1.Count no of digits in a number:

**Public class count{**

**Public static void main (string[]args){**

**Int num=123456;**

**Int count=0;**

**While(num>0){ //check num=0 stop the process**

**Num=num/10; //123456/6=12345,12345/10=1234 etc…**

**Count++;**

**}**

**System.out.println(number of digits”+count);**

**}**

**2.Reverse a number:**

**Public class reverse{**

**Public static void main(string args[])**

**{**

**Int num=2345;**

**Int rev=0;**

**While(num!=0)**

**{**

**Rev=rev\*10+num%10;**

**Num=num/10;**

**}**

**System.out.println(“reverse a number:”+rev);**

**3.Number of even and odd digits in a number:**

**Public class countoddeven{**

**Public stati void main(string[]args){**

**Int num=1234;**

**Int even\_count=0;**

**Int odd\_count=0;**

**While(num>0)**

**{**

**Int rem=num%10: //4**

**If(rem%2==0){**

**Even\_count++; //1**

**}**

**else{**

**Odd\_count++;**

**}**

**Num=num/10 //123**

**4.sum of digits in a number:**

**1234**

**Sum=0/4/3/2/1**

**Public sumdigit{**

**Public static void miain(string srgs[])**

**{**

**Int num=1234;**

**Int sum=0;**

**While(num>0){**

**Sum=sum+Num%10 //4+3+2+1**

**Num=num/10 //123,12,1**

**}**

**Sysout(“sum of digits”+sum)**

**5.largest of three numbers:**

**Public largesr3numbers{**

**Main{**

**Scanner sc=new scanner(system.in)**

**Sysot(“enter a nuber”)**

**Int a=sc.nextInt();**

**Sysot(“enter a nuber”)**

**Int b=sc.nextInt();**

**Sysot(“enter a nuber”)**

**Int c=sc.nextInt();**

**If(a>b&&a>c){**

**Sysout(a+”a is greter”);**

**Else if(b>a&&b>c){**

**Sysout(b+”b is greter”);**

**Else{**

**Sysout(c+”c is greater”);**

**}**

**6.fibonacci nuber:**

**public class Fibonacci {**

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

**int n1 = 0, n2 = 1, sum = 0;**

**System.out.print(n1 + " " + n2 + " "); // Initial values: 0 1**

**for (int i = 2; i < 10; i++) {**

**sum = n1 + n2; // Calculate the sum of the last two numbers**

**System.out.print(sum + " "); // Print the sum**

**n1 = n2; // Update n1 to the value of n2**

**n2 = sum; // Update n2 to the value of the sum**

**}**

**}**

**}7.prime or not:**

**public class Prime {**

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

**int num = 3;**

**int count = 0;**

**if (num > 1) {**

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

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

**count++;**

**}**

**}**

**if (count == 2) {**

**System.out.println(num + " is a prime number");**

**} else {**

**System.out.println(num + " is not a prime number");**

**}**

**} else {**

**System.out.println("Not a prime number");**

**}**

**}**

**}**

**8.Generate a random number or string:**

**Class{**

**Main{**

**1 approch:**

**Random rand=new random();**

**Int rand\_int=Rand.nextInt(10); //0-9 random number**

**Sys(rand\_int);**

**Rand.nextDouble(); //range 0.0 and less than 1.0 for decimal number**

**2 aroch:**

**Syaout(Math.random());**

**9.Factorial number:**

**public class FactorialExample {**

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

**int num = 5;**

**long factorial = 1;**

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

**factorial = factorial \* i;**

**}**

**System.out.println("Factorial of " + num + " is: " + factorial);**

**}**

**}**

**10.Sum of the elements in an array:**

**Public class sumofarray{**

**Public main{**

**Int a[]={5,2,3,6,7};**

**It sum=0;**

**For(int i=o;i<=4;i++) //if in case we don’t know then i<=a.length-1**

**Sum=sum+arr[i];**

**Sys(sum);**

**11.Array eve and odd nubres fidig:**

**public class EvenOddArray {**

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

**int[] arr = {2, 3, 4, 5, 6, 7};**

**System.out.println("Even numbers in the array:");**

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

**if (arr[i] % 2 == 0) {**

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

**}**

**}**

**System.out.println("Odd numbers in the array:");**

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

**if (arr[i] % 2 != 0) {**

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

**}**

**}**

**}**

**}**

**12.Check 2 arrays are equal or not:**

**public class ArrayComparison {**

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

