1. Armstrong Number

package Day\_2;

public class p1 {

//Recursion 0 1 1 2 3 5 8 13 21

static int fib(int n ) {

if ( n <= 1 ) {

return n ;

}

return *fib* (n-1)+*fib*(n-2);

}

public static void main(String []args) {

int num = 100 ;

for(int i=1 ; i <= num ; i++ ) {

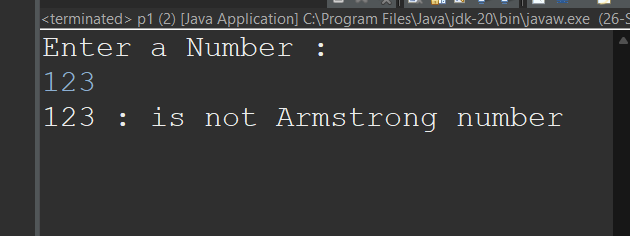
System.***out***.println(*fib*(i)+" ");

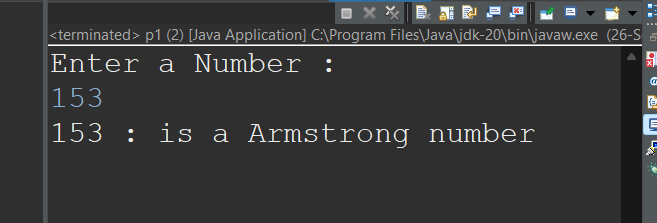
}

}

}

OUTPUT :





Explanation :

The program checks if input number is equal to the sum of the cubes of its digits.It extracts each digit using modulus and division operations, cubes it, and adds the result to a cumulative sum. Finally, it compares this sum with the original number to determine if it’s an Armstrong number.

1. Prime Number

package Assignment\_1;

import java.util.Scanner;

public class p2 {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.***in***);

System.***out***.print("Enter a number: ");

int n = scanner.nextInt();

boolean isPrime = *ckprime*(n);

System.***out***.println("Output: " + isPrime);

scanner.close();

}

public static boolean ckprime(int num) {

if (num <= 1) {

return false;

}

else if (num == 2) {

return true;

}

else if (num % 2 == 0) {

return false;

}

for (int i = 3; i \* i <= num; i += 2) {

if (num % i == 0) {

return false;

}

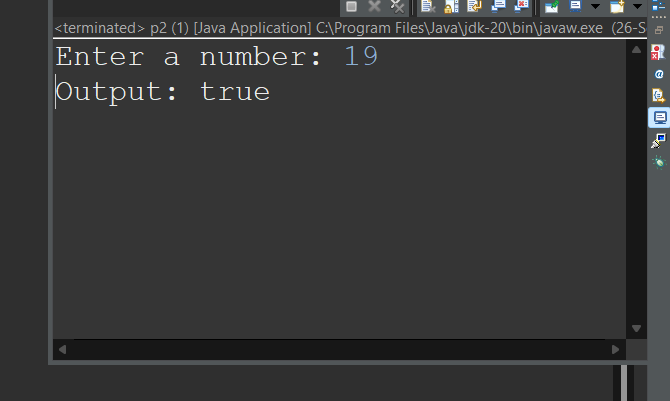
}

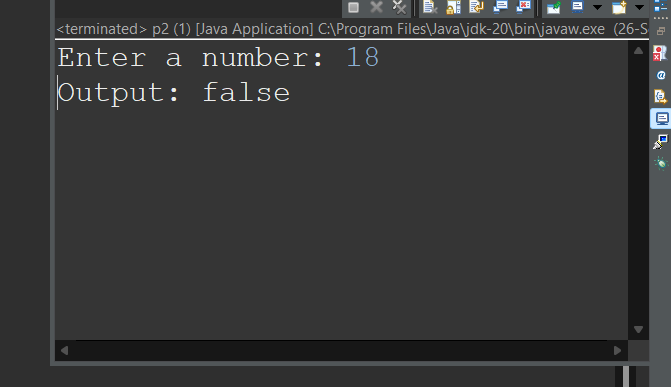
return true;

}

}

OUTPUT :





Explanation :

The program checks whether the input number is prime by testing divisibility from 2 to the square root of the number.

It returns false if the number is divisible by any number in this range, otherwise, it returns true, indicating the number is prime.

