**Programming Practices**

**Program 1:WAP to find factorial of a number.**

Code:

import java.util.Scanner;

public class Prog1 {

public static void factorial(int number) {

long fact = 1;

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

fact = fact \* i;

}

System.out.println("Factorial of " + number + " is:" + fact);

}

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

System.out.print("Enter the number:");

int number = sc.nextInt();

factorial(number);

}

}

**Prog.2 WAP to print Fibonacci Series.**

Code:

import java.util.Scanner;

public class Prog2 {

public static void fibonacci(int number) {

int temp1 = 0;

int temp2 = 1;

int temp3;

System.out.print("Fibonacci Series : ");

System.out.print(temp1 + " " + temp2);

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

temp3 = temp1 + temp2;

System.out.print(" " + temp3);

temp1 = temp2;

temp2 = temp3;

}

}

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

System.out.print("Enter the number: ");

int number = sc.nextInt();

fibonacci(number);

}

}

**Prog.3 Greatest Factorial as the factor of a number.**

Code:

import java.util.Scanner;

public class Prog3 {

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

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

long number = sc.nextLong();

long fact = 1;

int ans = 0;

for (int i = 1; i <= 30; i++) {

fact = fact \* i;

if (number % fact == 0) {

ans = i;

}

}

System.out.println("Greatest factiorial as factor in number is: " + ans);

}

}

**Prog.4 WAP to check for a palindrome**.

**Code:**

import java.util.Scanner;

public class Prog4 {

public static void isPalindrome(long number) {

long temp, rev, rem;

temp = number;

rev = 0;

while (temp > 0) {

rem = temp % 10;

rev = (rev \* 10) + rem;

temp = temp / 10;

}

if (rev == number) {

System.out.println("Number is Palindrome.");

} else {

System.out.println("Not a Palindrome.");

}

}

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

System.out.print("Enter a Number:");

long number = sc.nextLong();

isPalindrome(number);

}

}

**Prog.5 WAP a program to reverse a string.**

Code:

import java.util.Scanner;

public class Prog5 {

public static void reverse(String str) {

String revstr = "";

char ch;

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

ch = str.charAt(i);

revstr = ch + revstr;

}

System.out.println("Reversed String :" + revstr);

}

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

System.out.print("Enter a String:");

String str = sc.nextLine();

reverse(str);

}

}

**Prog.6 Check weather the number is prime or not.**

Code:

import java.util.Scanner;

public class Prog6 {

public static void isPrime(int number) {

int count = 0;

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

if (number % i == 0) {

count = count + 1;

}

}

if (count == 0) {

System.out.println("Number is prime.");

} else {

System.out.println("Number is not prime.");

}

}

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

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

int number = sc.nextInt();

isPrime(number);

}

}

**Prog.7 Perfect Square in Range.**

Code:

import java.util.Scanner;

public class Prog7 {

public static void perfectSquare(int number) {

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

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

if (i % j == 0 && i == j \* j) {

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

}

}

}

}

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

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

int number = sc.nextInt();

perfectSquare(number);

}

}

**Prog.8 Largest number in the unsorted Array.**

Code:

import java.util.Scanner;

public class Prog8 {

static int LargestNumber(int[] array) {

if (array.length == 0) {

System.out.println("Array is empty.");

return 0;

}

int largest = array[0];

for (int i = 1; i < array.length; i++) {

if (array[i] > largest) {

largest = array[i];

}

}

return largest;

}

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

System.out.print("Enter the size of the array: ");

int size = sc.nextInt();

int[] numbers = new int[size];

System.out.println("Enter the elements of the array:");

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

numbers[i] = sc.nextInt();

int largest = LargestNumber(numbers);

System.out.println("The largest number in the array is: " + largest);

}

}

}

**Prog.9 Decimal number to Binary number.**

Code:

import java.util.Scanner;

public class Prog9 {

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

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

int number = sc.nextInt();

String binaryRepresentation = decimalToBinary(number);

System.out.println("Binary representation: " + binaryRepresentation);

}

static String decimalToBinary(int number) {

if (number == 0) {

return "0";

}

StringBuilder binary = new StringBuilder();

while (number > 0) {

int remainder = number % 2;

binary.insert(0, remainder);

number /= 2;

}

return binary.toString();

}

}

**Prog.10 Decimal number to hexadecimal number**

Code:

import java.util.Scanner;

public class Prog10 {

static String decimalToHexadecimal(int number) {

if (number == 0) {

return "0";

}

StringBuilder hexadecimal = new StringBuilder();

char[] hexChars = "0123456789ABCDEF".toCharArray();

while (number > 0) {

int remainder = number % 16;

hexadecimal.insert(0, hexChars[remainder]);

number /= 16;

}

return hexadecimal.toString();

}

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

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

int number = sc.nextInt();

String hexadecimal = decimalToHexadecimal(number);