**int[] a1 = {1, 2, 3, 4};**

**int[] a2 = {1, 2, 3, 4};**

**boolean status = true;**

**if (a1.length == a2.length) {**

**for (int i = 0; i <= a1.length - 1; i++) {**

**if (a1[i] != a2[i]) {**

**status = false;**

**}**

**}**

**if (status) {**

**System.out.println("Arrays are equal");**

**} else {**

**System.out.println("Arrays are not equal");**

**}**

**} else {**

**System.out.println("Arrays are not equal");**

**}**

**}**

**}**

**13.Reverse unsorted array:**

**public class ReverseUnsortedArray {**

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

**int[] array = {1, 2, 8, 6, 4, 7};**

**System.out.println("Original Unsorted Array:");**

**printArray(array);**

**System.out.println("Reversed Unsorted Array:");**

**reverseArray(array);**

**printArray(array);**

**}**

**static void reverseArray(int[] arr) {**

**int start = 0;**

**int end = arr.length - 1;**

**while (start < end) {**

**// Swap elements at start and end indices**

**int temp = arr[start];**

**arr[start] = arr[end];**

**arr[end] = temp;**

**// Move indices towards the center**

**start++;**

**end--;**

**}**

**}**

**static void printArray(int[] arr) {**

**for (int num : arr) {**

**System.out.print(num + " ");**

**}**

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

**}**

**}**

**14.Missing umber in an array:**

**public class MissingNumber {**

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

**int[] a = {1, 2, 3, 4, 5, 6, 8, 9}; // Assuming array should contain 1 to 9**

**int sum1 = 0;**

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

**sum1 = sum1 + a[i];**

**}**

**System.out.println("Sum of array elements: " + sum1);**

**int sum2 = 0;**

**for (int i = 1; i <= 9; i++) {**

**sum2 = sum2 + i;**

**}**

**System.out.println("Sum of consecutive numbers from 1 to 9: " + sum2);**

**System.out.println("Missing Number: " + (sum2 - sum1));**

**}**

**}**

**15.Palindrom or not:**

**public class PalindromeNumber {**

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

**int num=121;**

**int original = number;**

**int reversed = 0;**

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

**int digit = number % 10;**

**reversed = reversed \* 10 + digit;**

**number /= 10;**

**}**

**if (original=number) {**

**System.out.println(num + " is a palindrome number.");**

**} else {**

**System.out.println(num + " is not a palindrome number.");**

**}**

**}**

**}**

**16.Reverse a string:**

**Public class reversestring{**

**Public static void main(string[] args)**

**{**

**String a=”abdfgr”**

**Strig rev=””;**

**Len=a.legth();**

**For(i=len-1;i>=0;i++**

**)**

**Rev=rev+charAt(i);**

**}**

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

**}**

**17.Maximum and minimum of the array:**

**public class MaxMinArray {**

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

**int a[] = {45, 34, 65, 223, 5};**

**int max = a[0];**

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

**if (a[i] > max) {**

**max = a[i];**

**}**

**}**

**System.out.println("Max Value: " + max);**

**int min = a[0];**

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

**if (a[i] < min) {**

**min = a[i];**

**}**

**}**

**System.out.println("Min Value: " + min);**

**}**

**}**

**18.Duplicate element in the array:**

**public class Duplicate {**

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

**int a[] = {2, 3, 4, 5, 4, 6, 2};**

**for (int i = 0; i < a.length - 1; i++) {**

**for (int j = i + 1; j < a.length; j++) {**

**if (a[i] == a[j]) {**

**System.out.println("Duplicate found: " + a[i]);**

**}**

**}**

**}**

**}**

**}**

**19.Linear search and give index of particular key element:**

**public class LinearSearch {**

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

**int a[] = {1, 23, 45, 69, 97};**

**int key = 69;**

**boolean flagValue = false;**

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

**if (a[i] == key) {**

**System.out.println("Key is found: " + a[i]);**

**flagValue = true;**

**}**

**}**

**if (!flagValue) {**

**System.out.println("Key is not found");**

**}**

**}**

**}**

**20.Binary search:**

**public class BinarySearch {**

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

