1. If we list all the natural numbers below 10 that are multiples of 3 or 5, we get 3, 5, 6 and 9. The sum of these multiples is 23.Find the sum of all the multiples of 3 or 5 below 1000.Ans: 233168

public class multiple {

public static void main(String[] args) {

int sum = 0;

for (int i = 0; i < 1000 ; i++) {

if(i%3 == 0 || i%5 ==0 ){ //check both conditions

sum += i;

}

}

System.out.println(sum);

}

}

2. Each new term in the Fibonacci sequence is generated by adding the previous two terms. By starting with 1 and 2, the first 10 terms will be: 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, ... By considering the terms in the Fibonacci sequence whose values do not exceed four million, find the sum of the even-valued terms. Ans: 4613732

public class Fibonacci {

public static void main(String[] args) {

int a = 0, b = 1, temp = 0, res = 0;

while(a <= 4000000){

temp = a;

a = b;

b += temp;

if( b%2 == 0){

res += b;

}

}

System.out.println(res);

}

}

3. The prime factors of 13195 are 5, 7, 13 and 29. What is the largest prime factor of the number 600851475143? Ans: 6857

public class LargestPrimeFactor {

public static void main(String[] args) {

long n = 600851475143L;

for (long i = 2; i < n/2 ; i++) {

while (n%i==0) {

n=n/i;

}

}

System.out.println(n);

}

}

4. A palindromic number reads the same both ways. The largest palindrome made from the product of two 2-digit numbers is 9009 = 91 × 99. Find the largest palindrome made from the product of two 3-digit numbers. Ans: 906609

public class LargestPalindromeNum {

public static boolean isPalindrome(int n){

int sum = 0, temp, rem;

temp = n;

while(n>0){

rem = n % 10;

sum = (sum\*10)+rem;

n = n/10;

}

return temp == sum;

}

public static void main(String[] args) {

int max = -1;

for(int i = 999 ; i >= 100; i--){

if(max >= i\*999){

break;

}

for(int j = 999 ; j>=i ; j--){

int prod = i\*j;

while(max < prod && isPalindrome(prod)){

max = prod;

}

}

}

System.out.println(max > -1 ? max : "No Palindrome found");

}

}

5. 2520 is the smallest number that can be divided by each of the numbers from 1 to 10 without any remainder. What is the smallest positive number that is evenly divisible by all of the numbers from 1 to 20? Ans: 232792560

public class SmallestPositivNum {

public static boolean isDivisible(int n){

for (int i = 1; i <= 20 ; i++) { //we can also start from 11

if(n%i !=0 ){

return false;

}

}

return true;

}

public static void main(String[] args) {

int num = 1;

while(!isDivisible(num)){

num+=1;

}

System.out.println(num);

}

}

6. Find the difference between the sum of the squares of the first one hundred natural numbers and the square of the sum. Ans: 25164150

public class Diff\_SumofSquares {

public static void main(String[] args) {

int n = 100;

int sumofsquares = (n\*(n+1)\*(2\*n+1))/6;

int sumofn = (n\*(n+1))/2;

sumofn = sumofn\*sumofn;

int m = Math.abs(sumofsquares-sumofn);

System.out.println(m);

}

}

7. By listing the first six prime numbers: 2, 3, 5, 7, 11, and 13, we can see that the 6th

prime is 13. What is the 10,001st prime number? Ans: 104743

public class NthPrime {

public static void main(String[] args) {

long numOfPrimes = 0;

int num = 1;

int max = 10000000;

boolean[] prime = new boolean[max];

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

if ( prime[i] == true ) continue;

numOfPrimes++;

if ( numOfPrimes == 10001 ) {

num = i;

break;

}

for ( int j = i+i; j < max; j += i )

prime[j] = true;

}

System.out.println(num);

}

}

10. The sum of the primes below 10 is 2 + 3 + 5 + 7 = 17. Find the sum of all the primes below two million. Ans: 142913828922

public class SumofPrime {

public static void main(String[] args) {

long num = 2000000;

boolean[] prime = new boolean[(int)(num+1)];

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

prime[i] = true;

}

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

if(prime[i]){

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

prime[i\*j] = false;

}

}

}

long sum = 0;

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

if(prime[i]){

sum += i;

}

}

System.out.println(sum);

}

}