System.out.println("Hexadecimal representation: " + hexadecimal);

}

}

**Prog.11 Program to check for a happy number.**

Code:

import java.util.Scanner;

public class Prog11 {

public static void isHappyNumber(int number) {

int temp = number;

int sum = 0;

int rem = 0;

while (temp > 1) {

while (temp > 0) {

rem = temp % 10;

sum = sum + (rem \* rem);

temp = temp / 10;

}

temp = sum;

}

if (temp == 1) {

System.out.println("Is a happy number.");

} else {

System.out.println("Is not a happy number.");

}

}

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

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

int number = sc.nextInt();

isHappyNumber(number);

}

}

**Prog.12 Program to check for a Armstrong number**

Code:

import java.util.Scanner;

public class Prog12 {

public static void isArmstrong(int number) {

int temp = number;

int sum = 0;

int rem = 0;

while (temp != 0) {

rem = temp % 10;

sum = sum + (rem \* rem \* rem);

temp = temp / 10;

}

System.out.println(sum);

if (number == sum) {

System.out.println("Is Armstrong Number.");

} else {

System.out.println("Not a Armstrong.");

}

}

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

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

int number = sc.nextInt();

isArmstrong(number);

}

}

**Prog.13 WAP to find a closest number to given number which is divisible by another given number.**

Code:

import java.util.Scanner;

public class Prog13 {

public static void closest(int number, int factor) {

if (number % factor != 0) {

for (int i = number; i > 1; i--) {

if (i % factor == 0) {

System.out.println(i + " is closest multiple of " + factor + " to " + number);

break;

}

}

}

}

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

System.out.print("Enter the number: ");

int number = sc.nextInt();

System.out.print("Enter the factor: ");

int factor = sc.nextInt();

closest(number, factor);

}

}

**Prog 14. WAP to compute nCr.**

Code:

import java.util.Scanner;

public class Prog14 {

public static long factorial(int number) {

long fact = 1;

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

fact = fact \* i;

}

return fact;

}

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

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

int n = sc.nextInt();

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

int r = sc.nextInt();

long ncr;

ncr = factorial(n) / (factorial(r) \* factorial(n - r));

System.out.print("nCr : ");

System.out.println(ncr);

}

}

**Prog.15 WAP to compute nPr**

Code:

import java.util.Scanner;

public class Prog15 {

public static long factorial(int number) {

long fact = 1;

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

fact = fact \* i;

}

return fact;

}

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

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

int n = sc.nextInt();

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

int r = sc.nextInt();

long npr;

ncr = factorial(n) / factorial(n - r);

System.out.print("nPr : ");

System.out.println(npr);

}

}

**Prog.16 Check for anagram**

Code:

import java.util.Arrays;

import java.util.Scanner;

public class Prog16 {

static boolean anagrams(String str1, String str2) {

char[] charArray1 = str1.replaceAll("\\s", "").toCharArray();

char[] charArray2 = str2.replaceAll("\\s", "").toCharArray();

Arrays.sort(charArray1);

Arrays.sort(charArray2);

return Arrays.equals(charArray1, charArray2);

}

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

System.out.print("Enter the first string: ");

String str1 = scanner.nextLine().toLowerCase();

System.out.print("Enter the second string: ");

String str2 = scanner.nextLine().toLowerCase();

if (anagrams(str1, str2)) {

System.out.println("The strings are anagrams.");

} else {

System.out.println("The strings are not anagrams.");

}

}

}

**Prog.17 WAP to find the roots of quadratic equation.**

import java.util.Scanner;

public class Prog17 {

public static void roots(float a, float b, float c) {

float root1 = (-b + Math.sqrt((b \* b) - (4 \* a \* c))) / (2.0 \* a);

float root2 = (-b - Math.sqrt((b \* b) - (4 \* a \* c))) / (2.0 \* a);

System.out.println("Roots are : " + root1 + " , " + root2);

}

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

float a, b, c;

System.out.print("Enter Coefficiant if x^2 : ");

a = sc.nextFloat();

System.out.print("Enter Coefficiant if x : ");

b = sc.nextFloat();

System.out.print("Enter Constant : ");

c = sc.nextFloat();

roots(a, b, c);

}

}

**Prog.18 WAP a program to perform linear search.**

Code:

import java.util.Scanner;

public class Prog18 {

public static int linearSearch(int[] array, int target) {

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

if (array[i] == target) {

return i;

}

}

return -1;

}

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

System.out.print("Enter the size of the array: ");

int size = sc.nextInt();

int[] array = new int[size];

System.out.println("Enter the elements of the array:");

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

array[i] = sc.nextInt();

}

System.out.print("Enter the element to search for: ");

int number = sc.nextInt();

int index = linearSearch(array, number);

if (index != -1) {

System.out.println("Element " + number + " found at index " + index);

} else {

System.out.println("Element " + number + " not found in the array");