**int a[] = {4, 5, 7, 8, 23, 24, 56};**

**int key = 4;**

**boolean flagValue = false;**

**int l = 0;**

**int h = a.length - 1;**

**while (l <= h) {**

**int mid = (l + h) / 2;**

**if (a[mid] == key) {**

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

**flagValue = true;**

**break;**

**} else if (key > a[mid]) {**

**l = mid + 1;**

**} else {**

**h = mid - 1;**

**}**

**}**

**if (!flagValue) {**

**System.out.println("Element is not found");**

**}**

**}**

**}**

**21.Bubble sort:**

**public class BubbleSort {**

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

**int[] array = {64, 34, 25, 12, 22, 11, 90};**

**System.out.println("Original Array:");**

**printArray(array);**

**// Perform Bubble Sort**

**bubbleSort(array);**

**System.out.println("Sorted Array:");**

**printArray(array);**

**}**

**static void bubbleSort(int[] arr) {**

**int n = arr.length;**

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

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

**if (arr[j] > arr[j + 1]) {**

**// Swap arr[j] and arr[j+1]**

**int temp = arr[j];**

**arr[j] = arr[j + 1];**

**arr[j + 1] = temp;**

**}**

**}**

**}**

**}**

**static void printArray(int[] arr) {**

**for (int num : arr) {**

**System.out.print(num + " ");**

**}**

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

**}**

**}**

**Arrays:**

**23.Second largest number in the array:**

**public class SecondLargest {**

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

**// Define the array of integers**

**int[] a = {12, 345, 67, 78, 45};**

**// Initialize variables to store the two largest elements**

**int max1, max2;**

**// Compare the first two elements to determine max1 and max2**

**if (a[0] > a[1]) {**

**max1 = a[0];**

**max2 = a[1];**

**} else {**

**max1 = a[1];**

**max2 = a[0];**

**}**

**// Iterate through the array starting from the third element**

**for (int i = 2; i < a.length; i++) {**

**// If the current element is greater than the current max1**

**if (a[i] > max1) {**

**// Update max2 to the previous max1**

**max2 = max1;**

**// Update max1 to the current element**

**max1 = a[i];**

**} else if (a[i] > max2) {**

**// If the current element is greater than max2 but not greater than max1**

**// Update max2 to the current element**

**max2 = a[i];**

**}**

**}**

**// Print the second largest element**

**System.out.println("Second Largest Element: " + max2);**

**}**

**}**

**24.Count the large consecutive in aaarticular number:**

import java.util.Scanner;

public class MyClass {

static int countConse(int n)

{

int count=0;

while(n>0)

{

n=( n & (n << 1));

count++;

}

return count;

}

public static void main(String args[]) {

Scanner sc=new Scanner(System.in);

int n=sc.nextInt();

System.out.println((countConse(n)));

}

}

25.Tower honoi:

It will take 7 steps.they fallow some rules only one disk move at one time and larger disk can’t place on smaller one.

The method takes four parameters: **n** (number of disks), **src** (source rod), **aux** (auxiliary rod), and **dest** (destination rod).

If there is only one disk to move (base case), it prints the move from the source rod to the destination rod.

If there are more than one disk, it recursively calls the **towerOfHanoi** method to move **n-1** disks from the source rod to the auxiliary rod, then moves one disk from the source rod to the destination rod, and finally, it recursively moves the **n-1** disks from the auxiliary rod to the destination rod.

Your **main** method then calls **towerOfHanoi(3, 'A', 'B', 'C');**, starting with 3 disks on rod 'A', using rod 'B' as the auxiliary, and moving the disks to rod 'C'.

public class hanoi

{

static void towerOfHonai(int n,char src,char aux,char dest){

if(n==1){

System.out.println(src+"-->"+dest);

return;

}

towerOfHonai(n-1,src,dest,aux);

towerOfHonai(1,src,aux,dest);

towerOfHonai(n-1,aux,src,dest);

}

public static void main(String[] args) {

towerOfHonai(3,'A','B','C');

}

}

26.Power of a number:

import java.util.Scanner;

public class Main

{

public static void main(String[] args) {

Scanner sc=new Scanner(System.in);

System.out.println("enter the base");

int base=sc.nextInt();

