1. Odd or Even
2. Prime Number or not
3. Print Prime Numbers from a to b
4. Sum of digits of a Number
5. Palindrome Number
6. Perfect Number
7. Armstrong Number
8. Factorial of a Number
9. Factorial of a Number Using Recursion
10. Power of a Number
11. Simple Interest Calculation
12. Calculator Using Switch

**Odd or Even**

**package** test;

**public** **class** OddOrEvenDemo {

**public** **static** **void** main(String[] args) {

OddOrEvenDemo obj = **new** OddOrEvenDemo();

**int**[] arr = { 3, 12, 88, 55, 134 };

**int** n = arr.length;

obj.checkEvenOrOdd(arr, n);

}

**public** **void** checkEvenOrOdd(**int**[] arr, **int** n) {

**for** (**int** i = 0; i < n; i++) {

**boolean** result = isEven(arr[i], n);

System.***out***.println(arr[i] + " " + result);

}

}

**public** **boolean** isEven(**int** num, **int** n) {

**if** (num % 2 == 0) {

**return** **true**;

}

**return** **false**;

}

}

**Prime Number or not**

**package** test;

**public** **class** PrimeOrNotDemo {

**public** **static** **void** main(String[] args) {

PrimeOrNotDemo obj = **new** PrimeOrNotDemo();

**int**[] arr = { 3, 12, 88, 55, 134 };

**int** n = arr.length;

obj.checkPrimeOrNot(arr, n);

}

**public** **void** checkPrimeOrNot(**int**[] arr, **int** n) {

**for** (**int** i = 0; i < n; i++) {

**boolean** result = isPrime(arr[i], n);

System.***out***.println(arr[i] + " " + result);

}

}

**public** **boolean** isPrime(**int** num, **int** n) {

**for** (**int** i = 2; i < num; i++) {

**if** (num % i == 0) {

**return** **false**;

}

}

**return** **true**;

}

}

**Print Prime Numbers from a to b**

**package** test;

**public** **class** PrimeOrNotDemo {

**public** **static** **void** main(String[] args) {

PrimeOrNotDemo obj = **new** PrimeOrNotDemo();

**int**[] arr = { 3, 12, 88, 55, 134 };

**int** n = arr.length;

obj.printPrimeFromStartToEnd(7, 100);

}

**public** **void** printPrimeFromStartToEnd(**int** start, **int** end) {

**int** k = 0;

**int** n = 0;

**int**[] arr = **new** **int**[end - start];

**for** (**int** i = start; i < end; i++) {

arr[k++] = i;

n = arr.length;

}

checkPrimeOrNot(arr, n);

}

**public** **void** checkPrimeOrNot(**int**[] arr, **int** n) {

**for** (**int** i = 0; i < n; i++) {

**boolean** result = isPrime(arr[i], n);

**if**(result)

System.***out***.println(arr[i] + " " + result);

}

}

**public** **boolean** isPrime(**int** num, **int** n) {

**for** (**int** i = 2; i < num; i++) {

**if** (num % i == 0) {

**return** **false**;

}

}

**return** **true**;

}

}

**Sum of digits of a Number**

**package** test;

**public** **class** SumOfDigitsOfANumberDemo {

**public** **static** **void** main(String[] args) {

SumOfDigitsOfANumberDemo obj = **new** SumOfDigitsOfANumberDemo();

**int** n = 12345;

**int** result = obj.sumOfDigits(n);

System.***out***.println(result);

}

**public** **int** sumOfDigits(**int** n) {

**if** (n < 0)

**return** 0;

**int** r = 0;

**int** sum = 0;

**while** (n > 0) {

r = n % 10;

sum = sum + r;

n = n / 10;

}

**return** sum;

}

}

**Palindrome Number**

**package** test;

**public** **class** PalindromeNumberDemo {

**public** **static** **void** main(String[] args) {

PalindromeNumberDemo obj = **new** PalindromeNumberDemo();

String number = "123456789987654321";

**int** n = number.length();

**boolean** result = obj.isPalindrome(number, n);

System.***out***.println(result);

}

**public** **boolean** isPalindrome(String number, **int** n) {

**int**[] intArr = **new** **int**[n];

**for** (**int** i = 0; i < n; i++) {

intArr[i] = Integer.*parseInt*(Character.*toString*(number.charAt(i)));