}

}

}

**Prog.19 WAP to perform binary search.**

Code:

import java.util.Scanner;

public class Prog19 {

static int binarySearch(int[] array, int number) {

int left = 0;

int right = array.length - 1;

while (left <= right) {

int mid = left + (right - left) / 2;

if (array[mid] == number) {

return mid;

} else if (array[mid] < number) {

left = mid + 1;

} else {

right = mid - 1;

}

}

return -1;

}

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

System.out.print("Enter the size of the sorted array: ");

int size = sc.nextInt();

int[] array = new int[size];

System.out.println("Enter the sorted elements of the array:");

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

array[i] = sc.nextInt();

}

System.out.print("Enter the element to search for: ");

int number = sc.nextInt();

int index = binarySearch(array, number);

if (index != -1) {

System.out.println("Element " + number + " found at index " + index);

} else {

System.out.println("Element " + number + " not found in the array");

}

}

}

**Prog.20 Separate out the array in odd and even .**

Code:

import java.util.Scanner;

public class Prog20 {

static void printArray(int[] array, int size) {

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

System.out.print(array[i] + " ");

}

System.out.println();

}

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

System.out.print("Enter the size of the array: ");

int size = sc.nextInt();

int[] array = new int[size];

System.out.println("Enter the elements of the array:");

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

array[i] = sc.nextInt();

}

int[] evenArray = new int[size];

int[] oddArray = new int[size];

int evenIndex = 0, oddIndex = 0;

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

if (array[i] % 2 == 0) {

evenArray[evenIndex++] = array[i];

} else {

oddArray[oddIndex++] = array[i];

}

}

System.out.println("Even numbers:");

printArray(evenArray, evenIndex);

System.out.println("Odd numbers:");

printArray(oddArray, oddIndex);

}

}

**Prog.21 Number of zeros and ones in binary number system.**

Code:

import java.util.Scanner;

public class Prog21 {

static int noOnes(long binaryNumber) {

int count = 0;

while (binaryNumber > 0) {

if (binaryNumber % 10 == 1) {

count++;

}

binaryNumber /= 10;

}

return count;

}

static int noZeros(long binaryNumber) {

int count = 0;

while (binaryNumber > 0) {

if (binaryNumber % 10 == 0) {

count++;

}

binaryNumber /= 10;

}

return count;

}

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

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

long binaryNumber = sc.nextLong();

int Ones = noOnes(binaryNumber);

int Zeros = noZeros(binaryNumber);

System.out.println("Number of 0s: " + Zeros);

System.out.println("Number of 1s: " + Ones);

}

}

**Prog.22 WAP to print Star pattern.**

Code:

import java.util.Scanner;

public class Prog22{

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

System.out.print("Enter the number of rows: ");

int numRows = sc.nextInt();

printStarPattern(numRows);

}

static void printStarPattern(int rows) {

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

for (int j = 1; j <= rows - i; j++) {

System.out.print(" ");

}

for (int k = 1; k <= 2 \* i - 1; k++) {

System.out.print("\*");

}

System.out.println();

}

}

}

**Prog.23 WAP to perform a bitwise OR of 2 binary number in python.**

Code:

def binary\_or(binary\_num1, binary\_num2):

num1 = int(binary\_num1, 2)

num2 = int(binary\_num2, 2)

result = num1 | num2

result\_binary = bin(result)[2:]

return result\_binary

binary\_number1 = input("Enter the first binary number: ")

binary\_number2 = input("Enter the second binary number: ")

result\_binary = binary\_or(binary\_number1, binary\_number2)

print(f"Bitwise OR of {binary\_number1} and {binary\_number2} is: {result\_binary}")

**Prog.24 WAP to perform a bitwise AND of 2 binary numbers.**

Code:

def binary\_or(binary\_num1, binary\_num2):

num1 = int(binary\_num1, 2)

num2 = int(binary\_num2, 2)

result = num1 & num2

result\_binary = bin(result)[2:]

return result\_binary

binary\_number1 = input("Enter the first binary number: ")

binary\_number2 = input("Enter the second binary number: ")

result\_binary = binary\_or(binary\_number1, binary\_number2)

print(f"Bitwise AND of {binary\_number1} and {binary\_number2} is: {result\_binary}")

**Prog.25 ASCII of 0-9 and Alphabate.**

Code:

public class Prog25 {

public static void main(String[] args) {

// Print ASCII values of digits 0-9

System.out.println("ASCII values of digits 0-9:");

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

int asciiValue = (int) '0' + digit;

System.out.println(digit + ": " + asciiValue);

}

// Print ASCII values of alphabets a-z

System.out.println("\nASCII values of alphabets a-z:");

for (char ch = 'a'; ch <= 'z'; ch++) {

int asciiValue = (int) ch;

System.out.println(ch + ": " + asciiValue);

}

}

}