System.out.println("enter the exponent");

int exponet=sc.nextInt();

int result=1;

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

result=result\*base;

}

System.out.println("power of"+result);

}

}

**27.Tap academy:**

**Gcd:**

**ex a=20 and b=15**

**first which one is maximum**

**20%15=5🡪a=5,b=15**

**15%5=0🡪a=5,b=0(when ever the zero in any number then non zero elemnt is the gcd of that numbers).**

**import java.util.Scanner;**

**public class GCD {**

**static int euclidGCD(int a, int b) {**

**while (a != 0 && b != 0) {**

**if (a > b) {**

**a = a % b;**

**} else {**

**b = b % a;**

**}**

**}**

**if (a != 0) {**

**return a;**

**} else {**

**return b;**

**}**

**}**

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

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

**// Taking input for two integers**

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

**int a = sc.nextInt();**

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

**int b = sc.nextInt();**

**// Calculating and displaying the GCD**

**int result = euclidGCD(a, b);**

**System.out.println("The Greatest Common Divisor (GCD) of " + a + " and " + b + " is: " + result);**

**// Closing the scanner**

**sc.close();**

**}**

**}**

**28.Prime number:**

**One or itself only devisiable.**

import java.util.Scanner;

public class prime{

static boolean isprime(int n){

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

if(n%i==0){

return false;

}

else{

return true;

}

}

return true;

}

public static void main (String[]args){

Scanner sc=new Scanner(System.in);

Int n=sc.nextInt();

System.out.println(isprie(n));

29.Find divisors:

In this we start with 1 only not the zero.

import java.util.Scanner;

public class divisors{

static void div(int n){

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

if(n%i==0){

System.out.println(i);

}

}

}

public static void main (String[]args){

Scanner sc=new Scanner(System.in);

System.out.println("enter a integer");

int n=sc.nextInt();

div(n);

}

}

30.Square root finding:

public class Main {

public static void main(String[] args) {

boolean t = false;

int n = 33;

int i;

for (i = 1; i <= n / 2; i++) {

if (i \* i == n) {

t = true;

break; // Break out of the loop if a square root is found

}

}

if (t) {

System.out.println("Yes, square root is " + i);

} else {

System.out.println("No, " + n + " is not a perfect square.");

}

}

}

31.Perfect number:

If a number divible by zero(that factors) adding after also that is same number is aerfect number.

import java.util.Scanner;

public class Main {

static boolean isperfectnumber(int n){

if(n<=0){

return false;

}

int sum=0;

for(int i=1;i<=n/2;i++){

if(n%i==0){

sum=sum+i;

}

}

return sum==n;

}

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

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

int n=sc.nextInt();

if(isperfectnumber(n)){

System.out.println("perfect number");

}

else{

System.out.println("not a perfect umber");

}

}

}

32.Binary to decimal:

import java.util.Scanner;

public class Main {

public static int binaryToDecimal(long num) {

int decimal = 0, power = 0;

long rem;

while (num > 0) {

rem = num % 10;

num = num / 10;

decimal = decimal + (int) (rem \* Math.pow(2, power));

power++;

}

return decimal;

}

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

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

long num = sc.nextLong();

int n = binaryToDecimal(num);

System.out.println("Decimal: " + n);

}

}

Or)

import java.util.Scanner;

public class OctalToDecimal {

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

System.out.println("Enter an octal number:");

String octalStr = sc.next();

try {

// Converting octal to decimal

int decimalNumber = Integer.parseInt(octalStr, 8);

// Printing the result

System.out.println("Decimal: " + decimalNumber);

} catch (NumberFormatException e) {

System.out.println("Invalid octal number. Please enter a valid octal number.");

}

}

}

33.Any to decimal same to all:

import java.util.Scanner;

public class HexToDecimal {

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

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

String hexStr = sc.next();

try {

// Converting hexadecimal to decimal

int decimalNumber = Integer.parseInt(hexStr, 16);

// Printing the result

System.out.println("Decimal: " + decimalNumber);

} catch (NumberFormatException e) {

System.out.println("Invalid hexadecimal number. Please enter a valid hexadecimal number.");

}

}

}

34.Decimal to binary:

import java.util.Scanner;

public class DecimalToBinary {

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

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

try {

int decimalNumber = sc.nextInt();