}

**for** (**int** i = 0, j = n - 1; i < j; i++, j--) {

**if** (intArr[i] != intArr[j]) {

**return** **false**;

}

}

**return** **true**;

}

}

**Perfect Number**

**package** test;

**public** **class** PerfectNumberDemo {

**public** **static** **void** main(String[] args) {

PerfectNumberDemo obj = **new** PerfectNumberDemo();

**int**[] arr = { 1, 2, 3, 4, 5, 6, 26, 27, 28, 100 };

**for** (**int** i = 0; i < arr.length; i++) {

**boolean** result = obj.isPerfectNumber(arr[i]);

System.***out***.println(arr[i] + " " + result);

}

}

**public** **boolean** isPerfectNumber(**int** n) {

**if** (n <= 0)

**return** **false**;

**int** sumOfDivisors = 0;

**for** (**int** i = 1; i < n; i++) {

**if** (n % i == 0) {

sumOfDivisors += i;

}

}

**if** (sumOfDivisors == n)

**return** **true**;

**return** **false**;

}

}

**Armstrong Number**

**package** test;

**public** **class** ArmstrongNumberDemo {

**public** **static** **void** main(String[] args) {

ArmstrongNumberDemo obj = **new** ArmstrongNumberDemo();

**int** n = 100000;

**int**[] arr = **new** **int**[n];

**for** (**int** i = 0; i < n; i++) {

arr[i] = i;

}

**for** (**int** i = 0; i < arr.length; i++) {

**boolean** result = obj.isArmstrongNumber(arr[i]);

**if** (result)

System.***out***.println(arr[i] + " " + result);

}

}

**public** **boolean** isArmstrongNumber(**int** n) {

**if** (n <= 0)

**return** **false**;

**int** num = n;

**int** r = 0;

**int** sum = 0;

**while** (n != 0) {

r = n % 10;

sum = sum + r \* r \* r;

n = n / 10;

}

**if** (sum == num)

**return** **true**;

**return** **false**;

}

}

**Factorial of a Number**

**package** test;

**public** **class** FactorialDemo {

**public** **static** **void** main(String[] args) {

FactorialDemo obj = **new** FactorialDemo();

**int** n = 10;

**int**[] arr = **new** **int**[n];

**for** (**int** i = 0; i < n; i++) {

arr[i] = i;

}

**for** (**int** i = 0; i < arr.length; i++) {

**int** result = obj.getFactorial(arr[i]);

System.***out***.println(arr[i] + " " + result);

}

}

**public** **int** getFactorial(**int** n) {

**if** (n == 0 || n == 1)

**return** 1;

**int** fact = 1;

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

fact = fact \* i;

}

**return** fact;

}

}

**Factorial of a Number Using Recursion**

**package** test;

**public** **class** FactorialRecursionDemo {

**public** **static** **void** main(String[] args) {

FactorialRecursionDemo obj = **new** FactorialRecursionDemo();

**int** n = 10;

**int**[] arr = **new** **int**[n];

**for** (**int** i = 0; i < n; i++) {

arr[i] = i;

}

**for** (**int** i = 0; i < arr.length; i++) {

**int** result = obj.fact(arr[i]);

System.***out***.println(arr[i] + " " + result);

}

}

**public** **int** fact(**int** n) {

**if** (n == 0 || n == 1)

**return** 1;

**return** n \* fact(n - 1);

}

}

**Power of a Number**

**package** test;

**public** **class** PowerOfANumberDemo {

**public** **static** **void** main(String[] args) {

PowerOfANumberDemo obj = **new** PowerOfANumberDemo();

**int** base = 2;

**int** exponent = 5;

**int** result = obj.power(base, exponent);

System.***out***.println(base + " to the power " + exponent + " is " + result);

}

**public** **int** power(**int** base, **int** exponent) {

**if** (exponent == 0)

**return** 1;

**int** result = 1;

**for** (**int** i = 0; i < exponent; i++) {

result = result \* base;

}

**return** result;

}

}

**Calculate Simple Interest**

**package** test;

**public** **class** SimpleInterestDemo {

**public** **static** **void** main(String[] args) {

SimpleInterestDemo obj = **new** SimpleInterestDemo();

**int** p = 100;

**int** r = 5;

**int** t = 3;

**int** result = obj.calculateSimpleInterest(p, r, t);

System.***out***.println(result);

}

**public** **int** calculateSimpleInterest(**int** p, **int** r, **int** t) {

**int** interest = 0;

interest = p \* r \* t / 100;

**return** interest;

}

}

**Calculator Using Switch**