1. Factorial Number

package Assignment\_1;

import java.util.Scanner;

public class p3 {

public static void main(String args[]) {

Scanner sc = new Scanner(System.***in***);

System.***out***.println("Enter any Number : ");

int num = sc.nextInt() ;

int n = num ;

int sum = 1 ;

int i = 1 ;// for counting the factorial

while(i<= num ) {

sum = sum \*i;

i++;

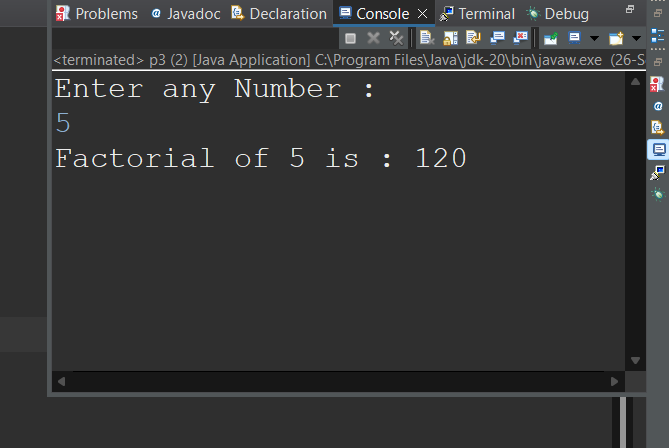
}

sc.close() ;

System.***out***.println("Factorial of "+ n+" is : "+sum);

}

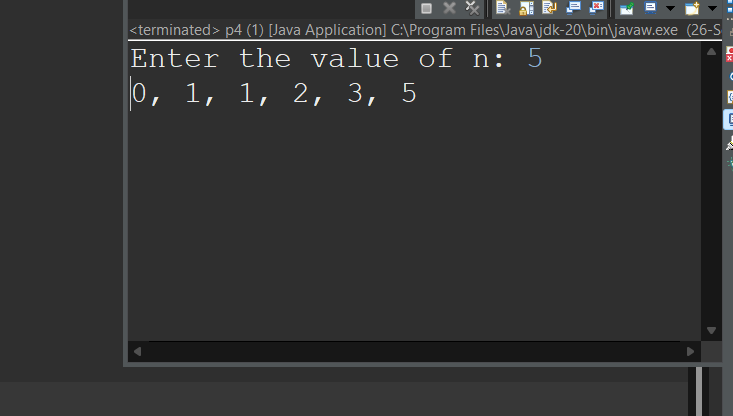
}

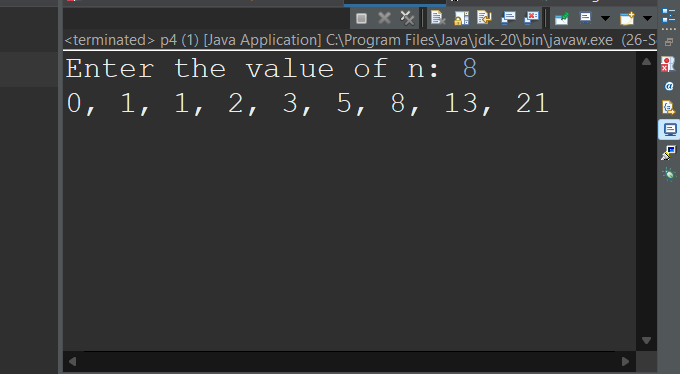


The program calculates the factorial of an input number by multiplying all integers from 1 to the number.

It uses a loop to compute the product iteratively and prints the result.

1. Fibonacci Series
2. package Assignment\_1;
3. import java.util.Scanner;
4. public class p4 {
5. public static void main(String[] args) {
6. Scanner scanner = new Scanner(System.***in***);
7. System.***out***.print("Enter the value of n: ");
8. int n = scanner.nextInt();
10. *printFibonacci*(n);
11. }
12. public static void printFibonacci(int n) {
13. int a = 0, b = 1;
14. if (n > 0) System.***out***.print(a);
15. for (int i = 0; i < n; i++) {
16. System.***out***.print(", " + b);
17. int next = a + b;
18. a = b;
19. b = next;
20. }
22. System.***out***.println();
23. }
24. }
25. OUTPUT :





Explanation :

The program prints the Fibonacci sequence up to the nth term, where each term is the sum of the previous two.

It uses two variables to store the previous terms, updating them iteratively to generate the sequence.

5. . Find GCD

package Assignment\_1;

import java.util.Scanner;

public class p5{

public static int fgcd(int a5 , int b5) {

if( b5==0) {

return a5 ;

}

return *fgcd*(b5 ,a5 % b5);

}

public static void main(String a[]) {

Scanner sc = new Scanner(System.***in***);

System.***out***.println("Enter First Number : ");

int num = sc.nextInt() ;

System.***out***.println("Enter Second Number : ");

int num2 = sc.nextInt() ;

int gcd = *fgcd*(num,num2);

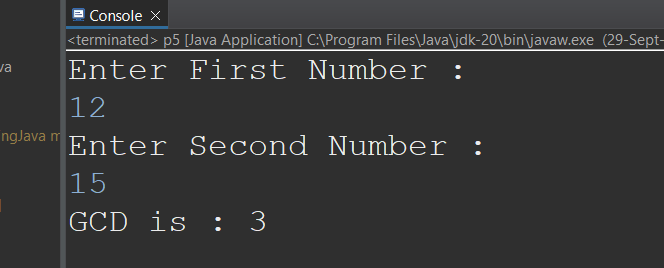
System.***out***.println("GCD is : "+gcd);

sc.close();

}

}

OUTPUT :



The program calculates the greatest common divisor (GCD) of two numbers

It recursively finds the GCD by replacing the larger number with its remainder when divided by the smaller number until the remainder is zero.