// Converting decimal to binary

String binaryStr = Integer.toBinaryString(decimalNumber);

// Printing the result

System.out.println("Binary: " + binaryStr);

} catch (java.util.InputMismatchException e) {

System.out.println("Invalid input. Please enter a valid decimal number.");

}

}

}

35.Perfect number or abudent number or deficient:

import java.util.Scanner;

public class Main {

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

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

int n=sc.nextInt();

int sum=0;

for(int i=1;i<=n/2;i++){

if(n%i==0){

sum=sum+i;

}

}

if(sum>n){

System.out.println("abudent number");

}

else {

if(sum<n)

System.out.println("is a deficient number");

else{

System.out.println("perfect number");

}

}

}

}

36.Area of rectangle:

public class Main {

public static void main(String[] args) {

int l = 20;

int w = 30;

long area = (long) l \* w; // Corrected the variable names

long perimeter = 2L \* (l + w); // Corrected the variable names

System.out.println("Area: " + area);

System.out.println("Perimeter: " + perimeter);

}

}

37.Array

import java.util.Arrays;

public class Main {

public static void main(String[] args) {

int a[] = {10, 3, 3, 44, 44, 40, 56, 30};

// Sorting the array

Arrays.sort(a);

// Printing the sorted array

for (int i = 0; i < a.length-1; i++) {

if(a[i]==a[i+1]){

System.out.println("duplicates are:"+a[i]);

}

//else if(a[i]!=a[i+1]){

//System.out.println("unique"+a[i]);

}

}

}

38.Max and value of array:

public class Main {

public static void main(String[] args) {

int a[] = {16, 9, 6, 13};

int n = a.length; // Added to get the length of the array

int maxAndValue = 0;

int maxAndElement1 = 0;

int maxAndElement2 = 0;

for (int i = 0; i < n - 1; i++) { // Corrected loop condition

for (int j = i + 1; j < n; j++) { // Start j from i+1 to avoid comparing elements with themselves

int currentAndValue = a[i] & a[j];

if (currentAndValue > maxAndValue) {

maxAndValue = currentAndValue;

maxAndElement1 = a[i];

maxAndElement2 = a[j];

}

}

}

System.out.println("Maximum AND value: " + maxAndValue);

System.out.println("Pair of elements: (" + maxAndElement1 + ", " + maxAndElement2 + ")");

}

}

39.Reverse the array:

public class Main{

static void reverse(int a[]){

int i=0;

int j=a.length-1;

while(i<j){

int t=a[i];

a[i]=a[j];

a[j]=t;

i++;

j--;

}

for(int k=0;k<a.length;k++){

System.out.print(a[k]);

}

}

public static void main(String[]args){

int a[]={1,2,3,4,5};

reverse(a);

}

}

40.Subarray:

public class Main {

public static void subarray(int a[]) {

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

for (int j = i; j < a.length; j++) {

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

System.out.print(a[k] + " ");

}

System.out.println();

}

}

}

public static void main(String[] args) {

int a[] = {1, 2, 3, 4, 5};

subarray(a);

}

}

41.Sorted array or not checking:

public class Main {

static boolean sorted(int a[]) {

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

if (a[i] < a[i - 1]) {

return false;

}

}

return true;

}

public static void main(String[] args) {

int a[] = {1, 4, 6, 7, 8, 9};

boolean isSorted = sorted(a);

if (isSorted) {

System.out.println("The array is sorted.");

} else {

System.out.println("The array is not sorted.");

}

}

}

42.Linear search:

public class LinearSearch {

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

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

if (arr[i] == target) {

return i; // Return the index if the target is found

}

}

return -1; // Return -1 if the target is not found

}

public static void main(String[] args) {

int[] array = {2, 5, 8, 12, 16, 23, 38, 42, 50};

int targetElement = 23;

int result = linearSearch(array, targetElement);

if (result != -1) {

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

} else {

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

}

}

}

43Patterns:

public class Main

{

public static void main(String[] args) {

int n=5;

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

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

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

}

System.out.println();

}

}

Output:

\*

\*\*

\*\*\*

44

public class Main

{

public static void main(String[] args) {

int n=5;

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

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

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

}

System.out.println();

