import java.util.Scanner;

public class AlternatePrimes {

// Method to check if a number is prime

public static boolean isPrime(int num) {

if (num <= 1) return false;

if (num == 2) return true;

if (num % 2 == 0) return false;

// Check divisibility from 3 to sqrt(num)

for (int i = 3; i <= Math.sqrt(num); i += 2) {

if (num % i == 0) {

return false;

}

}

return true;

}

public static void printAlternatePrimes(int n) {

int count = 0;

System.out.println("Alternate prime numbers up to " + n + ":");

for (int i = 2; i <= n; i++) {

if (isPrime(i)) {

if (count % 2 == 0) {

System.out.print(i + " ");

}

count++;

}

}

System.out.println();

}

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

System.out.print("Enter the value of n: ");

int n = scanner.nextInt();

printAlternatePrimes(n);

}

}



Q3. The perfect square or square number is a positive integer that is square

of an integer. In other words, when we multiply two same numbers together,

then the product that we get is called the perfect square. In short, it is the

product of two positive equal integers or product of an integer with itself.

The property of the perfect square number is that it ends only with 0, 1, 4, 6,

9, and 25. The examples of perfect square are:

Code:

ublic class Main {

public static void main(String[] args) {

long x=40;

if(perfectNumber(x)) {

System.out.println("yes");

} else {

System.out.println("no");

}

}

static boolean perfectNumber(long x) {

if(x>=0) {

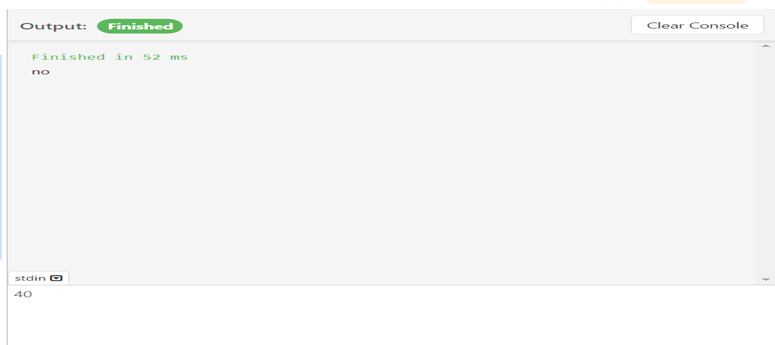
long sr=(long)Math.sqrt(x);

return(sr\*sr==x);

}

return false;

}

}

Q4. The natural numbers are the numbers that include all the positive

integers from 1 to infinity. For example, 1, 2, 3, 4, 5, ......, n. When we

add these numbers together, we get the sum of natural numbers.

Code:

public class Main {

public static void main(String[] args) {

Scanner sc=new Scanner(System.in);

int num =sc.nextInt();

int sum=0;

for(int i=0; i<=num; i++) {

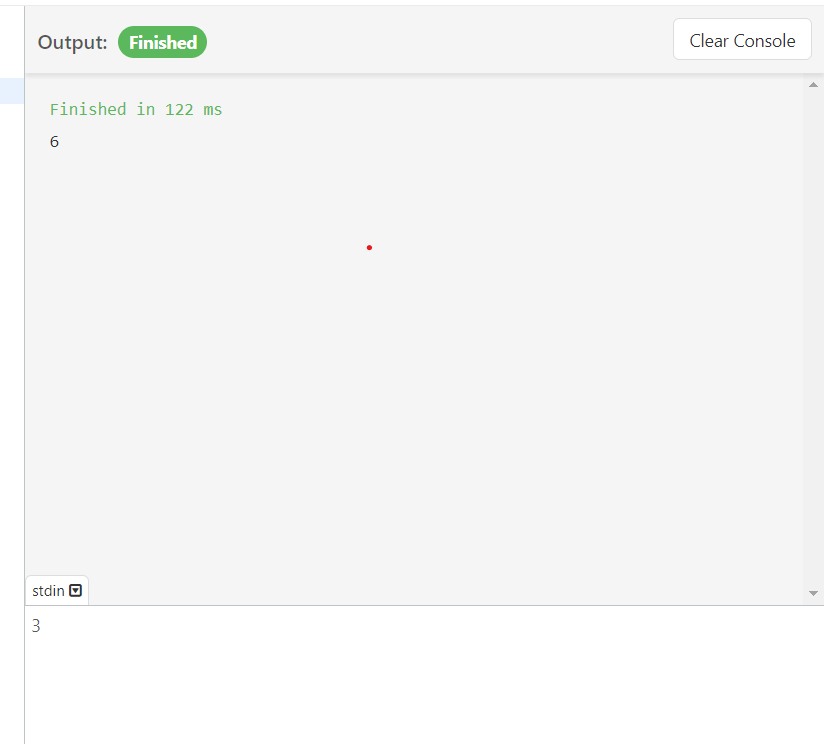
sum=sum+i;

}

System.out.println(sum);

}

}



A number is prime if it is divisible by 1 and itself. In other words, a prime

number is a natural number with exactly two distinct natural number divisors 1

and itself. For example, 2, 3, 5, 7, 11, etc. are the prime numbers. Note that 0

and 1 are not prime numbers. The number 2 is the only even prime number

because all the other even numbers are divisible by 2.

Code:

import java.util.Scanner;

public class NthPrime {

public static boolean isPrime(int num) {

if (num <= 1) return false;

if (num == 2) return true;

if (num % 2 == 0) return false;

for (int i = 3; i <= Math.sqrt(num); i += 2) {

if (num % i == 0) {

return false;

}

}

return true;

}

public static int nthPrime(int n) {

int count = 0;

int num = 1;

while (count < n) {

num++;

if (isPrime(num)) {

count++;

}

}

return num;

}

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

System.out.print("Enter the value of n to find the nth prime number: ");

int n = scanner.nextInt();

int nthPrimeNumber = nthPrime(n);

System.out.println("The " + n + "th prime number is: " + nthPrimeNumber);

}

}