6. Find Square Root

package Assignment\_1;

import java.util.Scanner ;

//package Assignment\_1;

import java.util.Scanner;

public class p6 {

public static void main(String[] a) {

Scanner sc = new Scanner(System.***in***);

System.***out***.println("Enter a number to find its square root: ");

int x = sc.nextInt();

int result = *integerSqrt*(x);

System.***out***.println("The integer square root of " + x + " is : " + result);

sc.close();

}

public static int integerSqrt(int x) {

if (x < 0) return -1;

int i;

for (i = 0; i \* i <= x; i++) {

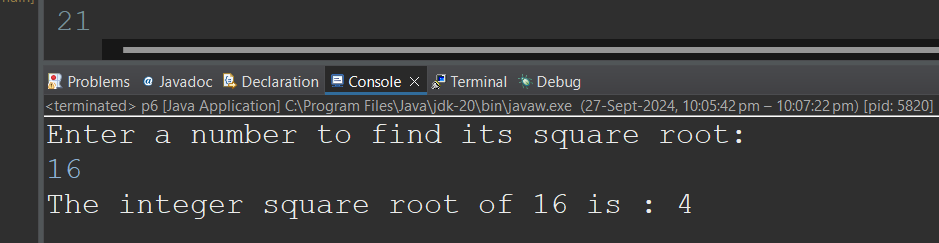
}

return i - 1;

}

}

OUTPUT



Explanation :

The program computes the integer square root of a given number by incrementing a value until its square exceeds the input number.

The largest value whose square is less than or equal to the input is returned as the integer square root.

7. Find Repeated Characters in a String

package Assignment\_1;

import java.util.Scanner;

public class p7 {

public static char findFirstNonRepeatedCharacter(String input) {

int[] charCount = new int[256];

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

charCount[input.charAt(i)]++;

}

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

if (charCount[input.charAt(i)] == 1) {

return input.charAt(i);

}

}

return 0;

}

public static void main(String[] args) {

Scanner scanner = new Scanner(System.***in***);

System.***out***.print("Enter a string: ");

String input = scanner.nextLine();

char result = *findFirstNonRepeatedCharacter*(input);

if (result != 0) {

System.***out***.println("First non-repeated character: " + result);

} else {

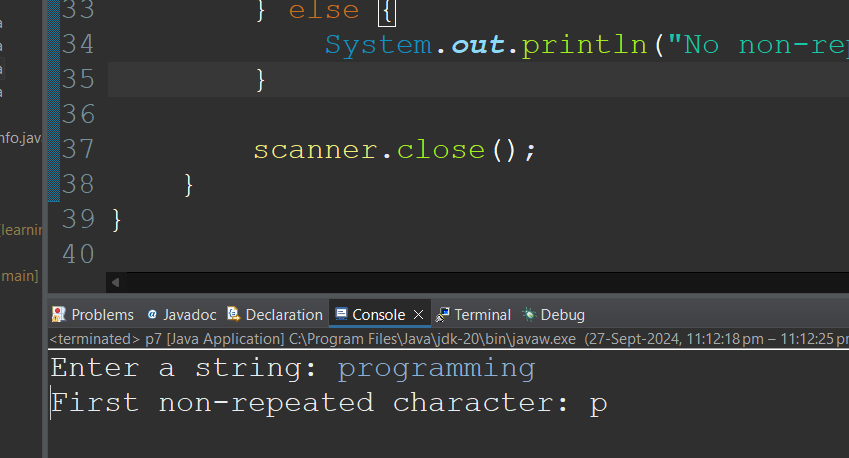
System.***out***.println("No non-repeated character found.");

}

scanner.close();

}

}



Explanation :

The program reads a string input from the user and counts the frequency of each character using an array. It then checks the array to find the first character that appears only once in the string. If such a character is found, it prints it; otherwise, it informs the user that no non-repeated character was found.