}

}

}

Output:

\*\*\*\*

\*\*\*

\*\*

\*

45.

public class Main

{

public static void main(String[] args) {

for(int i=1;i<=4;i++){

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

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

}

System.out.println();

}

for(int i=1;i<=4;i++){

for(int j=3;j>=i;j--)

{

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

}

System.out.println();

}

}

Output:

\*

\*\*

\*\*\*

\*\*

\*

46.

public class Main

{

public static void main(String[] args) {

for(int i=1;i<=4;i++){

for(int j=3;j>=i;j--){

System.out.print(" ");

}

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

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

}

System.out.println();

}

}

}

Output:

\*

\*\*

\*\*\*

\*\*\*\*

47.

public class Main

{

public static void main(String[] args) {

for(int i=1;i<=4;i++){

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

System.out.print(" ");

}

for(int k=4;k>=i;k--){

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

}

System.out.println();

}

}

Output:

\*\*\*\*

\*\*\*

\*\*

\*

48.

public class Main

{

public static void main(String[] args) {

for(int i=1;i<=4;i++){

for(int j=3;j>=i;j--){

System.out.print(" ");

}

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

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

}

System.out.println();

}

for(int i=1;i<=4;i++){

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

System.out.print(" ");

}

for(int k=3;k>=i;k--){

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

}

System.out.println();

}

Output:

\*

\*\*

\*\*\*

\*\*\*\*

\*\*\*

\*\*

\*

49.Pyramid:

public class Main

{

public static void main(String[] args) {

for(int i=1;i<=4;i++)

{

for(int j=3;j>=i;j--){

System.out.print(" ");

}

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

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

}

System.out.println();

}

}

Output:

\*

\_ \*

\*\_ \* \_\*

50

public class Main

{

public static void main(String[] args) {

for(int i=1;i<=4;i++)

{

for(int j=3;j>=i;j--){

System.out.print(" ");

}

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

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

}

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

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

}

System.out.println();

}

}

}

\*

\*\*\*

\*\*\*\*\*

\*\*\*\*\*\*\*

51.

public class Main

{

public static void main(String[] args) {

for(int i=1;i<=5;i++)

{

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

System.out.print(" ");

}

for(int k=9;k>(i\*2);k--){

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

}

System.out.println();

}

}

}

(Or)

public class Main

{

public static void main(String[] args) {

for(int i=1;i<=5;i++)

{

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

System.out.print(" ");

}

for(int k=4;k>=i;k--){

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

}

for(int l=3;l>=i;l--){

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

}

System.out.println();

}

}

}

Output:

\*\*\*\*\*

\*\*\*

\*

52.

public class Main {

public static void main(String[] args) {

for (int i = 1; i <= 5; i++) {

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

if (i >= 2 && j <= i - 1) {

System.out.print(" ");

} else {

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

}

}

System.out.println();

}

}

}

\*

\*

\*

53.

public class Main {

public static void main(String[] args) {

for (int i = 1; i <= 5; i++) {

for(int j=5;j>i;j--){

System.out.print(" ");

}

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

if (k>1 && k < (i\*2) - 1) {

System.out.print(" ");

} else {

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

}

}

System.out.println();

}

}

}

Output:

public class Main {

public static void main(String[] args) {

for (int i = 1; i <= 5; i++) {

for(int j=5;j>i;j--){

System.out.print(" ");

}

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

if (k>1 && k < (i\*2) - 1) {

System.out.print(" ");

} else {

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

}

}

System.out.println();

}

}

}

54.Numbers:

public class Main

{

public static void main(String[] args) {

for(int i=1;i<=4;i++){

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

System.out.print(k);

}

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

System.out.print(j);

}

System.out.println();

}

}

}

Output:

11

1221

123321

12344321

55.public class Main

{

public static void main(String[] args) {

for(int i=1;i<=5;i++){

int no=i;

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

System.out.print(no);

no=no+5-j;

}

System.out.println();

}

}

}

Output:

1

2 5

3 6 8

4 7 9 10

56.

public class Main

{

public static void main(String[] args) {

int count=0;

for(int i=1;i<=5;i++){

for(int j=1;j<=3;j++){

count=count+1;

System.out.print(count);

}

System.out.println();

}

}

}

Output:

123

456

789

101112