8. First Non-Repeated Character

package Assignment\_1;

import java.util.Scanner;

public class p8 {

public static char findFirstNonRepeatedCharacter(String input) {

int[] charCount = new int[256];

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

charCount[input.charAt(i)]++;

}

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

if (charCount[input.charAt(i)] == 1) {

return input.charAt(i);

}

}

return 0;

}

public static void main(String[] args) {

Scanner scanner = new Scanner(System.***in***);

System.***out***.print("Enter a string: ");

String input = scanner.nextLine();

char result = *findFirstNonRepeatedCharacter*(input);

if (result != 0) {

System.***out***.println("First non-repeated character: " + result);

} else {

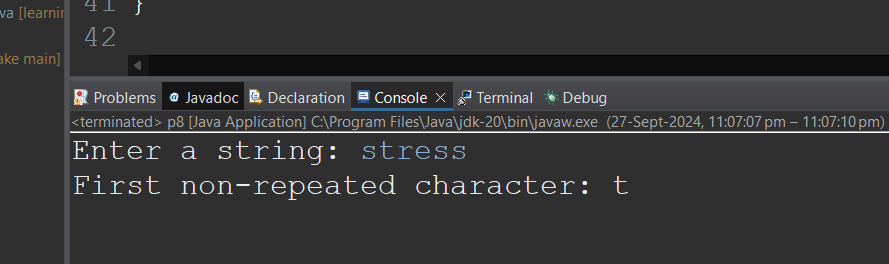
System.***out***.println("No non-repeated character found.");

}

scanner.close();

}

}



Explanation :

The program takes a string input and counts the frequency of each character using an array. It then iterates through the string again to find the first character that appears only once by checking the frequency array. If found, it returns and prints that character, otherwise it prints that no non-repeated character was found.

9. Palindrome

import java.util.Scanner;

public class PalindromeNumber {

    public static void main (String a[]){

        Scanner scanner = new Scanner (System.in);

        System.out.println("Enter a number");

        int number = scanner.nextInt();

        if(isPalindrome(number)){

            System.out.println(number+" It is Palindrome.");

        }else{

            System.out.println(number+" It is not a Palindrome.");

        }

        scanner.close();

    }

    public static boolean isPalindrome(int number){

        int reversed = 0 , original = number;

        while (number!=0);{

            reversed = reversed \* 10  + number % 10 ;

            number /= 10;

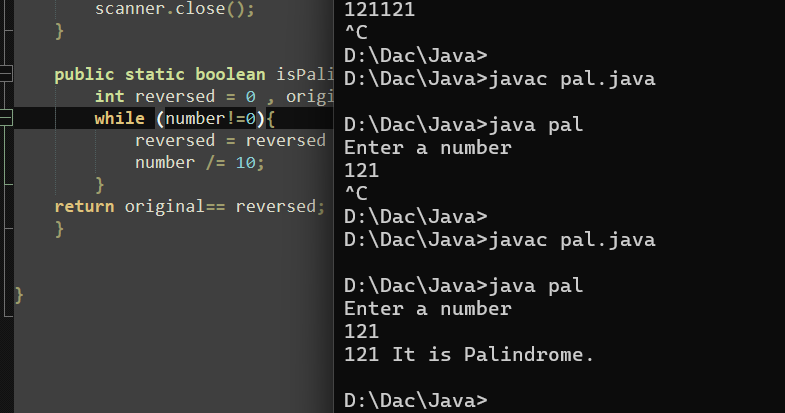
        }

    return original== reversed;

    }

}

OUTPUT :



Explanation :

The program checks if a number is a palindrome by reversing its digits and comparing the reversed number with the original.

If both are equal, the number is a palindrome; otherwise, it is not.

10. Leap year

package Assignment\_1;

import java.util.Scanner;

public class p10{

public static void main(String[] args) {

Scanner sc = new Scanner(System.***in***);

System.***out***.println("Enter a year: ");

int year = sc.nextInt();

//boolean isLeap = false;

boolean isLeap = (year % 4 == 0 && year % 100 != 0) || (year % 400 == 0);

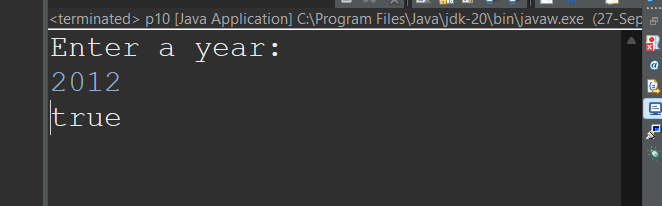
System.***out***.println( isLeap);

sc.close();

}

}

Output :



Explanation :

The program checks if the input year is a leap year by applying leap year rules: divisible by 4 but not 100, unless divisible by 400.

It reads a year from the user and uses a boolean expression to determine and print if the year is a leap year.